

glucat

0.12.0

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	cga3 Namespace Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Function Documentation . . . . .	9
5.1.2.1	agc3() . . . . .	9
5.1.2.2	cga3() . . . . .	10
5.1.2.3	cga3std() . . . . .	10
5.2	glucat Namespace Reference . . . . .	10
5.2.1	Typedef Documentation . . . . .	23
5.2.1.1	index_t . . . . .	23
5.2.1.2	intfn . . . . .	23
5.2.1.3	intintfn . . . . .	23
5.2.1.4	set_value_t . . . . .	23

5.2.1.5	tuning_fast	24
5.2.1.6	tuning_naive	24
5.2.1.7	tuning_slow	24
5.2.2	Function Documentation	24
5.2.2.1	_GLUCAT_CTAssert() [1/3]	24
5.2.2.2	_GLUCAT_CTAssert() [2/3]	25
5.2.2.3	_GLUCAT_CTAssert() [3/3]	25
5.2.2.4	abs()	25
5.2.2.5	acos() [1/2]	25
5.2.2.6	acos() [2/2]	26
5.2.2.7	acosh() [1/2]	26
5.2.2.8	acosh() [2/2]	26
5.2.2.9	approx_equal() [1/2]	27
5.2.2.10	approx_equal() [2/2]	27
5.2.2.11	asin() [1/2]	27
5.2.2.12	asin() [2/2]	28
5.2.2.13	asinh() [1/2]	28
5.2.2.14	asinh() [2/2]	28
5.2.2.15	atan() [1/2]	29
5.2.2.16	atan() [2/2]	29
5.2.2.17	atanh() [1/2]	29
5.2.2.18	atanh() [2/2]	30
5.2.2.19	cascade_log()	30
5.2.2.20	check_complex()	30
5.2.2.21	clifford_exp()	31
5.2.2.22	compare()	31
5.2.2.23	complexifier()	31
5.2.2.24	conj()	32
5.2.2.25	cos() [1/2]	32
5.2.2.26	cos() [2/2]	32

5.2.2.27	<code>cosh()</code>	33
5.2.2.28	<code>cr_sqrt()</code>	33
5.2.2.29	<code>crd_of_mult()</code> [1/2]	33
5.2.2.30	<code>crd_of_mult()</code> [2/2]	34
5.2.2.31	<code>db_sqrt()</code>	34
5.2.2.32	<code>db_step()</code>	34
5.2.2.33	<code>elliptic()</code>	35
5.2.2.34	<code>error_squared()</code>	35
5.2.2.35	<code>error_squared_tol()</code>	35
5.2.2.36	<code>even()</code>	36
5.2.2.37	<code>exp()</code> [1/2]	36
5.2.2.38	<code>exp()</code> [2/2]	36
5.2.2.39	<code>fast()</code>	37
5.2.2.40	<code>folded_dim()</code>	37
5.2.2.41	<code>imag()</code>	37
5.2.2.42	<code>inv()</code>	37
5.2.2.43	<code>inverse_gray()</code>	38
5.2.2.44	<code>inverse_reversed_gray()</code>	38
5.2.2.45	<code>involute()</code>	38
5.2.2.46	<code>log()</code> [1/4]	38
5.2.2.47	<code>log()</code> [2/4]	39
5.2.2.48	<code>log()</code> [3/4]	39
5.2.2.49	<code>log()</code> [4/4]	39
5.2.2.50	<code>log2()</code>	40
5.2.2.51	<code>matrix_log()</code>	40
5.2.2.52	<code>matrix_sqrt()</code>	40
5.2.2.53	<code>max_abs()</code>	41
5.2.2.54	<code>max_pos()</code>	41
5.2.2.55	<code>min_neg()</code>	41
5.2.2.56	<code>norm()</code>	41

5.2.2.57	<code>odd()</code>	42
5.2.2.58	<code>offset_level()</code>	42
5.2.2.59	<code>operator &amp;()</code> [1/8]	42
5.2.2.60	<code>operator &amp;()</code> [2/8]	43
5.2.2.61	<code>operator &amp;()</code> [3/8]	43
5.2.2.62	<code>operator &amp;()</code> [4/8]	43
5.2.2.63	<code>operator &amp;()</code> [5/8]	44
5.2.2.64	<code>operator &amp;()</code> [6/8]	44
5.2.2.65	<code>operator &amp;()</code> [7/8]	44
5.2.2.66	<code>operator &amp;()</code> [8/8]	45
5.2.2.67	<code>operator !=()</code> [1/3]	45
5.2.2.68	<code>operator !=()</code> [2/3]	45
5.2.2.69	<code>operator !=()</code> [3/3]	46
5.2.2.70	<code>operator %()</code> [1/3]	46
5.2.2.71	<code>operator %()</code> [2/3]	46
5.2.2.72	<code>operator %()</code> [3/3]	47
5.2.2.73	<code>operator *()</code> [1/6]	47
5.2.2.74	<code>operator *()</code> [2/6]	47
5.2.2.75	<code>operator *()</code> [3/6]	48
5.2.2.76	<code>operator *()</code> [4/6]	48
5.2.2.77	<code>operator *()</code> [5/6]	48
5.2.2.78	<code>operator *()</code> [6/6]	49
5.2.2.79	<code>operator +()</code> [1/3]	49
5.2.2.80	<code>operator +()</code> [2/3]	49
5.2.2.81	<code>operator +()</code> [3/3]	50
5.2.2.82	<code>operator -()</code> [1/3]	50
5.2.2.83	<code>operator -()</code> [2/3]	50
5.2.2.84	<code>operator -()</code> [3/3]	51
5.2.2.85	<code>operator /()</code> [1/5]	51
5.2.2.86	<code>operator /()</code> [2/5]	51

5.2.2.87	<code>operator/()</code> [3/5]	52
5.2.2.88	<code>operator/()</code> [4/5]	52
5.2.2.89	<code>operator/()</code> [5/5]	52
5.2.2.90	<code>operator&lt;&lt;()</code> [1/4]	53
5.2.2.91	<code>operator&lt;&lt;()</code> [2/4]	53
5.2.2.92	<code>operator&lt;&lt;()</code> [3/4]	53
5.2.2.93	<code>operator&lt;&lt;()</code> [4/4]	53
5.2.2.94	<code>operator&gt;&gt;()</code> [1/3]	54
5.2.2.95	<code>operator&gt;&gt;()</code> [2/3]	54
5.2.2.96	<code>operator&gt;&gt;()</code> [3/3]	54
5.2.2.97	<code>operator^()</code> [1/4]	54
5.2.2.98	<code>operator^()</code> [2/4]	55
5.2.2.99	<code>operator^()</code> [3/4]	55
5.2.2.100	<code>operator^()</code> [4/4]	55
5.2.2.101	<code>operator"   ()</code> [1/4]	56
5.2.2.102	<code>operator"   ()</code> [2/4]	56
5.2.2.103	<code>operator"   ()</code> [3/4]	56
5.2.2.104	<code>operator"   ()</code> [4/4]	57
5.2.2.105	<code>outer_pow()</code>	57
5.2.2.106	<code>pade_approx()</code>	57
5.2.2.107	<code>pade_log()</code>	58
5.2.2.108	<code>pos_mod()</code>	58
5.2.2.109	<code>pow()</code> [1/2]	58
5.2.2.110	<code>pow()</code> [2/2]	59
5.2.2.111	<code>pure()</code>	59
5.2.2.112	<code>quad()</code>	59
5.2.2.113	<code>real()</code>	59
5.2.2.114	<code>reframe()</code>	60
5.2.2.115	<code>reverse()</code>	60
5.2.2.116	<code>scalar()</code>	60

5.2.2.117	<code>sign_of_square()</code>	61
5.2.2.118	<code>sin()</code> [1/2]	61
5.2.2.119	<code>sin()</code> [2/2]	61
5.2.2.120	<code>sinh()</code>	62
5.2.2.121	<code>sqrt()</code> [1/4]	62
5.2.2.122	<code>sqrt()</code> [2/4]	62
5.2.2.123	<code>sqrt()</code> [3/4]	63
5.2.2.124	<code>sqrt()</code> [4/4]	63
5.2.2.125	<code>star()</code> [1/3]	63
5.2.2.126	<code>star()</code> [2/3]	64
5.2.2.127	<code>star()</code> [3/3]	64
5.2.2.128	<code>tan()</code> [1/2]	64
5.2.2.129	<code>tan()</code> [2/2]	65
5.2.2.130	<code>tanh()</code>	65
5.2.2.131	<code>to_demote()</code>	65
5.2.2.132	<code>to_promote()</code>	66
5.2.2.133	<code>try_catch()</code> [1/2]	66
5.2.2.134	<code>try_catch()</code> [2/2]	66
5.2.2.135	<code>vector_part()</code>	66
5.2.3	Variable Documentation	67
5.2.3.1	<code>BITS_PER_SET_VALUE</code>	67
5.2.3.2	<code>DEFAULT_HI</code>	67
5.2.3.3	<code>I_ln2</code>	67
5.2.3.4	<code>I_pi</code>	67
5.2.3.5	<code>MS_PER_S</code>	67
5.2.3.6	<code>Tuning_Fast_Basis_Max_Count</code>	68
5.2.3.7	<code>Tuning_Fast_CR_Sqrt_Max_Steps</code>	68
5.2.3.8	<code>Tuning_Fast_DB_Sqrt_Max_Steps</code>	68
5.2.3.9	<code>Tuning_Fast_Div_Max_Steps</code>	68
5.2.3.10	<code>Tuning_Fast_Fast_Size_Threshold</code>	68



5.2.3.11	Tuning_Fast_Inv_Fast_Dim_Threshold . . . . .	68
5.2.3.12	Tuning_Fast_Log_Max_Inner_Steps . . . . .	69
5.2.3.13	Tuning_Fast_Log_Max_Outer_Steps . . . . .	69
5.2.3.14	Tuning_Fast_Mult_Matrix_Threshold . . . . .	69
5.2.3.15	Tuning_Fast_Products_Size_Threshold . . . . .	69
5.2.3.16	Tuning_Int_Digits . . . . .	69
5.2.3.17	Tuning_Max_Threshold . . . . .	69
5.2.3.18	Tuning_Naive_Basis_Max_Count . . . . .	70
5.2.3.19	Tuning_Naive_Fast_Size_Threshold . . . . .	70
5.2.3.20	Tuning_Naive_Inv_Fast_Dim_Threshold . . . . .	70
5.2.3.21	Tuning_Naive_Mult_Matrix_Threshold . . . . .	70
5.2.3.22	Tuning_Slow_Basis_Max_Count . . . . .	70
5.2.3.23	Tuning_Slow_Fast_Size_Threshold . . . . .	70
5.2.3.24	Tuning_Slow_Inv_Fast_Dim_Threshold . . . . .	71
5.2.3.25	Tuning_Slow_Mult_Matrix_Threshold . . . . .	71
5.2.3.26	Tuning_Slow_Products_Size_Threshold . . . . .	71
5.3	glucat::gen Namespace Reference . . . . .	71
5.3.1	Typedef Documentation . . . . .	71
5.3.1.1	signature_t . . . . .	72
5.3.2	Variable Documentation . . . . .	72
5.3.2.1	offset_to_super . . . . .	72
5.4	glucat::matrix Namespace Reference . . . . .	72
5.4.1	Typedef Documentation . . . . .	74
5.4.1.1	eig_case_t . . . . .	74
5.4.2	Function Documentation . . . . .	74
5.4.2.1	classify_eigenvalues() . . . . .	74
5.4.2.2	eigenvalues() . . . . .	74
5.4.2.3	inner() . . . . .	75
5.4.2.4	isinf() . . . . .	75
5.4.2.5	isnan() . . . . .	75

5.4.2.6	<code>kron()</code>	75
5.4.2.7	<code>mono_kron()</code>	76
5.4.2.8	<code>mono_prod()</code>	76
5.4.2.9	<code>nnz()</code>	76
5.4.2.10	<code>nork()</code>	77
5.4.2.11	<code>nork_range()</code>	77
5.4.2.12	<code>norm_frob2()</code>	77
5.4.2.13	<code>prod()</code>	78
5.4.2.14	<code>signed_perm_nork()</code>	78
5.4.2.15	<code>sparse_prod()</code>	78
5.4.2.16	<code>to_lapack()</code>	79
5.4.2.17	<code>trace()</code>	79
5.4.2.18	<code>unit()</code>	79
5.5	<code>glucat::timing</code> Namespace Reference	79
5.5.1	Function Documentation	80
5.5.1.1	<code>elapsed()</code>	80
5.5.2	Variable Documentation	80
5.5.2.1	<code>EXTRA_TRIALS</code>	80
5.5.2.2	<code>MS_PER_CLOCK</code>	80
5.5.2.3	<code>MS_PER_SEC</code>	80
5.6	<code>pade</code> Namespace Reference	81
5.7	<code>PyClical</code> Namespace Reference	81
5.7.1	Function Documentation	82
5.7.1.1	<code>_test()</code>	82
5.7.1.2	<code>clifford_hidden_doctests()</code>	82
5.7.1.3	<code>e()</code>	83
5.7.1.4	<code>index_set_hidden_doctests()</code>	84
5.7.1.5	<code>istpq()</code>	85
5.7.2	Variable Documentation	85
5.7.2.1	<code>__version__</code>	86

5.7.2.2	cl	86
5.7.2.3	fill	86
5.7.2.4	i	86
5.7.2.5	ist	86
5.7.2.6	ixt	87
5.7.2.7	lhs	87
5.7.2.8	nbar3	87
5.7.2.9	ninf3	87
5.7.2.10	None	87
5.7.2.11	obj	88
5.7.2.12	pi	88
5.7.2.13	rhs	88
5.7.2.14	scalar_epsilon	88
5.7.2.15	tau	88
5.7.2.16	threshold	89
5.7.2.17	tol	89
5.8	std Namespace Reference	89
<b>6</b>	<b>Class Documentation</b>	<b>91</b>
6.1	glucat::basis_table< Scalar_T, LO, HI, Matrix_T > Class Template Reference	91
6.1.1	Detailed Description	92
6.1.2	Constructor & Destructor Documentation	92
6.1.2.1	basis_table() [1/2]	92
6.1.2.2	~basis_table()	92
6.1.2.3	basis_table() [2/2]	92
6.1.3	Member Function Documentation	92
6.1.3.1	basis()	93
6.1.3.2	operator=()	93
6.1.4	Friends And Related Function Documentation	93
6.1.4.1	friend_for_private_destructor	93
6.2	glucat::bool_to_type< truth_value > Class Template Reference	93

6.2.1	Detailed Description	94
6.2.2	Member Enumeration Documentation	94
6.2.2.1	anonymous enum	94
6.3	PyClical.clifford Class Reference	94
6.3.1	Detailed Description	96
6.3.2	Member Function Documentation	96
6.3.2.1	__add__()	96
6.3.2.2	__and__()	96
6.3.2.3	__call__()	97
6.3.2.4	__cinit__()	97
6.3.2.5	__contains__()	98
6.3.2.6	__dealloc__()	98
6.3.2.7	__getitem__()	98
6.3.2.8	__iadd__()	99
6.3.2.9	__iand__()	99
6.3.2.10	__idiv__()	99
6.3.2.11	__imod__()	100
6.3.2.12	__imul__()	100
6.3.2.13	__ior__()	100
6.3.2.14	__isub__()	101
6.3.2.15	__iter__()	101
6.3.2.16	__ixor__()	101
6.3.2.17	__mod__()	102
6.3.2.18	__mul__()	102
6.3.2.19	__neg__()	102
6.3.2.20	__or__()	103
6.3.2.21	__pos__()	103
6.3.2.22	__pow__()	103
6.3.2.23	__repr__()	104
6.3.2.24	__richcmp__()	104

6.3.2.25	<a href="#">__str__()</a>	104
6.3.2.26	<a href="#">__sub__()</a>	105
6.3.2.27	<a href="#">__truediv__()</a>	105
6.3.2.28	<a href="#">__xor__()</a>	105
6.3.2.29	<a href="#">abs()</a>	106
6.3.2.30	<a href="#">conj()</a>	106
6.3.2.31	<a href="#">even()</a>	106
6.3.2.32	<a href="#">frame()</a>	107
6.3.2.33	<a href="#">inv()</a>	107
6.3.2.34	<a href="#">involute()</a>	107
6.3.2.35	<a href="#">isinf()</a>	108
6.3.2.36	<a href="#">isnan()</a>	108
6.3.2.37	<a href="#">max_abs()</a>	108
6.3.2.38	<a href="#">norm()</a>	109
6.3.2.39	<a href="#">odd()</a>	109
6.3.2.40	<a href="#">outer_pow()</a>	109
6.3.2.41	<a href="#">pow()</a>	110
6.3.2.42	<a href="#">pure()</a>	110
6.3.2.43	<a href="#">quad()</a>	110
6.3.2.44	<a href="#">reframe()</a>	111
6.3.2.45	<a href="#">reverse()</a>	111
6.3.2.46	<a href="#">scalar()</a>	111
6.3.2.47	<a href="#">truncated()</a>	112
6.3.2.48	<a href="#">vector_part()</a>	112
6.3.3	<a href="#">Member Data Documentation</a>	112
6.3.3.1	<a href="#">instance</a>	112
6.4	<a href="#">glucat::clifford_algebra&lt; Scalar_T, Index_Set_T, Multivector_T &gt; Class Template Reference</a>	113
6.4.1	<a href="#">Detailed Description</a>	115
6.4.2	<a href="#">Member Typedef Documentation</a>	115
6.4.2.1	<a href="#">index_set_t</a>	115

6.4.2.2	<code>multivector_t</code>	116
6.4.2.3	<code>pair_t</code>	116
6.4.2.4	<code>scalar_t</code>	116
6.4.2.5	<code>vector_t</code>	116
6.4.3	Constructor & Destructor Documentation	116
6.4.3.1	<code>~clifford_algebra()</code>	116
6.4.4	Member Function Documentation	117
6.4.4.1	<code>classname()</code>	117
6.4.4.2	<code>conj()</code>	117
6.4.4.3	<code>even()</code>	117
6.4.4.4	<code>frame()</code>	117
6.4.4.5	<code>grade()</code>	118
6.4.4.6	<code>inv()</code>	118
6.4.4.7	<code>involute()</code>	118
6.4.4.8	<code>isinf()</code>	118
6.4.4.9	<code>isnan()</code>	118
6.4.4.10	<code>max_abs()</code>	119
6.4.4.11	<code>norm()</code>	119
6.4.4.12	<code>odd()</code>	119
6.4.4.13	<code>operator &amp;=()</code>	119
6.4.4.14	<code>operator %=()</code>	119
6.4.4.15	<code>operator ()()</code>	120
6.4.4.16	<code>operator *=()</code> [1/2]	120
6.4.4.17	<code>operator *=()</code> [2/2]	120
6.4.4.18	<code>operator +=()</code> [1/2]	120
6.4.4.19	<code>operator +=()</code> [2/2]	120
6.4.4.20	<code>operator -()</code>	121
6.4.4.21	<code>operator -=()</code> [1/2]	121
6.4.4.22	<code>operator -=()</code> [2/2]	121
6.4.4.23	<code>operator /=()</code> [1/2]	121

6.4.4.24	<a href="#">operator/=( ) [ 2 / 2 ]</a>	121
6.4.4.25	<a href="#">operator==( ) [ 1 / 2 ]</a>	122
6.4.4.26	<a href="#">operator==( ) [ 2 / 2 ]</a>	122
6.4.4.27	<a href="#">operator[]()</a>	122
6.4.4.28	<a href="#">operator^=( )</a>	122
6.4.4.29	<a href="#">operator"   =( )</a>	122
6.4.4.30	<a href="#">outer_pow()</a>	123
6.4.4.31	<a href="#">pow()</a>	123
6.4.4.32	<a href="#">pure()</a>	123
6.4.4.33	<a href="#">quad()</a>	123
6.4.4.34	<a href="#">reverse()</a>	123
6.4.4.35	<a href="#">scalar()</a>	124
6.4.4.36	<a href="#">truncated()</a>	124
6.4.4.37	<a href="#">vector_part() [ 1 / 2 ]</a>	124
6.4.4.38	<a href="#">vector_part() [ 2 / 2 ]</a>	124
6.4.4.39	<a href="#">write() [ 1 / 2 ]</a>	124
6.4.4.40	<a href="#">write() [ 2 / 2 ]</a>	125
6.4.5	<a href="#">Member Data Documentation</a>	125
6.4.5.1	<a href="#">default_truncation</a>	125
6.4.5.2	<a href="#">v_hi</a>	125
6.4.5.3	<a href="#">v_lo</a>	125
6.5	<a href="#">glucat::compare_types&lt; LHS_T, RHS_T &gt; Class Template Reference</a>	126
6.5.1	<a href="#">Detailed Description</a>	126
6.5.2	<a href="#">Member Enumeration Documentation</a>	126
6.5.2.1	<a href="#">anonymous enum</a>	126
6.6	<a href="#">glucat::compare_types&lt; T, T &gt; Class Template Reference</a>	126
6.6.1	<a href="#">Detailed Description</a>	127
6.6.2	<a href="#">Member Enumeration Documentation</a>	127
6.6.2.1	<a href="#">anonymous enum</a>	127
6.7	<a href="#">glucat::control_t Class Reference</a>	127

6.7.1	Detailed Description	128
6.7.2	Constructor & Destructor Documentation	128
6.7.2.1	control_t() [1/3]	128
6.7.2.2	control_t() [2/3]	129
6.7.2.3	~control_t()	129
6.7.2.4	control_t() [3/3]	129
6.7.3	Member Function Documentation	129
6.7.3.1	call() [1/2]	129
6.7.3.2	call() [2/2]	129
6.7.3.3	catch_exceptions()	130
6.7.3.4	control()	130
6.7.3.5	operator=()	130
6.7.3.6	valid()	130
6.7.3.7	verbose()	130
6.7.4	Friends And Related Function Documentation	131
6.7.4.1	friend_for_private_destructor	131
6.7.5	Member Data Documentation	131
6.7.5.1	m_catch_exceptions	131
6.7.5.2	m_valid	131
6.7.5.3	m_verbose_output	131
6.8	glucat::CTAssertion< bool > Struct Template Reference	132
6.8.1	Detailed Description	132
6.9	glucat::CTAssertion< true > Struct Template Reference	132
6.9.1	Detailed Description	132
6.10	glucat::numeric_traits< Scalar_T >::demoted<> Struct Template Reference	132
6.10.1	Detailed Description	133
6.10.2	Member Typedef Documentation	133
6.10.2.1	type [1/2]	133
6.10.2.2	type [2/2]	133
6.11	glucat::matrix::eig_genus< Matrix_T > Struct Template Reference	133



6.11.1 Detailed Description . . . . .	134
6.11.2 Member Typedef Documentation . . . . .	134
6.11.2.1 Scalar_T . . . . .	134
6.11.3 Member Data Documentation . . . . .	134
6.11.3.1 m_eig_case . . . . .	134
6.11.3.2 m_is_singular . . . . .	135
6.11.3.3 m_safe_arg . . . . .	135
6.12 glucat::error< Class_T > Class Template Reference . . . . .	135
6.12.1 Detailed Description . . . . .	136
6.12.2 Constructor & Destructor Documentation . . . . .	136
6.12.2.1 error() [1/2] . . . . .	136
6.12.2.2 error() [2/2] . . . . .	137
6.12.3 Member Function Documentation . . . . .	137
6.12.3.1 classname() . . . . .	137
6.12.3.2 heading() . . . . .	137
6.12.3.3 print_error_msg() . . . . .	137
6.13 glucat::framed_multi< Scalar_T, LO, HI, Tune_P > Class Template Reference . . . . .	138
6.13.1 Detailed Description . . . . .	141
6.13.2 Member Typedef Documentation . . . . .	141
6.13.2.1 const_iterator . . . . .	141
6.13.2.2 error_t . . . . .	141
6.13.2.3 framed_multi_t . . . . .	141
6.13.2.4 framed_pair_t . . . . .	142
6.13.2.5 index_set_t . . . . .	142
6.13.2.6 iterator . . . . .	142
6.13.2.7 map_t . . . . .	142
6.13.2.8 matrix_multi_t . . . . .	142
6.13.2.9 matrix_t . . . . .	143
6.13.2.10 multivector_t . . . . .	143
6.13.2.11 scalar_t . . . . .	143

6.13.2.12	<code>size_type</code>	143
6.13.2.13	<code>sorted_map_t</code>	143
6.13.2.14	<code>term_t</code>	144
6.13.2.15	<code>tune_p</code>	144
6.13.2.16	<code>var_term_t</code>	144
6.13.2.17	<code>vector_t</code>	144
6.13.3	Constructor & Destructor Documentation	144
6.13.3.1	<code>~framed_multi()</code>	144
6.13.3.2	<code>framed_multi()</code> [1/15]	145
6.13.3.3	<code>framed_multi()</code> [2/15]	145
6.13.3.4	<code>framed_multi()</code> [3/15]	145
6.13.3.5	<code>framed_multi()</code> [4/15]	145
6.13.3.6	<code>framed_multi()</code> [5/15]	146
6.13.3.7	<code>framed_multi()</code> [6/15]	146
6.13.3.8	<code>framed_multi()</code> [7/15]	146
6.13.3.9	<code>framed_multi()</code> [8/15]	147
6.13.3.10	<code>framed_multi()</code> [9/15]	147
6.13.3.11	<code>framed_multi()</code> [10/15]	147
6.13.3.12	<code>framed_multi()</code> [11/15]	147
6.13.3.13	<code>framed_multi()</code> [12/15]	148
6.13.3.14	<code>framed_multi()</code> [13/15]	148
6.13.3.15	<code>framed_multi()</code> [14/15]	148
6.13.3.16	<code>framed_multi()</code> [15/15]	148
6.13.4	Member Function Documentation	149
6.13.4.1	<code>centre_pm4_qp4()</code>	149
6.13.4.2	<code>centre_pp4_qm4()</code>	149
6.13.4.3	<code>centre_qp1_pm1()</code>	149
6.13.4.4	<code>classname()</code>	150
6.13.4.5	<code>divide()</code>	150
6.13.4.6	<code>fast()</code>	150

6.13.4.7	<a href="#">fast_framed_multi()</a>	150
6.13.4.8	<a href="#">fast_matrix_multi()</a>	151
6.13.4.9	<a href="#">fold()</a>	151
6.13.4.10	<a href="#">nbr_terms()</a>	151
6.13.4.11	<a href="#">operator+=()</a>	151
6.13.4.12	<a href="#">random()</a>	152
6.13.4.13	<a href="#">unfold()</a>	152
6.13.5	<a href="#">Friends And Related Function Documentation</a>	152
6.13.5.1	<a href="#">exp</a>	152
6.13.5.2	<a href="#">framed_multi</a>	152
6.13.5.3	<a href="#">matrix_multi</a>	153
6.13.5.4	<a href="#">operator &amp;</a>	153
6.13.5.5	<a href="#">operator%</a>	153
6.13.5.6	<a href="#">operator*</a>	153
6.13.5.7	<a href="#">operator/</a>	153
6.13.5.8	<a href="#">operator&lt;&lt; [1/2]</a>	154
6.13.5.9	<a href="#">operator&lt;&lt; [2/2]</a>	154
6.13.5.10	<a href="#">operator&gt;&gt;</a>	154
6.13.5.11	<a href="#">operator^</a>	154
6.13.5.12	<a href="#">operator"  </a>	154
6.13.5.13	<a href="#">star</a>	155
6.14	<a href="#">glucat::gen::generator_table&lt; Matrix_T &gt; Class Template Reference</a>	155
6.14.1	<a href="#">Detailed Description</a>	156
6.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	156
6.14.2.1	<a href="#">generator_table() [1/2]</a>	157
6.14.2.2	<a href="#">~generator_table()</a>	157
6.14.2.3	<a href="#">generator_table() [2/2]</a>	157
6.14.3	<a href="#">Member Function Documentation</a>	157
6.14.3.1	<a href="#">gen_from_pm1_qm1()</a>	157
6.14.3.2	<a href="#">gen_from_pm4_qp4()</a>	157

6.14.3.3	<a href="#">gen_from_pp4_qm4()</a>	158
6.14.3.4	<a href="#">gen_from_qp1_pm1()</a>	158
6.14.3.5	<a href="#">gen_vector()</a>	158
6.14.3.6	<a href="#">generator()</a>	158
6.14.3.7	<a href="#">operator()()</a>	159
6.14.3.8	<a href="#">operator=()</a>	159
6.14.4	<a href="#">Friends And Related Function Documentation</a>	159
6.14.4.1	<a href="#">friend_for_private_destructor</a>	159
6.15	<a href="#">glucat::glucat_error Class Reference</a>	160
6.15.1	<a href="#">Detailed Description</a>	161
6.15.2	<a href="#">Constructor &amp; Destructor Documentation</a>	161
6.15.2.1	<a href="#">glucat_error()</a>	161
6.15.2.2	<a href="#">~glucat_error()</a>	161
6.15.3	<a href="#">Member Function Documentation</a>	161
6.15.3.1	<a href="#">classname()</a>	161
6.15.3.2	<a href="#">heading()</a>	161
6.15.3.3	<a href="#">print_error_msg()</a>	162
6.15.4	<a href="#">Member Data Documentation</a>	162
6.15.4.1	<a href="#">name</a>	162
6.16	<a href="#">glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt;::hash_size_t Class Reference</a>	162
6.16.1	<a href="#">Detailed Description</a>	162
6.16.2	<a href="#">Constructor &amp; Destructor Documentation</a>	162
6.16.2.1	<a href="#">hash_size_t()</a>	163
6.16.3	<a href="#">Member Function Documentation</a>	163
6.16.3.1	<a href="#">operator()()</a>	163
6.16.4	<a href="#">Member Data Documentation</a>	163
6.16.4.1	<a href="#">n</a>	163
6.17	<a href="#">glucat::index_set&lt; LO, HI &gt; Class Template Reference</a>	164
6.17.1	<a href="#">Detailed Description</a>	167
6.17.2	<a href="#">Member Typedef Documentation</a>	167

6.17.2.1	<a href="#">bitset_t</a>	167
6.17.2.2	<a href="#">error_t</a>	167
6.17.2.3	<a href="#">index_pair_t</a>	167
6.17.2.4	<a href="#">index_set_t</a>	168
6.17.3	<a href="#">Constructor &amp; Destructor Documentation</a>	168
6.17.3.1	<a href="#">index_set() [1/6]</a>	168
6.17.3.2	<a href="#">index_set() [2/6]</a>	168
6.17.3.3	<a href="#">index_set() [3/6]</a>	168
6.17.3.4	<a href="#">index_set() [4/6]</a>	169
6.17.3.5	<a href="#">index_set() [5/6]</a>	169
6.17.3.6	<a href="#">index_set() [6/6]</a>	169
6.17.4	<a href="#">Member Function Documentation</a>	169
6.17.4.1	<a href="#">BOOST_STATIC_ASSERT()</a>	169
6.17.4.2	<a href="#">classname()</a>	170
6.17.4.3	<a href="#">count()</a>	170
6.17.4.4	<a href="#">count_neg()</a>	170
6.17.4.5	<a href="#">count_pos()</a>	170
6.17.4.6	<a href="#">flip() [1/2]</a>	171
6.17.4.7	<a href="#">flip() [2/2]</a>	171
6.17.4.8	<a href="#">fold() [1/2]</a>	171
6.17.4.9	<a href="#">fold() [2/2]</a>	171
6.17.4.10	<a href="#">hash_fn()</a>	172
6.17.4.11	<a href="#">is_contiguous()</a>	172
6.17.4.12	<a href="#">lex_less_than()</a>	172
6.17.4.13	<a href="#">max()</a>	172
6.17.4.14	<a href="#">min()</a>	173
6.17.4.15	<a href="#">operator &amp;=()</a>	173
6.17.4.16	<a href="#">operator"! =()</a>	173
6.17.4.17	<a href="#">operator&lt;()</a>	173
6.17.4.18	<a href="#">operator==()</a>	174

6.17.4.19	<a href="#">operator[]()</a> [1/2]	174
6.17.4.20	<a href="#">operator[]()</a> [2/2]	174
6.17.4.21	<a href="#">operator^=()</a>	174
6.17.4.22	<a href="#">operator"  =()</a>	175
6.17.4.23	<a href="#">operator~()</a>	175
6.17.4.24	<a href="#">reset()</a> [1/2]	175
6.17.4.25	<a href="#">reset()</a> [2/2]	175
6.17.4.26	<a href="#">set()</a> [1/3]	176
6.17.4.27	<a href="#">set()</a> [2/3]	176
6.17.4.28	<a href="#">set()</a> [3/3]	176
6.17.4.29	<a href="#">sign_of_mult()</a>	176
6.17.4.30	<a href="#">sign_of_square()</a>	177
6.17.4.31	<a href="#">test()</a>	177
6.17.4.32	<a href="#">unfold()</a>	177
6.17.4.33	<a href="#">value_of_fold()</a>	177
6.17.5	<a href="#">Friends And Related Function Documentation</a>	178
6.17.5.1	<a href="#">compare</a>	178
6.17.5.2	<a href="#">operator &amp;</a>	178
6.17.5.3	<a href="#">operator^</a>	178
6.17.5.4	<a href="#">operator"  </a>	178
6.17.5.5	<a href="#">reference</a>	178
6.17.6	<a href="#">Member Data Documentation</a>	179
6.17.6.1	<a href="#">v_hi</a>	179
6.17.6.2	<a href="#">v_lo</a>	179
6.18	<a href="#">PyClical.index_set Class Reference</a>	179
6.18.1	<a href="#">Detailed Description</a>	180
6.18.2	<a href="#">Member Function Documentation</a>	180
6.18.2.1	<a href="#">__and__()</a>	181
6.18.2.2	<a href="#">__cinit__()</a>	181
6.18.2.3	<a href="#">__contains__()</a>	181

6.18.2.4	<a href="#">__dealloc__()</a>	182
6.18.2.5	<a href="#">__getitem__()</a>	182
6.18.2.6	<a href="#">__iand__()</a>	183
6.18.2.7	<a href="#">__invert__()</a>	183
6.18.2.8	<a href="#">__ior__()</a>	183
6.18.2.9	<a href="#">__iter__()</a>	184
6.18.2.10	<a href="#">__ixor__()</a>	184
6.18.2.11	<a href="#">__or__()</a>	184
6.18.2.12	<a href="#">__repr__()</a>	185
6.18.2.13	<a href="#">__richcmp__()</a>	185
6.18.2.14	<a href="#">__setitem__()</a>	185
6.18.2.15	<a href="#">__str__()</a>	186
6.18.2.16	<a href="#">__xor__()</a>	186
6.18.2.17	<a href="#">count()</a>	186
6.18.2.18	<a href="#">count_neg()</a>	187
6.18.2.19	<a href="#">count_pos()</a>	187
6.18.2.20	<a href="#">hash_fn()</a>	187
6.18.2.21	<a href="#">max()</a>	188
6.18.2.22	<a href="#">min()</a>	188
6.18.2.23	<a href="#">sign_of_mult()</a>	188
6.18.2.24	<a href="#">sign_of_square()</a>	189
6.18.3	<a href="#">Member Data Documentation</a>	189
6.18.3.1	<a href="#">instance</a>	189
6.19	<a href="#">glucat::index_set_hash&lt; LO, HI &gt; Class Template Reference</a>	189
6.19.1	<a href="#">Detailed Description</a>	190
6.19.2	<a href="#">Member Typedef Documentation</a>	190
6.19.2.1	<a href="#">index_set_t</a>	190
6.19.3	<a href="#">Member Function Documentation</a>	190
6.19.3.1	<a href="#">operator()()</a>	190
6.20	<a href="#">glucat::matrix_multi&lt; Scalar_T, LO, HI, Tune_P &gt; Class Template Reference</a>	191

6.20.1 Detailed Description . . . . .	193
6.20.2 Member Typedef Documentation . . . . .	194
6.20.2.1 basis_matrix_t . . . . .	194
6.20.2.2 error_t . . . . .	194
6.20.2.3 framed_multi_t . . . . .	194
6.20.2.4 index_set_t . . . . .	194
6.20.2.5 matrix_index_t . . . . .	195
6.20.2.6 matrix_multi_t . . . . .	195
6.20.2.7 matrix_t . . . . .	195
6.20.2.8 multivector_t . . . . .	195
6.20.2.9 orientation_t . . . . .	195
6.20.2.10 scalar_t . . . . .	196
6.20.2.11 term_t . . . . .	196
6.20.2.12 tune_p . . . . .	196
6.20.2.13 vector_t . . . . .	196
6.20.3 Constructor & Destructor Documentation . . . . .	196
6.20.3.1 ~matrix_multi() . . . . .	196
6.20.3.2 matrix_multi() [1/17] . . . . .	197
6.20.3.3 matrix_multi() [2/17] . . . . .	197
6.20.3.4 matrix_multi() [3/17] . . . . .	197
6.20.3.5 matrix_multi() [4/17] . . . . .	198
6.20.3.6 matrix_multi() [5/17] . . . . .	198
6.20.3.7 matrix_multi() [6/17] . . . . .	198
6.20.3.8 matrix_multi() [7/17] . . . . .	199
6.20.3.9 matrix_multi() [8/17] . . . . .	199
6.20.3.10 matrix_multi() [9/17] . . . . .	199
6.20.3.11 matrix_multi() [10/17] . . . . .	199
6.20.3.12 matrix_multi() [11/17] . . . . .	200
6.20.3.13 matrix_multi() [12/17] . . . . .	200
6.20.3.14 matrix_multi() [13/17] . . . . .	200



6.20.3.15	<a href="#">matrix_multi()</a> [14/17]	200
6.20.3.16	<a href="#">matrix_multi()</a> [15/17]	201
6.20.3.17	<a href="#">matrix_multi()</a> [16/17]	201
6.20.3.18	<a href="#">matrix_multi()</a> [17/17]	201
6.20.4	<a href="#">Member Function Documentation</a>	201
6.20.4.1	<a href="#">basis_element()</a>	202
6.20.4.2	<a href="#">classname()</a>	202
6.20.4.3	<a href="#">fast_framed_multi()</a>	202
6.20.4.4	<a href="#">fast_matrix_multi()</a>	202
6.20.4.5	<a href="#">operator+=()</a>	203
6.20.4.6	<a href="#">operator=()</a>	203
6.20.4.7	<a href="#">random()</a>	203
6.20.5	<a href="#">Friends And Related Function Documentation</a>	203
6.20.5.1	<a href="#">framed_multi</a>	204
6.20.5.2	<a href="#">matrix_log</a>	204
6.20.5.3	<a href="#">matrix_multi</a>	204
6.20.5.4	<a href="#">matrix_sqrt</a>	204
6.20.5.5	<a href="#">operator &amp;</a>	205
6.20.5.6	<a href="#">operator%</a>	205
6.20.5.7	<a href="#">operator*</a>	205
6.20.5.8	<a href="#">operator/</a>	205
6.20.5.9	<a href="#">operator&lt;&lt;</a>	205
6.20.5.10	<a href="#">operator&gt;&gt;</a>	206
6.20.5.11	<a href="#">operator^</a>	206
6.20.5.12	<a href="#">operator"  </a>	206
6.20.5.13	<a href="#">reframe</a>	206
6.20.5.14	<a href="#">star</a>	206
6.20.6	<a href="#">Member Data Documentation</a>	207
6.20.6.1	<a href="#">m_frame</a>	207
6.20.6.2	<a href="#">m_matrix</a>	207

6.21	<a href="#">std::numeric_limits&lt; glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt; &gt; Struct Template Reference</a>	208
6.21.1	Detailed Description	208
6.22	<a href="#">std::numeric_limits&lt; glucat::matrix_multi&lt; Scalar_T, LO, HI, Tune_P &gt; &gt; Struct Template Reference</a>	209
6.22.1	Detailed Description	209
6.23	<a href="#">glucat::numeric_traits&lt; Scalar_T &gt; Class Template Reference</a>	210
6.23.1	Detailed Description	212
6.23.2	Member Function Documentation	212
6.23.2.1	<a href="#">abs()</a>	212
6.23.2.2	<a href="#">acos()</a>	213
6.23.2.3	<a href="#">asin()</a>	213
6.23.2.4	<a href="#">atan()</a>	213
6.23.2.5	<a href="#">conj()</a>	213
6.23.2.6	<a href="#">cos()</a>	214
6.23.2.7	<a href="#">cosh()</a>	214
6.23.2.8	<a href="#">exp()</a>	214
6.23.2.9	<a href="#">fmod()</a>	214
6.23.2.10	<a href="#">imag()</a>	215
6.23.2.11	<a href="#">isInf()</a> [1/3]	215
6.23.2.12	<a href="#">isInf()</a> [2/3]	215
6.23.2.13	<a href="#">isInf()</a> [3/3]	215
6.23.2.14	<a href="#">isNaN()</a> [1/3]	216
6.23.2.15	<a href="#">isNaN()</a> [2/3]	216
6.23.2.16	<a href="#">isNaN()</a> [3/3]	216
6.23.2.17	<a href="#">isNaN_or_isInf()</a>	216
6.23.2.18	<a href="#">ln_2()</a> [1/2]	217
6.23.2.19	<a href="#">ln_2()</a> [2/2]	217
6.23.2.20	<a href="#">log()</a>	217
6.23.2.21	<a href="#">log2()</a>	217
6.23.2.22	<a href="#">NaN()</a>	218
6.23.2.23	<a href="#">pi()</a> [1/2]	218

6.23.2.24 <code>pi()</code> [2/2]	218
6.23.2.25 <code>pow()</code>	218
6.23.2.26 <code>real()</code>	219
6.23.2.27 <code>sin()</code>	219
6.23.2.28 <code>sinh()</code>	219
6.23.2.29 <code>sqrt()</code>	219
6.23.2.30 <code>tan()</code>	220
6.23.2.31 <code>tanh()</code>	220
6.23.2.32 <code>to_double()</code>	220
6.23.2.33 <code>to_int()</code>	220
6.23.2.34 <code>to_scalar_t()</code> [1/9]	221
6.23.2.35 <code>to_scalar_t()</code> [2/9]	221
6.23.2.36 <code>to_scalar_t()</code> [3/9]	221
6.23.2.37 <code>to_scalar_t()</code> [4/9]	221
6.23.2.38 <code>to_scalar_t()</code> [5/9]	222
6.23.2.39 <code>to_scalar_t()</code> [6/9]	222
6.23.2.40 <code>to_scalar_t()</code> [7/9]	222
6.23.2.41 <code>to_scalar_t()</code> [8/9]	222
6.23.2.42 <code>to_scalar_t()</code> [9/9]	223
6.24 <code>pade::pade_log_denom&lt; Scalar_T &gt;</code> Struct Template Reference	223
6.24.1 Detailed Description	223
6.24.2 Member Typedef Documentation	223
6.24.2.1 <code>array</code>	224
6.24.3 Member Data Documentation	224
6.24.3.1 <code>denom</code>	224
6.25 <code>pade::pade_log_denom&lt; dd_real &gt;</code> Struct Template Reference	224
6.25.1 Detailed Description	225
6.25.2 Member Typedef Documentation	225
6.25.2.1 <code>array</code>	225
6.25.3 Member Data Documentation	225

6.25.3.1	denom	225
6.26	pade::pade_log_denom< float > Struct Template Reference	225
6.26.1	Detailed Description	226
6.26.2	Member Typedef Documentation	226
6.26.2.1	array	226
6.26.3	Member Data Documentation	226
6.26.3.1	denom	226
6.27	pade::pade_log_denom< long double > Struct Template Reference	227
6.27.1	Detailed Description	227
6.27.2	Member Typedef Documentation	227
6.27.2.1	array	227
6.27.3	Member Data Documentation	227
6.27.3.1	denom	227
6.28	pade::pade_log_denom< qd_real > Struct Template Reference	228
6.28.1	Detailed Description	228
6.28.2	Member Typedef Documentation	228
6.28.2.1	array	228
6.28.3	Member Data Documentation	228
6.28.3.1	denom	229
6.29	pade::pade_log_numer< Scalar_T > Struct Template Reference	229
6.29.1	Detailed Description	230
6.29.2	Member Typedef Documentation	230
6.29.2.1	array	230
6.29.3	Member Data Documentation	230
6.29.3.1	numer	230
6.30	pade::pade_log_numer< dd_real > Struct Template Reference	230
6.30.1	Detailed Description	231
6.30.2	Member Typedef Documentation	231
6.30.2.1	array	231
6.30.3	Member Data Documentation	231

6.30.3.1	numer	231
6.31	pade::pade_log_numer< float > Struct Template Reference	232
6.31.1	Detailed Description	232
6.31.2	Member Typedef Documentation	232
6.31.2.1	array	232
6.31.3	Member Data Documentation	232
6.31.3.1	numer	232
6.32	pade::pade_log_numer< long double > Struct Template Reference	233
6.32.1	Detailed Description	233
6.32.2	Member Typedef Documentation	233
6.32.2.1	array	233
6.32.3	Member Data Documentation	233
6.32.3.1	numer	233
6.33	pade::pade_log_numer< qd_real > Struct Template Reference	234
6.33.1	Detailed Description	234
6.33.2	Member Typedef Documentation	234
6.33.2.1	array	234
6.33.3	Member Data Documentation	234
6.33.3.1	numer	235
6.34	pade::pade_sqrt_denom< Scalar_T > Struct Template Reference	235
6.34.1	Detailed Description	236
6.34.2	Member Typedef Documentation	236
6.34.2.1	array	236
6.34.3	Member Data Documentation	236
6.34.3.1	denom	236
6.35	pade::pade_sqrt_denom< dd_real > Struct Template Reference	236
6.35.1	Detailed Description	237
6.35.2	Member Typedef Documentation	237
6.35.2.1	array	237
6.35.3	Member Data Documentation	237

6.35.3.1	denom	237
6.36	pade::pade_sqrt_denom< float > Struct Template Reference	238
6.36.1	Detailed Description	238
6.36.2	Member Typedef Documentation	238
6.36.2.1	array	238
6.36.3	Member Data Documentation	238
6.36.3.1	denom	238
6.37	pade::pade_sqrt_denom< long double > Struct Template Reference	239
6.37.1	Detailed Description	239
6.37.2	Member Typedef Documentation	239
6.37.2.1	array	239
6.37.3	Member Data Documentation	239
6.37.3.1	denom	239
6.38	pade::pade_sqrt_denom< qd_real > Struct Template Reference	240
6.38.1	Detailed Description	240
6.38.2	Member Typedef Documentation	240
6.38.2.1	array	240
6.38.3	Member Data Documentation	240
6.38.3.1	denom	241
6.39	pade::pade_sqrt_numer< Scalar_T > Struct Template Reference	241
6.39.1	Detailed Description	242
6.39.2	Member Typedef Documentation	242
6.39.2.1	array	242
6.39.3	Member Data Documentation	242
6.39.3.1	numer	242
6.40	pade::pade_sqrt_numer< dd_real > Struct Template Reference	242
6.40.1	Detailed Description	243
6.40.2	Member Typedef Documentation	243
6.40.2.1	array	243
6.40.3	Member Data Documentation	243

6.40.3.1	<a href="#">numer</a>	243
6.41	<a href="#">pade::pade_sqrt_numer&lt; float &gt; Struct Template Reference</a>	244
6.41.1	<a href="#">Detailed Description</a>	244
6.41.2	<a href="#">Member Typedef Documentation</a>	244
6.41.2.1	<a href="#">array</a>	244
6.41.3	<a href="#">Member Data Documentation</a>	244
6.41.3.1	<a href="#">numer</a>	244
6.42	<a href="#">pade::pade_sqrt_numer&lt; long double &gt; Struct Template Reference</a>	245
6.42.1	<a href="#">Detailed Description</a>	245
6.42.2	<a href="#">Member Typedef Documentation</a>	245
6.42.2.1	<a href="#">array</a>	245
6.42.3	<a href="#">Member Data Documentation</a>	245
6.42.3.1	<a href="#">numer</a>	245
6.43	<a href="#">pade::pade_sqrt_numer&lt; qd_real &gt; Struct Template Reference</a>	246
6.43.1	<a href="#">Detailed Description</a>	246
6.43.2	<a href="#">Member Typedef Documentation</a>	246
6.43.2.1	<a href="#">array</a>	246
6.43.3	<a href="#">Member Data Documentation</a>	246
6.43.3.1	<a href="#">numer</a>	247
6.44	<a href="#">glucat::numeric_traits&lt; Scalar_T &gt;::promoted&lt;&gt; Struct Template Reference</a>	247
6.44.1	<a href="#">Detailed Description</a>	248
6.44.2	<a href="#">Member Typedef Documentation</a>	248
6.44.2.1	<a href="#">type [1/3]</a>	248
6.44.2.2	<a href="#">type [2/3]</a>	248
6.44.2.3	<a href="#">type [3/3]</a>	248
6.45	<a href="#">glucat::random_generator&lt; Scalar_T &gt; Class Template Reference</a>	249
6.45.1	<a href="#">Detailed Description</a>	249
6.45.2	<a href="#">Constructor &amp; Destructor Documentation</a>	250
6.45.2.1	<a href="#">random_generator() [1/2]</a>	250
6.45.2.2	<a href="#">random_generator() [2/2]</a>	250

6.45.2.3	<a href="#">~random_generator()</a>	250
6.45.3	<a href="#">Member Function Documentation</a>	250
6.45.3.1	<a href="#">generator()</a>	250
6.45.3.2	<a href="#">normal()</a>	251
6.45.3.3	<a href="#">operator=()</a>	251
6.45.3.4	<a href="#">uniform()</a>	251
6.45.4	<a href="#">Friends And Related Function Documentation</a>	251
6.45.4.1	<a href="#">friend_for_private_destructor</a>	251
6.45.5	<a href="#">Member Data Documentation</a>	251
6.45.5.1	<a href="#">normal_dist</a>	252
6.45.5.2	<a href="#">seed</a>	252
6.45.5.3	<a href="#">uint_gen</a>	252
6.45.5.4	<a href="#">uniform_dist</a>	252
6.46	<a href="#">glucat::index_set&lt; LO, HI &gt;::reference Class Reference</a>	253
6.46.1	<a href="#">Detailed Description</a>	254
6.46.2	<a href="#">Constructor &amp; Destructor Documentation</a>	254
6.46.2.1	<a href="#">reference() [1/2]</a>	254
6.46.2.2	<a href="#">reference() [2/2]</a>	254
6.46.2.3	<a href="#">~reference()</a>	254
6.46.3	<a href="#">Member Function Documentation</a>	254
6.46.3.1	<a href="#">flip()</a>	255
6.46.3.2	<a href="#">operator bool()</a>	255
6.46.3.3	<a href="#">operator=() [1/2]</a>	255
6.46.3.4	<a href="#">operator=() [2/2]</a>	255
6.46.3.5	<a href="#">operator==()</a>	256
6.46.3.6	<a href="#">operator~()</a>	256
6.46.4	<a href="#">Friends And Related Function Documentation</a>	256
6.46.4.1	<a href="#">index_set</a>	256
6.46.5	<a href="#">Member Data Documentation</a>	256
6.46.5.1	<a href="#">m_idx</a>	256



6.46.5.2	<code>m_pst</code>	257
6.47	<code>glucat::sorted_range&lt; Map_T, Sorted_Map_T &gt;</code> Class Template Reference	257
6.47.1	Detailed Description	257
6.47.2	Member Typedef Documentation	257
6.47.2.1	<code>map_t</code>	258
6.47.2.2	<code>sorted_iterator</code>	258
6.47.2.3	<code>sorted_map_t</code>	258
6.47.3	Constructor & Destructor Documentation	258
6.47.3.1	<code>sorted_range()</code>	258
6.47.4	Member Data Documentation	258
6.47.4.1	<code>sorted_begin</code>	259
6.47.4.2	<code>sorted_end</code>	259
6.48	<code>glucat::sorted_range&lt; Sorted_Map_T, Sorted_Map_T &gt;</code> Class Template Reference	259
6.48.1	Detailed Description	259
6.48.2	Member Typedef Documentation	260
6.48.2.1	<code>map_t</code>	260
6.48.2.2	<code>sorted_iterator</code>	260
6.48.2.3	<code>sorted_map_t</code>	260
6.48.3	Constructor & Destructor Documentation	260
6.48.3.1	<code>sorted_range()</code>	260
6.48.4	Member Data Documentation	260
6.48.4.1	<code>sorted_begin</code>	261
6.48.4.2	<code>sorted_end</code>	261
6.49	<code>glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt;::var_term</code> Class Reference	261
6.49.1	Detailed Description	262
6.49.2	Member Typedef Documentation	263
6.49.2.1	<code>var_pair_t</code>	263
6.49.3	Constructor & Destructor Documentation	263
6.49.3.1	<code>~var_term()</code>	263
6.49.3.2	<code>var_term()</code> [1/2]	263
6.49.3.3	<code>var_term()</code> [2/2]	263
6.49.4	Member Function Documentation	264
6.49.4.1	<code>classname()</code>	264
6.49.4.2	<code>operator*=( )</code>	264

<b>7 File Documentation</b>	<b>265</b>
7.1 glucat/clifford_algebra.h File Reference	265
7.1.1 Macro Definition Documentation	272
7.1.1.1 _GLUCAT_CLIFFORD_ALGEBRA_OPERATIONS	272
7.2 glucat/clifford_algebra_imp.h File Reference	272
7.3 glucat/errors.h File Reference	279
7.4 glucat/errors_imp.h File Reference	280
7.5 glucat/framed_multi.h File Reference	281
7.6 glucat/framed_multi_imp.h File Reference	283
7.6.1 Macro Definition Documentation	286
7.6.1.1 _GLUCAT_HASH_N	286
7.6.1.2 _GLUCAT_HASH_SIZE_T	286
7.7 glucat/generation.h File Reference	286
7.8 glucat/generation_imp.h File Reference	287
7.9 glucat/global.h File Reference	288
7.9.1 Macro Definition Documentation	290
7.9.1.1 _GLUCAT_CTAssert	290
7.10 glucat/glucat.h File Reference	290
7.11 glucat/glucat_config.h File Reference	291
7.11.1 Macro Definition Documentation	291
7.11.1.1 GLUCAT_HAVE_CXX11	291
7.11.1.2 GLUCAT_HAVE_INTTYPES_H	292
7.11.1.3 GLUCAT_HAVE_STDINT_H	292
7.11.1.4 GLUCAT_HAVE_STDIO_H	292
7.11.1.5 GLUCAT_HAVE_STDLIB_H	292
7.11.1.6 GLUCAT_HAVE_STRING_H	292
7.11.1.7 GLUCAT_HAVE_STRINGS_H	292
7.11.1.8 GLUCAT_HAVE_SYS_STAT_H	293
7.11.1.9 GLUCAT_HAVE_SYS_TYPES_H	293
7.11.1.10 GLUCAT_HAVE_UNISTD_H	293

7.11.1.11 GLUCAT_PACKAGE	293
7.11.1.12 GLUCAT_PACKAGE_BUGREPORT	293
7.11.1.13 GLUCAT_PACKAGE_NAME	293
7.11.1.14 GLUCAT_PACKAGE_STRING	294
7.11.1.15 GLUCAT_PACKAGE_TARNAME	294
7.11.1.16 GLUCAT_PACKAGE_URL	294
7.11.1.17 GLUCAT_PACKAGE_VERSION	294
7.11.1.18 GLUCAT_STDC_HEADERS	294
7.11.1.19 GLUCAT_VERSION	294
7.12 glucat/glucat_imp.h File Reference	295
7.13 glucat/index_set.h File Reference	295
7.14 glucat/index_set_imp.h File Reference	297
7.15 glucat/long_double.h File Reference	298
7.16 glucat/matrix.h File Reference	299
7.17 glucat/matrix_imp.h File Reference	301
7.18 glucat/matrix_multi.h File Reference	303
7.19 glucat/matrix_multi_imp.h File Reference	306
7.20 glucat/portability.h File Reference	309
7.20.1 Macro Definition Documentation	310
7.20.1.1 _GLUCAT_ISINF	310
7.20.1.2 _GLUCAT_ISNAN	310
7.20.1.3 UBLAS_ABS	310
7.20.1.4 UBLAS_SQRT	310
7.21 glucat/promotion.h File Reference	311
7.22 glucat/qd.h File Reference	312
7.23 glucat/random.h File Reference	312
7.24 glucat/scalar.h File Reference	313
7.25 glucat/scalar_imp.h File Reference	315
7.26 glucat/tuning.h File Reference	316
7.26.1 Function Documentation	316

7.26.1.1	<code>_GLUCAT_CTAssert()</code>	316
7.27	<code>test/tuning.h</code> File Reference	317
7.28	<code>pyclical/glucat.pxd</code> File Reference	318
7.29	<code>pyclical/PyClical.h</code> File Reference	318
7.29.1	Typedef Documentation	319
7.29.1.1	<code>Clifford</code>	319
7.29.1.2	<code>IndexSet</code>	320
7.29.1.3	<code>scalar_t</code>	320
7.29.1.4	<code>String</code>	320
7.29.2	Function Documentation	320
7.29.2.1	<code>clifford_to_repr()</code>	320
7.29.2.2	<code>clifford_to_str()</code>	320
7.29.2.3	<code>index_set_to_repr()</code>	321
7.29.2.4	<code>index_set_to_str()</code>	321
7.29.2.5	<code>PyFloat_FromDouble()</code>	321
7.29.3	Variable Documentation	321
7.29.3.1	<code>epsilon</code>	322
7.29.3.2	<code>glucat_package_version</code>	322
7.29.3.3	<code>hi_ndx</code>	322
7.29.3.4	<code>lo_ndx</code>	322
7.30	<code>pyclical/PyClical.pxd</code> File Reference	322
7.31	<code>pyclical/PyClical.pyx</code> File Reference	322
7.32	<code>pyclical/PyClical_nocython.cpp</code> File Reference	323
7.32.1	Macro Definition Documentation	324
7.32.1.1	<code>PY_SSIZE_T_CLEAN</code>	324
7.33	<code>test/control.h</code> File Reference	324
7.34	<code>test/driver.h</code> File Reference	325
7.35	<code>test/timing.h</code> File Reference	325
7.36	<code>test/try_catch.h</code> File Reference	326

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">cga3</a>	Definitions for 3D Conformal Geometric Algebra [DL]	9
<a href="#">glucat</a>		10
<a href="#">glucat::gen</a>		71
<a href="#">glucat::matrix</a>		72
<a href="#">glucat::timing</a>		79
<a href="#">pade</a>		81
<a href="#">PyClical</a>		81
<a href="#">std</a>		89



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

bitset	
glucat::index_set< LO, HI > . . . . .	164
glucat::bool_to_type< truth_value > . . . . .	93
cdef	
PyClical.clifford . . . . .	94
PyClical.index_set . . . . .	179
Clifford	
PyClical.clifford . . . . .	94
glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T > . . . . .	113
glucat::clifford_algebra< Scalar_T, index_set< LO, HI >, framed_multi< Scalar_T, LO, HI, Tune_P > > . . . . .	113
glucat::framed_multi< Scalar_T, LO, HI, Tune_P > . . . . .	138
glucat::clifford_algebra< Scalar_T, index_set< LO, HI >, matrix_multi< Scalar_T, LO, HI, Tune_P > > . . . . .	113
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P > . . . . .	191
glucat::compare_types< LHS_T, RHS_T > . . . . .	126
glucat::compare_types< T, T > . . . . .	126
glucat::control_t . . . . .	127
glucat::CTAssertion< bool > . . . . .	132
glucat::CTAssertion< true > . . . . .	132
glucat::numeric_traits< Scalar_T >::demoted<> . . . . .	132
glucat::matrix::eig_genus< Matrix_T > . . . . .	133
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::hash_size_t . . . . .	162
glucat::index_set_hash< LO, HI > . . . . .	189
IndexSet	
PyClical.index_set . . . . .	179
inline	
PyClical.clifford . . . . .	94
PyClical.index_set . . . . .	179
logic_error	
glucat::glucat_error . . . . .	160
glucat::error< Class_T > . . . . .	135
map	
glucat::basis_table< Scalar_T, LO, HI, Matrix_T > . . . . .	91
glucat::gen::generator_table< Matrix_T > . . . . .	155
numeric_limits	
std::numeric_limits< glucat::framed_multi< Scalar_T, LO, HI, Tune_P > > . . . . .	208

std::numeric_limits< glucat::matrix_multi< Scalar_T, LO, HI, Tune_P > > . . . . .	209
glucat::numeric_traits< Scalar_T > . . . . .	210
obj	
PyClical.clifford . . . . .	94
PyClical.index_set . . . . .	179
pade::pade_log_denom< Scalar_T > . . . . .	223
pade::pade_log_denom< dd_real > . . . . .	224
pade::pade_log_denom< float > . . . . .	225
pade::pade_log_denom< long double > . . . . .	227
pade::pade_log_denom< qd_real > . . . . .	228
pade::pade_log_numer< Scalar_T > . . . . .	229
pade::pade_log_numer< dd_real > . . . . .	230
pade::pade_log_numer< float > . . . . .	232
pade::pade_log_numer< long double > . . . . .	233
pade::pade_log_numer< qd_real > . . . . .	234
pade::pade_sqrt_denom< Scalar_T > . . . . .	235
pade::pade_sqrt_denom< dd_real > . . . . .	236
pade::pade_sqrt_denom< float > . . . . .	238
pade::pade_sqrt_denom< long double > . . . . .	239
pade::pade_sqrt_denom< qd_real > . . . . .	240
pade::pade_sqrt_numer< Scalar_T > . . . . .	241
pade::pade_sqrt_numer< dd_real > . . . . .	242
pade::pade_sqrt_numer< float > . . . . .	244
pade::pade_sqrt_numer< long double > . . . . .	245
pade::pade_sqrt_numer< qd_real > . . . . .	246
pair	
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term . . . . .	261
glucat::numeric_traits< Scalar_T >::promoted<> . . . . .	247
glucat::random_generator< Scalar_T > . . . . .	249
glucat::index_set< LO, HI >::reference . . . . .	253
glucat::sorted_range< Map_T, Sorted_Map_T > . . . . .	257
glucat::sorted_range< Sorted_Map_T, Sorted_Map_T > . . . . .	259
toClifford	
PyClical.clifford . . . . .	94
toIndexSet	
PyClical.index_set . . . . .	179
unordered_map	
glucat::framed_multi< Scalar_T, LO, HI, Tune_P > . . . . .	138



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">glucat::basis_table&lt; Scalar_T, LO, HI, Matrix_T &gt;</a>	91
Table of basis elements used as a cache by basis_element()	
<a href="#">glucat::bool_to_type&lt; truth_value &gt;</a>	93
Bool to type	
<a href="#">PyClical.clifford</a>	94
<a href="#">glucat::clifford_algebra&lt; Scalar_T, Index_Set_T, Multivector_T &gt;</a>	
Clifford_algebra<> declares the operations of a Clifford algebra	113
<a href="#">glucat::compare_types&lt; LHS_T, RHS_T &gt;</a>	
Type comparison	126
<a href="#">glucat::compare_types&lt; T, T &gt;</a>	126
<a href="#">glucat::control_t</a>	
Parameters to control tests	127
<a href="#">glucat::CTAssertion&lt; bool &gt;</a>	
Compile time assertion	132
<a href="#">glucat::CTAssertion&lt; true &gt;</a>	132
<a href="#">glucat::numeric_traits&lt; Scalar_T &gt;::demoted&lt;&gt;</a>	
Demoted type for long double	132
<a href="#">glucat::matrix::eig_genus&lt; Matrix_T &gt;</a>	
Structure containing classification of eigenvalues	133
<a href="#">glucat::error&lt; Class_T &gt;</a>	
Specific exception class	135
<a href="#">glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt;</a>	
A framed_multi<Scalar_T,LO,HI,Tune_P> is a framed approximation to a multivector	138
<a href="#">glucat::gen::generator_table&lt; Matrix_T &gt;</a>	
Table of generators for specific signatures	155
<a href="#">glucat::glucat_error</a>	
Abstract exception class	160
<a href="#">glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt;::hash_size_t</a>	162
<a href="#">glucat::index_set&lt; LO, HI &gt;</a>	
Index set class based on std::bitset<> in Gnu standard C++ library	164
<a href="#">PyClical.index_set</a>	179
<a href="#">glucat::index_set_hash&lt; LO, HI &gt;</a>	189
<a href="#">glucat::matrix_multi&lt; Scalar_T, LO, HI, Tune_P &gt;</a>	
A matrix_multi<Scalar_T,LO,HI,Tune_P> is a matrix approximation to a multivector	191
<a href="#">std::numeric_limits&lt; glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt; &gt;</a>	
Numeric limits for framed_multi inherit limits for the corresponding scalar type	208

<code>std::numeric_limits&lt; glucat::matrix_multi&lt; Scalar_T, LO, HI, Tune_P &gt; &gt;</code>	
Numeric limits for <code>matrix_multi</code> inherit limits for the corresponding scalar type . . . . .	209
<code>glucat::numeric_traits&lt; Scalar_T &gt;</code>	
Extra traits which extend numeric limits . . . . .	210
<code>pade::pade_log_denom&lt; Scalar_T &gt;</code>	
Coefficients of denominator polynomials of Pade approximations produced by <code>Pade1(log(1+x),x,n,n)</code>	
223	
<code>pade::pade_log_denom&lt; dd_real &gt;</code> . . . . .	224
<code>pade::pade_log_denom&lt; float &gt;</code> . . . . .	225
<code>pade::pade_log_denom&lt; long double &gt;</code> . . . . .	227
<code>pade::pade_log_denom&lt; qd_real &gt;</code> . . . . .	228
<code>pade::pade_log_numer&lt; Scalar_T &gt;</code>	
Coefficients of numerator polynomials of Pade approximations produced by <code>Pade1(log(1+x),x,n,n)</code>	
229	
<code>pade::pade_log_numer&lt; dd_real &gt;</code> . . . . .	230
<code>pade::pade_log_numer&lt; float &gt;</code> . . . . .	232
<code>pade::pade_log_numer&lt; long double &gt;</code> . . . . .	233
<code>pade::pade_log_numer&lt; qd_real &gt;</code> . . . . .	234
<code>pade::pade_sqrt_denom&lt; Scalar_T &gt;</code>	
Coefficients of denominator polynomials of Pade approximations produced by <code>Pade1(sqrt(1+x),x,n,n)</code>	
235	
<code>pade::pade_sqrt_denom&lt; dd_real &gt;</code> . . . . .	236
<code>pade::pade_sqrt_denom&lt; float &gt;</code> . . . . .	238
<code>pade::pade_sqrt_denom&lt; long double &gt;</code> . . . . .	239
<code>pade::pade_sqrt_denom&lt; qd_real &gt;</code> . . . . .	240
<code>pade::pade_sqrt_numer&lt; Scalar_T &gt;</code>	
Coefficients of numerator polynomials of Pade approximations produced by <code>Pade1(sqrt(1+x),x,n,n)</code>	
241	
<code>pade::pade_sqrt_numer&lt; dd_real &gt;</code> . . . . .	242
<code>pade::pade_sqrt_numer&lt; float &gt;</code> . . . . .	244
<code>pade::pade_sqrt_numer&lt; long double &gt;</code> . . . . .	245
<code>pade::pade_sqrt_numer&lt; qd_real &gt;</code> . . . . .	246
<code>glucat::numeric_traits&lt; Scalar_T &gt;::promoted&lt;&gt;</code>	
Extra traits which extend numeric limits . . . . .	247
<code>glucat::random_generator&lt; Scalar_T &gt;</code>	
Random number generator with single instance per <code>Scalar_T</code> . . . . .	249
<code>glucat::index_set&lt; LO, HI &gt;::reference</code>	
Index set member reference . . . . .	253
<code>glucat::sorted_range&lt; Map_T, Sorted_Map_T &gt;</code>	
Sorted range for use with output . . . . .	257
<code>glucat::sorted_range&lt; Sorted_Map_T, Sorted_Map_T &gt;</code> . . . . .	259
<code>glucat::framed_multi&lt; Scalar_T, LO, HI, Tune_P &gt;::var_term</code>	
Variable term . . . . .	261

## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

glucat/clifford_algebra.h	265
glucat/clifford_algebra_imp.h	272
glucat/errors.h	279
glucat/errors_imp.h	280
glucat/framed_multi.h	281
glucat/framed_multi_imp.h	283
glucat/generation.h	286
glucat/generation_imp.h	287
glucat/global.h	288
glucat/glucat.h	290
glucat/glucat_config.h	291
glucat/glucat_imp.h	295
glucat/index_set.h	295
glucat/index_set_imp.h	297
glucat/long_double.h	298
glucat/matrix.h	299
glucat/matrix_imp.h	301
glucat/matrix_multi.h	303
glucat/matrix_multi_imp.h	306
glucat/portability.h	309
glucat/promotion.h	311
glucat/qd.h	312
glucat/random.h	312
glucat/scalar.h	313
glucat/scalar_imp.h	315
glucat/tuning.h	316
pyclical/glucat.pxd	318
pyclical/PyClical.h	318
pyclical/PyClical.pxd	322
pyclical/PyClical.pyx	322
pyclical/PyClical_nocython.cpp	323
test/control.h	324
test/driver.h	325
test/timing.h	325
test/try_catch.h	326
test/tuning.h	317



## Chapter 5

# Namespace Documentation

### 5.1 cga3 Namespace Reference

Definitions for 3D Conformal Geometric Algebra [DL].

#### Functions

- `template<typename Multivector_T >`  
`Multivector_T cga3 (const Multivector_T &x)`  
*Convert Euclidean 3D vector to Conformal Geometric Algebra null vector [DL (10.50)].*
- `template<typename Multivector_T >`  
`Multivector_T cga3std (const Multivector_T &X)`  
*Convert CGA3 null vector to standard Conformal Geometric Algebra null vector [DL (10.52)].*
- `template<typename Multivector_T >`  
`Multivector_T agc3 (const Multivector_T &X)`  
*Convert CGA3 null vector to Euclidean 3D vector [DL (10.50)].*

#### 5.1.1 Detailed Description

Definitions for 3D Conformal Geometric Algebra [DL].

#### 5.1.2 Function Documentation

##### 5.1.2.1 agc3()

```
template<typename Multivector_T >
Multivector_T cga3::agc3 (
    const Multivector_T & X ) [inline]
```

Convert CGA3 null vector to Euclidean 3D vector [DL (10.50)].

Definition at line 126 of file PyClical.h.

References `cga3std()`, `PyClical::cl`, and `PyClical::ist`.

### 5.1.2.2 cga3()

```
template<typename Multivector_T >
Multivector_T cga3::cga3 (
    const Multivector_T & x ) [inline]
```

Convert Euclidean 3D vector to Conformal Geometric Algebra null vector [DL (10.50)].

Definition at line 103 of file PyClical.h.

References PyClical::cl, PyClical::ist, and PyClical::ninf3.

### 5.1.2.3 cga3std()

```
template<typename Multivector_T >
Multivector_T cga3::cga3std (
    const Multivector_T & X ) [inline]
```

Convert CGA3 null vector to standard Conformal Geometric Algebra null vector [DL (10.52)].

Definition at line 114 of file PyClical.h.

References PyClical::cl, PyClical::ist, and PyClical::ninf3.

Referenced by agc3().

## 5.2 glucat Namespace Reference

### Namespaces

- [gen](#)
- [matrix](#)
- [timing](#)

### Classes

- class [basis\\_table](#)  
*Table of basis elements used as a cache by basis\_element()*
- class [bool\\_to\\_type](#)  
*Bool to type.*
- class [clifford\\_algebra](#)  
*clifford\_algebra<> declares the operations of a Clifford algebra*
- class [compare\\_types](#)  
*Type comparison.*
- class [compare\\_types< T, T >](#)
- class [control\\_t](#)  
*Parameters to control tests.*
- struct [CTAssertion](#)

*Compile time assertion.*

- struct [CTAssertion< true >](#)
- class [error](#)

*Specific exception class.*

- class [framed\\_multi](#)

*A framed\_multi<Scalar\_T,LO,HI,Tune\_P> is a framed approximation to a multivector.*

- class [glucat\\_error](#)

*Abstract exception class.*

- class [index\\_set](#)

*Index set class based on std::bitset<> in Gnu standard C++ library.*

- class [index\\_set\\_hash](#)

- class [matrix\\_multi](#)

*A matrix\_multi<Scalar\_T,LO,HI,Tune\_P> is a matrix approximation to a multivector.*

- class [numeric\\_traits](#)

*Extra traits which extend numeric limits.*

- class [random\\_generator](#)

*Random number generator with single instance per Scalar\_T.*

- class [sorted\\_range](#)

*Sorted range for use with output.*

- class [sorted\\_range< Sorted\\_Map\\_T, Sorted\\_Map\\_T >](#)

## Typedefs

- using [index\\_t](#) = int

*Size of index\_t should be enough to represent LO, HI.*

- using [set\\_value\\_t](#) = unsigned long

*Size of set\_value\_t should be enough to contain index\_set<LO,HI>*

- typedef int(\* [intfn](#)) ()

*For exception catching: pointer to function returning int.*

- typedef int(\* [intintfn](#)) (int)

*For exception catching: pointer to function of int returning int.*

- using [tuning\\_slow](#) = tuning< [Tuning\\_Slow\\_Mult\\_Matrix\\_Threshold](#), [Tuning\\_Default\\_Div\\_Max\\_Steps](#), [Tuning\\_Default\\_CR\\_Sqrt\\_Max\\_Steps](#), [Tuning\\_Default\\_DB\\_Sqrt\\_Max\\_Steps](#), [Tuning\\_Default\\_Log\\_Max\\_Outer\\_Steps](#), [Tuning\\_Default\\_Log\\_Max\\_Inner\\_Steps](#), [Tuning\\_Slow\\_Basis\\_Max\\_Count](#), [Tuning\\_Slow\\_Fast\\_Size\\_Threshold](#), [Tuning\\_Slow\\_Inv\\_Fast\\_Dim\\_Threshold](#), [Tuning\\_Slow\\_Products\\_Size\\_Threshold](#), [Tuning\\_Default\\_Denom\\_Different\\_Bits](#), [Tuning\\_Default\\_Extra\\_Different\\_Bits](#), [Tuning\\_Default\\_Function\\_Precision](#) >
- using [tuning\\_naive](#) = tuning< [Tuning\\_Naive\\_Mult\\_Matrix\\_Threshold](#), [Tuning\\_Default\\_Div\\_Max\\_Steps](#), [Tuning\\_Default\\_CR\\_Sqrt\\_Max\\_Steps](#), [Tuning\\_Default\\_DB\\_Sqrt\\_Max\\_Steps](#), [Tuning\\_Default\\_Log\\_Max\\_Outer\\_Steps](#), [Tuning\\_Default\\_Log\\_Max\\_Inner\\_Steps](#), [Tuning\\_Naive\\_Basis\\_Max\\_Count](#), [Tuning\\_Naive\\_Fast\\_Size\\_Threshold](#), [Tuning\\_Naive\\_Inv\\_Fast\\_Dim\\_Threshold](#), [Tuning\\_Default\\_Products\\_Size\\_Threshold](#), [Tuning\\_Default\\_Denom\\_Different\\_Bits](#), [Tuning\\_Default\\_Extra\\_Different\\_Bits](#), [Tuning\\_Default\\_Function\\_Precision](#) >
- using [tuning\\_fast](#) = tuning< [Tuning\\_Fast\\_Mult\\_Matrix\\_Threshold](#), [Tuning\\_Fast\\_Div\\_Max\\_Steps](#), [Tuning\\_Fast\\_CR\\_Sqrt\\_Max\\_Steps](#), [Tuning\\_Fast\\_DB\\_Sqrt\\_Max\\_Steps](#), [Tuning\\_Fast\\_Log\\_Max\\_Outer\\_Steps](#), [Tuning\\_Fast\\_Log\\_Max\\_Inner\\_Steps](#), [Tuning\\_Fast\\_Basis\\_Max\\_Count](#), [Tuning\\_Fast\\_Fast\\_Size\\_Threshold](#), [Tuning\\_Fast\\_Inv\\_Fast\\_Dim\\_Threshold](#), [Tuning\\_Fast\\_Products\\_Size\\_Threshold](#), [Tuning\\_Default\\_Denom\\_Different\\_Bits](#), [Tuning\\_Default\\_Extra\\_Different\\_Bits](#), [Tuning\\_Default\\_Function\\_Precision](#) >

## Functions

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator!= (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> bool`  
*Test for inequality of multivectors.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator!= (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const Scalar_T &scr) -> bool`  
*Test for inequality of multivector and scalar.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator!= (const Scalar_T &scr, const Multivector< Scalar_T, LO, HI, Tune_P > &rhs) -> bool`  
*Test for inequality of scalar and multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto error_squared_tol (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Quadratic norm error tolerance relative to a specific multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto error_squared (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs, const Scalar_T threshold) -> Scalar_T`  
*Relative or absolute error using the quadratic norm.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto approx_equal (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs, const Scalar_T threshold, const Scalar_T tolerance) -> bool`  
*Test for approximate equality of multivectors.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto approx_equal (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> bool`  
*Test for approximate equality of multivectors.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator+ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const Scalar_T &scr) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Geometric sum of multivector and scalar.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator+ (const Scalar_T &scr, const Multivector< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Geometric sum of scalar and multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator+ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Geometric sum.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator- (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const Scalar_T &scr) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Geometric difference of multivector and scalar.*





- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator/ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Geometric quotient.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator| (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Transformation via twisted adjoint action.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto inv (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Geometric multiplicative inverse.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, int rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Integer power of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Multivector power of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto outer_pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, int rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Outer product power of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto scalar (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Scalar part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto real (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Real part: synonym for scalar part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto imag (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Imaginary part: deprecated (always 0)*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto pure (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Pure part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto even (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Even part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto odd (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Odd part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto vector_part (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const std::vector< Scalar_T >`

*Vector part of multivector, as a vector\_t with respect to frame()*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto involute (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Main involution, each {i} is replaced by -{i} in each term, eg. {1}\*{2} -> (-{2})\*(-{1})*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto reverse (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Reversion, eg. {1}\*{2} -> {2}\*{1}.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto conj (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Conjugation, rev o invo == invo o rev.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto quad (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Scalar\_T quadratic form == (rev(x)\*x)(0)*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto norm (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Scalar\_T norm == sum of norm of coordinates.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto abs (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Absolute value == sqrt(norm)*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto max_abs (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Maximum of absolute values of components of multivector: multivector infinity norm.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto complexifier (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of -1 which commutes with all members of the frame of the given multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto elliptic (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto sqrt (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto sqrt (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto clifford\_exp (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Exponential of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto log (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto log (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto cos (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Cosine of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto cos (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto acos (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse cosine of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto acos (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto cosh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Hyperbolic cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto acosh (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse hyperbolic cosine of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto acosh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse hyperbolic cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >  
auto sin (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Sine of multivector with specified complexifier.*

- [illegible]

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto atanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inverse hyperbolic tangent of multivector with specified complexifier.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto atanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inverse hyperbolic tangent of multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator & (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inner product.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static void check\_complex (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false)`  
*Check that i is a valid complexifier for val.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator\* (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> const framed\_multi< Scalar_T, LO, HI, Tune_P >`  
*Geometric product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator^ (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> const framed\_multi< Scalar_T, LO, HI, Tune_P >`  
*Outer product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator & (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> const framed\_multi< Scalar_T, LO, HI, Tune_P >`  
*Inner product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator% (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> const framed\_multi< Scalar_T, LO, HI, Tune_P >`  
*Left contraction.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto star (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> Scalar_T`  
*Hestenes scalar product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator/ (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> const framed\_multi< Scalar_T, LO, HI, Tune_P >`  
*Geometric quotient.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator| (const framed\_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed\_multi< Scalar_T, LO, HI, Tune_P > &rhs) -> const framed\_multi< Scalar_T, LO, HI, Tune_P >`  
*Transformation via twisted adjoint action.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator>> (std::istream &s, framed\_multi< Scalar_T, LO, HI, Tune_P > &val) -> std::istream &`  
*Read multivector from input.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator<< (std::ostream &os, const framed\_multi< Scalar_T, LO, HI, Tune_P > &val) -> std::ostream &`  
*Write multivector to output.*

- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`auto operator<< (std::ostream &os, const std::pair< const index_set< LO, HI >, Scalar_T > &term) ->`  
`std::ostream &`  
*Write term to output.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto exp (const framed_multi< Scalar_T, LO, HI, Tune_P > &val) -> const framed_multi< Scalar_T, LO, HI,`  
`Tune_P >`  
*Exponential of multivector.*
- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`static auto crd_of_mult (const std::pair< const index_set< LO, HI >, Scalar_T > &lhs, const std::pair< const`  
`index_set< LO, HI >, Scalar_T > &rhs) -> Scalar_T`  
*Coordinate of product of terms.*
- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`auto operator* (const std::pair< const index_set< LO, HI >, Scalar_T > &lhs, const std::pair< const`  
`index_set< LO, HI >, Scalar_T > &rhs) -> const std::pair< const index_set< LO, HI >, Scalar_T >`  
*Product of terms.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto sqrt (const framed_multi< Scalar_T, LO, HI, Tune_P > &val, const framed_multi< Scalar_T, LO, HI,`  
`Tune_P > &i, bool prechecked) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Square root of multivector with specified complexifier.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto log (const framed_multi< Scalar_T, LO, HI, Tune_P > &val, const framed_multi< Scalar_T, LO, HI,`  
`Tune_P > &i, bool prechecked) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Natural logarithm of multivector with specified complexifier.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator & (const framed_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed_multi< Scalar_T, LO,`  
`HI, Tune_P > &rhs) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Inner product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`static auto crd_of_mult (const std::pair< const index_set< LO, HI >, Scalar_T > &lhs, const std::pair< const`  
`index_set< LO, HI >, Scalar_T > &rhs) -> Scalar_T`  
*Coordinate of product of terms.*
- `_GLUCAT_CTAssert (std::numeric_limits< unsigned char >::radix==2, CannotDetermineBitsPerChar) const`  
`index_t BITS_PER_CHAR`  
*If radix of unsigned char is not 2, we can't easily determine number of bits from sizeof.*
- `_GLUCAT_CTAssert (_GLUCAT_BITS_PER_ULONG==BITS_PER_SET_VALUE, BitsPerULongDoes←`  
`NotMatchSetValueT) const index_t DEFAULT_LO`  
*Default lowest index in an index set.*
- `template<typename LHS_T , typename RHS_T >`  
`auto pos_mod (LHS_T lhs, RHS_T rhs) -> LHS_T`  
*Modulo function which works reliably for lhs < 0.*
- `template<const index_t LO, const index_t HI>`  
`auto operator^ (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO,`  
`HI >`  
*Symmetric set difference: exclusive or.*
- `template<const index_t LO, const index_t HI>`  
`auto operator & (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set<`  
`LO, HI >`  
*Set intersection: and.*
- `template<const index_t LO, const index_t HI>`  
`auto operator| (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO,`  
`HI >`  
*Set union: or.*



- `template<const index_t LO, const index_t HI>`  
`auto compare (const index_set< LO, HI > &a, const index_set< LO, HI > &b) -> int`  
*"lexicographic compare" eg. {3,4,5} is less than {3,7,8}*
- `_GLUCAT_CTAssert (sizeof(set_value_t) >= sizeof(std::bitset< DEFAULT_HI-DEFAULT_LO >), Default←`  
`_index_set_too_big_for_value) template< const index_t LO`  
*Size of set\_value\_t should be enough to contain bitset<DEFAULT\_HI-DEFAULT\_LO>*
- `const index_t HI auto operator<< (std::ostream &os, const index_set< LO, HI > &ist) -> std::ostream &`  
*Write out index set.*
- `template<const index_t LO, const index_t HI>`  
`auto operator>> (std::istream &s, index_set< LO, HI > &ist) -> std::istream &`  
*Read in index set.*
- `auto sign_of_square (index_t j) -> int`  
*Square of generator {j}.*
- `template<const index_t LO, const index_t HI>`  
`auto min_neg (const index_set< LO, HI > &ist) -> index_t`  
*Minimum negative index, or 0 if none.*
- `template<const index_t LO, const index_t HI>`  
`auto max_pos (const index_set< LO, HI > &ist) -> index_t`  
*Maximum positive index, or 0 if none.*
- `template<const index_t LO, const index_t HI>`  
`auto operator & (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set<`  
`LO, HI >`  
*Set intersection: and.*
- `static auto inverse_reversed_gray (unsigned long x) -> unsigned long`  
*Inverse reversed Gray code.*
- `static auto inverse_gray (unsigned long x) -> unsigned long`  
*Inverse Gray code.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator* (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO, HI,`  
`Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Geometric product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator^ (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Outer product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator & (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Inner product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator% (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Left contraction.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto star (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO, HI,`  
`Tune_P > &rhs) -> Scalar_T`  
*Hestenes scalar product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator/ (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO, HI,`  
`Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Geometric quotient.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto operator| (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO, HI,`  
`Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`



*Transformation via twisted adjoint action.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [operator>>](#) (std::istream &s, [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val) -> std::istream &

*Read multivector from input.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [operator<<](#) (std::ostream &os, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val) -> std::ostream &

*Write multivector to output.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [reframe](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &lhs, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &rhs, [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &lhs\_reframed, [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &rhs\_reframed) -> const [index\\_set](#)< LO, HI >

*Find a common frame for operands of a binary operator.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [sqrt](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &i, bool prechecked) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Square root of multivector with specified complexifier.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [matrix\\_sqrt](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &i, const [index\\_t](#) level) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Square root of multivector with specified complexifier.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [log](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &i, bool prechecked) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Natural logarithm of multivector with specified complexifier.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [matrix\\_log](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &i, const [index\\_t](#) level) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Natural logarithm of multivector with specified complexifier.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [exp](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &val) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Exponential of multivector.*

- auto [offset\\_level](#) (const [index\\_t](#) p, const [index\\_t](#) q) -> [index\\_t](#)

*Determine the log2 dim corresponding to signature p, q.*

- template<typename Matrix\_Index\_T , const index\_t LO, const index\_t HI>  
static auto [folded\\_dim](#) (const [index\\_set](#)< LO, HI > &sub) -> Matrix\_Index\_T

*Determine the matrix dimension of the fold of a subalgebra.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [operator &](#) (const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &lhs, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Inner product.*

- template<typename Multivector\_T , typename Matrix\_T , typename Basis\_Matrix\_T >  
static auto [fast](#) (const Matrix\_T &X, [index\\_t](#) level) -> Multivector\_T

*Inverse generalized Fast Fourier Transform.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P , const size\_t Size>  
static auto [pade\\_approx](#) (const std::array< Scalar\_T, Size > &numer, const std::array< Scalar\_T, Size > &denom, const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &X) -> const [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P >

*Pade' approximation.*

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
static void [db\\_step](#) ([matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &M, [matrix\\_multi](#)< Scalar\_T, LO, HI, Tune\_P > &Y)

*Single step of product form of Denman-Beavers square root iteration.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto db\_sqrt (const matrix\_multi< Scalar_T, LO, HI, Tune_P > &val, Scalar_T norm_tol=std::pow(std::numeric_limits< Scalar_T >::epsilon(), 4)) -> const matrix\_multi< Scalar_T, LO, HI, Tune_P >`  
*Product form of Denman-Beavers square root iteration.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto cr\_sqrt (const matrix\_multi< Scalar_T, LO, HI, Tune_P > &val, Scalar_T norm_Y_tol=std::pow(std::numeric_limits< Scalar_T >::epsilon(), 1)) -> const matrix\_multi< Scalar_T, LO, HI, Tune_P >`  
*Cyclic reduction square root iteration.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto pade\_log (const matrix\_multi< Scalar_T, LO, HI, Tune_P > &val) -> const matrix\_multi< Scalar_T, LO, HI, Tune_P >`  
*Pade' approximation of log.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto cascade\_log (const matrix\_multi< Scalar_T, LO, HI, Tune_P > &val) -> const matrix\_multi< Scalar_T, LO, HI, Tune_P >`  
*Incomplete square root cascade and Pade' approximation of log.*
- `template<typename Scalar_T >`  
`auto log2 (const Scalar_T &x) -> Scalar_T`  
*Log base 2 of scalar.*
- `template<typename Scalar_T >`  
`auto to\_promote (const Scalar_T &val) -> typename numeric\_traits< Scalar_T >::promoted::type`  
*Cast to promote.*
- `template<typename Scalar_T >`  
`auto to\_demote (const Scalar_T &val) -> typename numeric\_traits< Scalar_T >::demoted::type`  
*Cast to demote.*
- `int try\_catch (intfn f)`  
*Exception catching for functions returning int.*
- `int try\_catch (intintfn f, int arg)`  
*Exception catching for functions of int returning int.*

## Variables

- `const double MS\_PER\_S = 1000.0`  
*Timing constant: deprecated here - moved to [test/timing.h](#).*
- `const index\_t BITS\_PER\_SET\_VALUE = std::numeric_limits<set\_value\_t>::digits`  
*Number of bits in [set\\_value\\_t](#).*
- `const index\_t DEFAULT\_HI = index\_t(BITS\_PER\_SET\_VALUE / 2)`  
*Default highest index in an index set.*
- `static const long double l\_pi = 3.1415926535897932384626433832795029L`
- `static const long double l\_ln2 = 0.6931471805599453094172321214581766L`
- `const unsigned int Tuning\_Int\_Digits = std::numeric_limits<int>::digits`
- `const unsigned int Tuning\_Max\_Threshold = 1 << Tuning\_Int\_Digits`
- `const unsigned int Tuning\_Slow\_Mult\_Matrix\_Threshold = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Slow\_Basis\_Max\_Count = 0`
- `const unsigned int Tuning\_Slow\_Fast\_Size\_Threshold = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Slow\_Inv\_Fast\_Dim\_Threshold = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Slow\_Products\_Size\_Threshold = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Naive\_Mult\_Matrix\_Threshold = 0`
- `const unsigned int Tuning\_Naive\_Basis\_Max\_Count = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Naive\_Fast\_Size\_Threshold = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Naive\_Inv\_Fast\_Dim\_Threshold = Tuning\_Max\_Threshold`
- `const unsigned int Tuning\_Fast\_Mult\_Matrix\_Threshold = 0`

- const unsigned int [Tuning\\_Fast\\_Div\\_Max\\_Steps](#) = 0
- const unsigned int [Tuning\\_Fast\\_CR\\_Sqrt\\_Max\\_Steps](#) = 256
- const unsigned int [Tuning\\_Fast\\_DB\\_Sqrt\\_Max\\_Steps](#) = 256
- const unsigned int [Tuning\\_Fast\\_Log\\_Max\\_Outer\\_Steps](#) = 16
- const unsigned int [Tuning\\_Fast\\_Log\\_Max\\_Inner\\_Steps](#) = 8
- const unsigned int [Tuning\\_Fast\\_Basis\\_Max\\_Count](#) = 1
- const unsigned int [Tuning\\_Fast\\_Fast\\_Size\\_Threshold](#) = 0
- const unsigned int [Tuning\\_Fast\\_Inv\\_Fast\\_Dim\\_Threshold](#) = 0
- const unsigned int [Tuning\\_Fast\\_Products\\_Size\\_Threshold](#) = 0

### 5.2.1 Typedef Documentation

#### 5.2.1.1 `index_t`

using `glucat::index_t` = typedef int

Size of `index_t` should be enough to represent LO, HI.

Definition at line 77 of file `global.h`.

#### 5.2.1.2 `intfn`

typedef int(\* `glucat::intfn`) ()

For exception catching: pointer to function returning int.

Definition at line 37 of file `try_catch.h`.

#### 5.2.1.3 `intintfn`

typedef int(\* `glucat::intintfn`) (int)

For exception catching: pointer to function of int returning int.

Definition at line 40 of file `try_catch.h`.

#### 5.2.1.4 `set_value_t`

using `glucat::set_value_t` = typedef unsigned long

Size of `set_value_t` should be enough to contain `index_set<LO,HI>`

Definition at line 79 of file `global.h`.

### 5.2.1.5 tuning\_fast

```
using glucat::tuning_fast = typedef tuning < Tuning_Fast_Mult_Matrix_Threshold, Tuning_Fast_Div_Max_Steps,
Tuning_Fast_CR_Sqrt_Max_Steps, Tuning_Fast_DB_Sqrt_Max_Steps, Tuning_Fast_Log_Max_Outer_Steps,
Tuning_Fast_Log_Max_Inner_Steps, Tuning_Fast_Basis_Max_Count, Tuning_Fast_Fast_Size_Threshold,
Tuning_Fast_Inv_Fast_Dim_Threshold, Tuning_Fast_Products_Size_Threshold, Tuning_Default_↔
Denom_Different_Bits, Tuning_Default_Extra_Different_Bits, Tuning_Default_Function_Precision
>
```

Definition at line 112 of file tuning.h.

### 5.2.1.6 tuning\_naive

```
using glucat::tuning_naive = typedef tuning < Tuning_Naive_Mult_Matrix_Threshold, Tuning_↔
_Default_Div_Max_Steps, Tuning_Default_CR_Sqrt_Max_Steps, Tuning_Default_DB_Sqrt_Max_Steps,
Tuning_Default_Log_Max_Outer_Steps, Tuning_Default_Log_Max_Inner_Steps, Tuning_Naive_Basis_Max_Count,
Tuning_Naive_Fast_Size_Threshold, Tuning_Naive_Inv_Fast_Dim_Threshold, Tuning_Default_Products_↔
_Size_Threshold, Tuning_Default_Denom_Different_Bits, Tuning_Default_Extra_Different_Bits,
Tuning_Default_Function_Precision >
```

Definition at line 84 of file tuning.h.

### 5.2.1.7 tuning\_slow

```
using glucat::tuning_slow = typedef tuning < Tuning_Slow_Mult_Matrix_Threshold, Tuning_↔
_Default_Div_Max_Steps, Tuning_Default_CR_Sqrt_Max_Steps, Tuning_Default_DB_Sqrt_Max_Steps,
Tuning_Default_Log_Max_Outer_Steps, Tuning_Default_Log_Max_Inner_Steps, Tuning_Slow_Basis_Max_Count,
Tuning_Slow_Fast_Size_Threshold, Tuning_Slow_Inv_Fast_Dim_Threshold, Tuning_Slow_Products_Size_Threshold,
Tuning_Default_Denom_Different_Bits, Tuning_Default_Extra_Different_Bits, Tuning_Default_↔
Function_Precision >
```

Definition at line 62 of file tuning.h.

## 5.2.2 Function Documentation

### 5.2.2.1 \_GLUCAT\_CTAssert() [1/3]

```
glucat::_GLUCAT_CTAssert (
    std::numeric_limits< unsigned char >::radix  = =2,
    CannotDetermineBitsPerChar  ) const
```

If radix of unsigned char is not 2, we can't easily determine number of bits from sizeof.

Number of bits per char is used to determine number of bits in set\_value\_t

5.2.2.2 `_GLUCAT_CTAssert()` [2/3]

```
glucat::_GLUCAT_CTAssert (
    _GLUCAT_BITS_PER_ULONG  = BITS_PER_SET_VALUE,
    BitsPerUlongDoesNotMatchSetValueT ) const
```

Default lowest index in an index set.

5.2.2.3 `_GLUCAT_CTAssert()` [3/3]

```
glucat::_GLUCAT_CTAssert (
    sizeof(set_value_t) >= sizeof(std::bitset< DEFAULT_HI-DEFAULT_LO >) ,
    Default_index_set_too_big_for_value ) const
```

Size of `set_value_t` should be enough to contain `bitset<DEFAULT_HI-DEFAULT_LO>`

Write out index set

5.2.2.4 `abs()`

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::abs (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T [inline]
```

Absolute value == `sqrt(norm)`

Definition at line 577 of file `clifford_algebra_imp.h`.

References `glucat::numeric_traits< Scalar_T >::sqrt()`.

Referenced by `PyClical.clifford::abs()`, `acos()`, `asin()`, `clifford_to_str()`, `matrix_log()`, `matrix_sqrt()`, and `pow()`.

5.2.2.5 `acos()` [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::acos (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Inverse cosine of multivector with specified complexifier.

Definition at line 883 of file `clifford_algebra_imp.h`.

References `abs()`, `acosh()`, `check_complex()`, and `PyClical::i`.

Referenced by `glucat::numeric_traits< Scalar_T >::acos()`, and `acos()`.

**5.2.2.6** `acos()` [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::acos (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Inverse cosine of multivector.

Definition at line 903 of file `clifford_algebra_imp.h`.

References `acos()`, and `complexifier()`.

**5.2.2.7** `acosh()` [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::acosh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Inverse hyperbolic cosine of multivector with specified complexifier.

Definition at line 825 of file `clifford_algebra_imp.h`.

References `check_complex()`, `PyClical::i`, `log()`, `norm()`, and `sqrt()`.

Referenced by `acos()`, and `acosh()`.

**5.2.2.8** `acosh()` [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::acosh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Inverse hyperbolic cosine of multivector.

Definition at line 844 of file `clifford_algebra_imp.h`.

References `acosh()`, and `complexifier()`.

## 5.2.2.9 approx\_equal() [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::approx_equal (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs,
    const Scalar_T threshold,
    const Scalar_T tolerance ) -> bool [inline]
```

Test for approximate equality of multivectors.

Definition at line 154 of file clifford\_algebra\_imp.h.

References `error_squared()`, `PyClical::lhs`, `PyClical::rhs`, and `PyClical::threshold`.

Referenced by `approx_equal()`, and `matrix_sqrt()`.

## 5.2.2.10 approx\_equal() [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::approx_equal (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> bool [inline]
```

Test for approximate equality of multivectors.

Definition at line 169 of file clifford\_algebra\_imp.h.

References `approx_equal()`, `error_squared_tol()`, `PyClical::lhs`, and `PyClical::rhs`.

## 5.2.2.11 asin() [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::asin (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Inverse sine of multivector with specified complexifier.

Definition at line 988 of file clifford\_algebra\_imp.h.

References `abs()`, `asinh()`, `check_complex()`, and `PyClical::i`.

Referenced by `glucat::numeric_traits< Scalar_T >::asin()`, and `asin()`.

**5.2.2.12 asin()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::asin (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Inverse sine of multivector.

Definition at line 1008 of file clifford\_algebra\_imp.h.

References [asin\(\)](#), and [complexifier\(\)](#).

**5.2.2.13 asinh()** [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::asinh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Inverse hyperbolic sine of multivector with specified complexifier.

Definition at line 930 of file clifford\_algebra\_imp.h.

References [check\\_complex\(\)](#), [PyClical::i](#), [log\(\)](#), [norm\(\)](#), and [sqrt\(\)](#).

Referenced by [asin\(\)](#), and [asinh\(\)](#).

**5.2.2.14 asinh()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::asinh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Inverse hyperbolic sine of multivector.

Definition at line 949 of file clifford\_algebra\_imp.h.

References [asinh\(\)](#), and [complexifier\(\)](#).



**5.2.2.15 atan()** [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::atan (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Inverse tangent of multivector with specified complexifier.

Definition at line 1088 of file clifford\_algebra\_imp.h.

References `atanh()`, `check_complex()`, and `PyClical::i`.

Referenced by `glucat::numeric_traits< Scalar_T >::atan()`, and `atan()`.

**5.2.2.16 atan()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::atan (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Inverse tangent of multivector.

Definition at line 1108 of file clifford\_algebra\_imp.h.

References `atan()`, and `complexifier()`.

**5.2.2.17 atanh()** [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::atanh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Inverse hyperbolic tangent of multivector with specified complexifier.

Definition at line 1035 of file clifford\_algebra\_imp.h.

References `check_complex()`, `PyClical::i`, `log()`, and `norm()`.

Referenced by `atan()`, and `atanh()`.

**5.2.2.18 atanh()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::atanh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Inverse hyperbolic tangent of multivector.

Definition at line 1052 of file clifford\_algebra\_imp.h.

References [atanh\(\)](#), and [complexifier\(\)](#).

**5.2.2.19 cascade\_log()**

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
static auto glucat::cascade_log (
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & val ) -> const matrix\_multi<Scalar_T,LO,HI,Tune_P> [static]
```

Incomplete square root cascade and Pade' approximation of log.

Definition at line 1920 of file matrix\_multi\_imp.h.

References [db\\_step\(\)](#), [epsilon](#), [glucat::clifford\\_algebra< Scalar\\_T, index\\_set< LO, HI >, matrix\\_multi< Scalar\\_T, LO, HI, Tune\\_P > >::isnan\(\)](#), [norm\(\)](#), [pade\\_log\(\)](#), and [pow\(\)](#).

Referenced by [matrix\\_log\(\)](#).

**5.2.2.20 check\_complex()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
static void glucat::check_complex (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) [inline], [static]
```

Check that i is a valid complexifier for val.

Definition at line 652 of file clifford\_algebra\_imp.h.

References [complexifier\(\)](#), and [PyClical::i](#).

Referenced by [acos\(\)](#), [acosh\(\)](#), [asin\(\)](#), [asinh\(\)](#), [atan\(\)](#), [atanh\(\)](#), [cos\(\)](#), [log\(\)](#), [sin\(\)](#), [sqrt\(\)](#), and [tan\(\)](#).

**5.2.2.21 clifford\_exp()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::clifford_exp (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
```

Exponential of multivector.

Definition at line 690 of file clifford\_algebra\_imp.h.

References `exp()`, `log2()`, and `pow()`.

Referenced by `exp()`.

**5.2.2.22 compare()**

```
template<const index_t LO, const index_t HI>
auto glucat::compare (
    const index_set< LO, HI > & a,
    const index_set< LO, HI > & b ) -> int [inline]
```

"lexicographic compare" eg. {3,4,5} is less than {3,7,8}

Lexicographic ordering of two sets: -1 if  $a < b$ , +1 if  $a > b$ , 0 if  $a == b$ .

Definition at line 574 of file index\_set\_imp.h.

**5.2.2.23 complexifier()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::complexifier (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
```

Square root of -1 which commutes with all members of the frame of the given multivector.

Definition at line 592 of file clifford\_algebra\_imp.h.

References `pos_mod()`.

Referenced by `acos()`, `acosh()`, `asin()`, `asinh()`, `atan()`, `atanh()`, `check_complex()`, `cos()`, `elliptic()`, `log()`, `sin()`, `sqrt()`, and `tan()`.

**5.2.2.24 conj()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::conj (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Conjugation, rev o invo == invo o rev.

Definition at line 553 of file clifford\_algebra\_imp.h.

**5.2.2.25 cos()** [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::cos (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
```

Cosine of multivector with specified complexifier.

Definition at line 851 of file clifford\_algebra\_imp.h.

References `check_complex()`, `exp()`, `PyClical::i`, and `PyClical::pi`.

Referenced by `glucat::numeric_traits< Scalar_T >::cos()`, `cos()`, and `tan()`.

**5.2.2.26 cos()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::cos (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Cosine of multivector.

Definition at line 874 of file clifford\_algebra\_imp.h.

References `complexifier()`, and `cos()`.

**5.2.2.27 cosh()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::cosh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Hyperbolic cosine of multivector.

Definition at line 807 of file `clifford_algebra_imp.h`.

References `exp()`.

Referenced by `glucat::numeric_traits< Scalar_T >::cosh()`, and `tanh()`.

**5.2.2.28 cr\_sqrt()**

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
static auto glucat::cr_sqrt (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val,
    Scalar_T norm_Y_tol = std::pow(std::numeric_limits<Scalar_T>::epsilon(), 1) ) ->
const matrix_multi<Scalar_T,LO,HI,Tune_P> [static]
```

Cyclic reduction square root iteration.

Definition at line 1349 of file `matrix_multi_imp.h`.

References `glucat::numeric_traits< Scalar_T >::NaN()`, and `norm()`.

Referenced by `matrix_sqrt()`.

**5.2.2.29 crd\_of\_mult()** [1/2]

```
template<typename Scalar_T , const index_t LO, const index_t HI>
static auto glucat::crd_of_mult (
    const std::pair< const index_set< LO, HI >, Scalar_T > & lhs,
    const std::pair< const index_set< LO, HI >, Scalar_T > & rhs ) -> Scalar_T [inline], [static]
```

Coordinate of product of terms.

Referenced by `operator &()`, `operator%()`, `operator*()`, and `operator^()`.

**5.2.2.30** `crd_of_mult()` [2/2]

```
template<typename Scalar_T , const index_t LO, const index_t HI>
static auto glucat::crd_of_mult (
    const std::pair< const index\_set< LO, HI >, Scalar_T > & lhs,
    const std::pair< const index\_set< LO, HI >, Scalar_T > & rhs ) -> Scalar_T
T [inline], [static]
```

Coordinate of product of terms.

Definition at line 1709 of file `framed_multi_imp.h`.

References `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.31** `db_sqrt()`

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
static auto glucat::db_sqrt (
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & val,
    Scalar_T norm_tol = std::pow(std::numeric_limits<Scalar_T>::epsilon(), 4) ) ->
const matrix\_multi<Scalar_T,LO,HI,Tune_P> [static]
```

Product form of Denman-Beavers square root iteration.

Definition at line 1320 of file `matrix_multi_imp.h`.

References `db_step()`, and `norm()`.

Referenced by `matrix_sqrt()`.

**5.2.2.32** `db_step()`

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
static void glucat::db_step (
    matrix\_multi< Scalar_T, LO, HI, Tune_P > & M,
    matrix\_multi< Scalar_T, LO, HI, Tune_P > & Y ) [inline], [static]
```

Single step of product form of Denman-Beavers square root iteration.

Definition at line 1308 of file `matrix_multi_imp.h`.

References `inv()`.

Referenced by `cascade_log()`, and `db_sqrt()`.

### 5.2.2.33 elliptic()

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::elliptic (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T, LO, HI, Tune_P> [inline]
```

Square root of -1 which commutes with all members of the frame of the given multivector The name "elliptic" is now deprecated: use "complexifier" instead.

Definition at line 643 of file clifford\_algebra\_imp.h.

References `complexifier()`.

### 5.2.2.34 error\_squared()

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::error_squared (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs,
    const Scalar_T threshold ) -> Scalar_T [inline]
```

Relative or absolute error using the quadratic norm.

Definition at line 134 of file clifford\_algebra\_imp.h.

References `PyClical::lhs`, `norm()`, `PyClical::rhs`, and `PyClical::threshold`.

Referenced by `approx_equal()`.

### 5.2.2.35 error\_squared\_tol()

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::error_squared_tol (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T
```

Quadratic norm error tolerance relative to a specific multivector.

Definition at line 112 of file clifford\_algebra\_imp.h.

References `epsilon`, and `glucat::numeric_traits< Scalar_T >::pow()`.

Referenced by `approx_equal()`.

**5.2.2.36 even()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::even (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar↵
_T,LO,HI,Tune_P> [inline]
```

Even part.

Definition at line 513 of file clifford\_algebra\_imp.h.

**5.2.2.37 exp()** [1/2]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::exp (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & val ) -> const framed_multi<Scalar↵
_T,LO,HI,Tune_P>
```

Exponential of multivector.

Definition at line 1750 of file framed\_multi\_imp.h.

References clifford\_exp(), and scalar().

Referenced by clifford\_exp(), cos(), cosh(), glucat::numeric\_traits< Scalar\_T >::exp(), exp(), matrix\_log(), matrix↵\_sqrt(), pow(), PyClical.clifford::pow(), sin(), and sinh().

**5.2.2.38 exp()** [2/2]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::exp (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val ) -> const matrix_multi<Scalar↵
_T,LO,HI,Tune_P>
```

Exponential of multivector.

Definition at line 2086 of file matrix\_multi\_imp.h.

References clifford\_exp(), exp(), glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::isnan(), and glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar↵\_T, LO, HI, Tune\_P > >::scalar().



**5.2.2.39 fast()**

```
template<typename Multivector_T , typename Matrix_T , typename Basis_Matrix_T >
static auto glucat::fast (
    const Matrix_T & X,
    index_t level ) -> Multivector_T    [static]
```

Inverse generalized Fast Fourier Transform.

Definition at line 1027 of file matrix\_multi\_imp.h.

References glucat::matrix::signed\_perm\_nork().

**5.2.2.40 folded\_dim()**

```
template<typename Matrix_Index_T , const index_t LO, const index_t HI>
static auto glucat::folded_dim (
    const index_set< LO, HI > & sub ) -> Matrix_Index_T    [inline], [static]
```

Determine the matrix dimension of the fold of a subalgebra.

Definition at line 101 of file matrix\_multi\_imp.h.

References offset\_level().

**5.2.2.41 imag()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::imag (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T    [inline]
```

Imaginary part: deprecated (always 0)

Definition at line 497 of file clifford\_algebra\_imp.h.

Referenced by glucat::matrix::classify\_eigenvalues().

**5.2.2.42 inv()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::inv (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P>    [inline]
```

Geometric multiplicative inverse.

Definition at line 400 of file clifford\_algebra\_imp.h.

Referenced by db\_step(), and matrix\_log().

**5.2.2.43 inverse\_gray()**

```
static auto glucat::inverse_gray (
    unsigned long x ) -> unsigned long    [inline], [static]
```

Inverse Gray code.

Definition at line 863 of file index\_set\_imp.h.

Referenced by glucat::index\_set< LO, HI >::sign\_of\_mult().

**5.2.2.44 inverse\_reversed\_gray()**

```
static auto glucat::inverse_reversed_gray (
    unsigned long x ) -> unsigned long    [inline], [static]
```

Inverse reversed Gray code.

Definition at line 846 of file index\_set\_imp.h.

Referenced by glucat::index\_set< LO, HI >::sign\_of\_mult().

**5.2.2.45 involute()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::involute (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Main involution, each {i} is replaced by -{i} in each term, eg. {1}\*{2} -> (-{2})\*(-{1})

Main involution, each {i} is replaced by -{i} in each term, eg. {1} -> -{1}.

Definition at line 537 of file clifford\_algebra\_imp.h.

**5.2.2.46 log()** [1/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::log (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & i,
    bool prechecked ) -> const matrix_multi<Scalar_T,LO,HI,Tune_P>
```

Natural logarithm of multivector with specified complexifier.

Definition at line 2045 of file matrix\_multi\_imp.h.

References check\_complex(), PyClical::i, glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::isnan(), and matrix\_log().

**5.2.2.47** `log()` [2/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::log (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & val,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & i,
    bool prechecked ) -> const framed_multi<Scalar_T,LO,HI,Tune_P>
```

Natural logarithm of multivector with specified complexifier.

Definition at line 1800 of file framed\_multi\_imp.h.

References `check_complex()`, `PyClical::i`, `log()`, and `PyClical::pi`.

**5.2.2.48** `log()` [3/4]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::log (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Natural logarithm of multivector with specified complexifier.

Definition at line 791 of file clifford\_algebra\_imp.h.

References `PyClical::i`.

Referenced by `acosh()`, `asinh()`, `atanh()`, `glucat::numeric_traits< Scalar_T >::log()`, `log()`, `matrix_log()`, `glucat::numeric_traits< Scalar_T >::NaN()`, `pow()`, and `PyClical.clifford::pow()`.

**5.2.2.49** `log()` [4/4]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::log (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Natural logarithm of multivector.

Definition at line 799 of file clifford\_algebra\_imp.h.

References `complexifier()`, and `log()`.

### 5.2.2.50 log2()

```
template<typename Scalar_T >
auto glucat::log2 (
    const Scalar_T & x ) -> Scalar_T    [inline]
```

Log base 2 of scalar.

Definition at line 303 of file scalar.h.

References glucat::numeric\_traits< Scalar\_T >::log2().

Referenced by clifford\_exp().

### 5.2.2.51 matrix\_log()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_log (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & i,
    const index_t level ) -> const matrix_multi<Scalar_T,LO,HI,Tune_P>
```

Natural logarithm of multivector with specified complexifier.

Definition at line 1967 of file matrix\_multi\_imp.h.

References abs(), cascade\_log(), glucat::matrix::classify\_eigenvalues(), exp(), PyClical::i, inv(), glucat::matrix::isnan(), glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::isnan(), log(), norm(), PyClical::pi, and glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::scalar().

Referenced by log().

### 5.2.2.52 matrix\_sqrt()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_sqrt (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & i,
    const index_t level ) -> const matrix_multi<Scalar_T,LO,HI,Tune_P>
```

Square root of multivector with specified complexifier.

Definition at line 1571 of file matrix\_multi\_imp.h.

References abs(), approx\_equal(), glucat::matrix::classify\_eigenvalues(), cr\_sqrt(), db\_sqrt(), exp(), PyClical::i, glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::isnan(), norm(), pade\_approx(), pow(), glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::scalar(), and sqrt().

Referenced by sqrt().

**5.2.2.53 max\_abs()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::max_abs (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T [inline]
```

Maximum of absolute values of components of multivector: multivector infinity norm.

Definition at line 585 of file clifford\_algebra\_imp.h.

**5.2.2.54 max\_pos()**

```
template<const index_t LO, const index_t HI>
auto glucat::max_pos (
    const index_set< LO, HI > & ist ) -> index_t [inline]
```

Maximum positive index, or 0 if none.

Definition at line 977 of file index\_set\_imp.h.

References PyClical::ist.

**5.2.2.55 min\_neg()**

```
template<const index_t LO, const index_t HI>
auto glucat::min_neg (
    const index_set< LO, HI > & ist ) -> index_t [inline]
```

Minimum negative index, or 0 if none.

Definition at line 970 of file index\_set\_imp.h.

References PyClical::ist.

**5.2.2.56 norm()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::norm (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T [inline]
```

Scalar\_T norm == sum of norm of coordinates.

Definition at line 569 of file clifford\_algebra\_imp.h.

Referenced by acosh(), asinh(), atanh(), cascade\_log(), glucat::matrix::classify\_eigenvalues(), cr\_sqrt(), db\_sqrt(), error\_squared(), matrix\_log(), and matrix\_sqrt().

**5.2.2.57 odd()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::odd (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_←
_T,LO,HI,Tune_P> [inline]
```

Odd part.

Definition at line 521 of file clifford\_algebra\_imp.h.

Referenced by glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::fast().

**5.2.2.58 offset\_level()**

```
auto glucat::offset_level (
    const index_t p,
    const index_t q ) -> index_t [inline]
```

Determine the log2 dim corresponding to signature p, q.

Definition at line 86 of file matrix\_multi\_imp.h.

References pos\_mod().

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::basis\_element(), and folded\_dim().

**5.2.2.59 operator &() [1/8]**

```
template<const index_t LO, const index_t HI>
auto glucat::operator& (
    const index_set< LO, HI > & lhs,
    const index_set< LO, HI > & rhs ) -> const index_set< LO, HI > [inline]
```

Set intersection: and.

Definition at line 186 of file index\_set\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.60 operator &()** [2/8]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator& (
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix\_multi<
Scalar_T, LO, HI, Tune_P > [inline]
```

Inner product.

Definition at line 562 of file [matrix\\_multi\\_imp.h](#).

References [PyClical::lhs](#), and [PyClical::rhs](#).

**5.2.2.61 operator &()** [3/8]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator& (
    const framed\_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed\_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed\_multi<
Scalar_T, LO, HI, Tune_P >
```

Inner product.

Definition at line 495 of file [framed\\_multi\\_imp.h](#).

References [\\_GLUCAT\\_HASH\\_SIZE\\_T](#), [crd\\_of\\_mult\(\)](#), [PyClical::lhs](#), and [PyClical::rhs](#).

**5.2.2.62 operator &()** [4/8]

```
template<const index_t LO, const index_t HI>
auto glucat::operator& (
    const index\_set< LO, HI > & lhs,
    const index\_set< LO, HI > & rhs ) -> const index\_set<LO,HI> [inline]
```

Set intersection: and.

Definition at line 186 of file [index\\_set\\_imp.h](#).

References [PyClical::lhs](#), and [PyClical::rhs](#).

**5.2.2.63 operator &()** [5/8]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator& (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,
T,LO,HI,Tune_P> [inline]
```

Inner product.

Definition at line 307 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.64 operator &()** [6/8]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator& (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector< Scalar_T,
LO, HI, Tune_P > [inline]
```

Inner product.

Definition at line 307 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.65 operator &()** [7/8]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator& (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed_multi<Scalar_T,
LO,HI,Tune_P>
```

Inner product.

Definition at line 495 of file framed\_multi\_imp.h.

References \_GLUCAT\_HASH\_SIZE\_T, crd\_of\_mult(), PyClical::lhs, and PyClical::rhs.



**5.2.2.66 operator &()** [8/8]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator& (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix_multi<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Inner product.

Definition at line 562 of file matrix\_multi\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.67 operator"!="()** [1/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator!= (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> bool [inline]
```

Test for inequality of multivectors.

Definition at line 86 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.68 operator"!="()** [2/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator!= (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const Scalar_T & scr ) -> bool [inline]
```

Test for inequality of multivector and scalar.

Definition at line 94 of file clifford\_algebra\_imp.h.

References PyClical::lhs.

**5.2.2.69 operator!=()** [3/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator!= (
    const Scalar_T & scr,
    const Multivector< Scalar_T, LO, HI, Tune_P > & rhs ) -> bool [inline]
```

Test for inequality of scalar and multivector.

Definition at line 102 of file clifford\_algebra\_imp.h.

References `PyClical::rhs`.

**5.2.2.70 operator%()** [1/3]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator% (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix_multi<Scalar↵
_T,LO,HI,Tune_P> [inline]
```

Left contraction.

Definition at line 581 of file matrix\_multi\_imp.h.

References `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.71 operator%()** [2/3]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator% (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed_multi<Scalar↵
_T,LO,HI,Tune_P>
```

Left contraction.

Definition at line 597 of file framed\_multi\_imp.h.

References `_GLUCAT_HASH_SIZE_T`, `crd_of_mult()`, `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.72 operator%()** [3/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator% (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Left contraction.

Definition at line 322 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.73 operator\*()** [1/6]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator* (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix_multi<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Geometric product.

Definition at line 502 of file matrix\_multi\_imp.h.

References PyClical::lhs, reframe(), and PyClical::rhs.

**5.2.2.74 operator\*()** [2/6]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator* (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed_multi<Scalar_↵
_T,LO,HI,Tune_P>
```

Geometric product.

Definition at line 374 of file framed\_multi\_imp.h.

References \_GLUCAT\_HASH\_SIZE\_T, PyClical::lhs, and PyClical::rhs.

**5.2.2.75 operator\*()** [3/6]

```
template<typename Scalar_T , const index_t LO, const index_t HI>
auto glucat::operator* (
    const std::pair< const index_set< LO, HI >, Scalar_T > & lhs,
    const std::pair< const index_set< LO, HI >, Scalar_T > & rhs ) -> const std::pair<const index_set<LO,HI>, Scalar_T> [inline]
```

Product of terms.

Definition at line 1717 of file framed\_multi\_imp.h.

References `crd_of_mult()`, `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.76 operator\*()** [4/6]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator* (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const Scalar_T & scr ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Product of multivector and scalar.

Definition at line 251 of file clifford\_algebra\_imp.h.

References `PyClical::lhs`.

**5.2.2.77 operator\*()** [5/6]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator* (
    const Scalar_T & scr,
    const Multivector< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Product of scalar and multivector.

Definition at line 262 of file clifford\_algebra\_imp.h.

References `PyClical::rhs`.

**5.2.2.78 operator\*()** [6/6]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator* (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Geometric product.

Definition at line 277 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.79 operator+()** [1/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator+ (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const Scalar_T & scr ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Geometric sum of multivector and scalar.

Definition at line 181 of file clifford\_algebra\_imp.h.

References PyClical::lhs.

**5.2.2.80 operator+()** [2/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator+ (
    const Scalar_T & scr,
    const Multivector< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Geometric sum of scalar and multivector.

Definition at line 192 of file clifford\_algebra\_imp.h.

References PyClical::rhs.

**5.2.2.81 operator+()** [3/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator+ (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Geometric sum.

Definition at line 206 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.82 operator-()** [1/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator- (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const Scalar_T & scr ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Geometric difference of multivector and scalar.

Definition at line 217 of file clifford\_algebra\_imp.h.

References PyClical::lhs.

**5.2.2.83 operator-()** [2/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator- (
    const Scalar_T & scr,
    const Multivector< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Geometric difference of scalar and multivector.

Definition at line 228 of file clifford\_algebra\_imp.h.

References PyClical::rhs.

**5.2.2.84 operator-()** [3/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator- (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Geometric difference.

Definition at line 240 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.85 operator/()** [1/5]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator/ (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix_multi<Scalar_↵
_T,LO,HI,Tune_P>
```

Geometric quotient.

Definition at line 614 of file matrix\_multi\_imp.h.

References glucat::matrix::isnan(), PyClical::lhs, reframe(), and PyClical::rhs.

**5.2.2.86 operator/()** [2/5]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator/ (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed_multi<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Geometric quotient.

Definition at line 734 of file framed\_multi\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.87 operator/()** [3/5]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator/ (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const Scalar_T & scr ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Quotient of multivector and scalar.

Definition at line 348 of file clifford\_algebra\_imp.h.

References PyClical::lhs.

**5.2.2.88 operator/()** [4/5]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator/ (
    const Scalar_T & scr,
    const Multivector< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_←
_T,LO,HI,Tune_P> [inline]
```

Quotient of scalar and multivector.

Definition at line 359 of file clifford\_algebra\_imp.h.

References PyClical::rhs.

**5.2.2.89 operator/()** [5/5]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator/ (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_←
T,LO,HI,Tune_P> [inline]
```

Geometric quotient.

Definition at line 374 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.



**5.2.2.90 operator<<()** [1/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator<< (
    std::ostream & os,
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & val ) -> std::ostream& [inline]
```

Write multivector to output.

Definition at line 956 of file [matrix\\_multi\\_imp.h](#).

**5.2.2.91 operator<<()** [2/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator<< (
    std::ostream & os,
    const framed\_multi< Scalar_T, LO, HI, Tune_P > & val ) -> std::ostream&
```

Write multivector to output.

Definition at line 1149 of file [framed\\_multi\\_imp.h](#).

References [pow\(\)](#), [scalar\(\)](#), and [glucat::sorted\\_range< Map\\_T, Sorted\\_Map\\_T >::sorted\\_begin](#).

**5.2.2.92 operator<<()** [3/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI>
auto glucat::operator<< (
    std::ostream & os,
    const std::pair< const index\_set< LO, HI >, Scalar_T > & term ) -> std::ostream&
```

Write term to output.

Definition at line 1210 of file [framed\\_multi\\_imp.h](#).

References [pow\(\)](#), [glucat::numeric\\_traits< Scalar\\_T >::to\\_double\(\)](#), and [PyClical::tol](#).

**5.2.2.93 operator<<()** [4/4]

```
auto glucat::operator<< (
    std::ostream & os,
    const index\_set< LO, HI > & ist ) -> std::ostream&
```

Write out index set.

Definition at line 612 of file [index\\_set\\_imp.h](#).

References [PyClical::i](#), and [PyClical::ist](#).

**5.2.2.94** `operator>>()` [1/3]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator>> (
    std::istream & s,
    matrix_multi< Scalar_T, LO, HI, Tune_P > & val ) -> std::istream& [inline]
```

Read multivector from input.

Definition at line 966 of file matrix\_multi\_imp.h.

**5.2.2.95** `operator>>()` [2/3]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator>> (
    std::istream & s,
    framed_multi< Scalar_T, LO, HI, Tune_P > & val ) -> std::istream&
```

Read multivector from input.

Definition at line 1248 of file framed\_multi\_imp.h.

References PyClical::ist.

**5.2.2.96** `operator>>()` [3/3]

```
template<const index_t LO, const index_t HI>
auto glucat::operator>> (
    std::istream & s,
    index_set< LO, HI > & ist ) -> std::istream&
```

Read in index set.

Definition at line 634 of file index\_set\_imp.h.

References PyClical::i, and PyClical::ist.

**5.2.2.97** `operator^()` [1/4]

```
template<const index_t LO, const index_t HI>
auto glucat::operator^ (
    const index_set< LO, HI > & lhs,
    const index_set< LO, HI > & rhs ) -> const index_set<LO,HI> [inline]
```

Symmetric set difference: exclusive or.

Definition at line 161 of file index\_set\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.98** `operator^()` [2/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator^ (
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix\_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix\_multi<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Outer product.

Definition at line 543 of file `matrix_multi_imp.h`.

References `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.99** `operator^()` [3/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator^ (
    const framed\_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed\_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed\_multi<Scalar_↵
_T,LO,HI,Tune_P>
```

Outer product.

Definition at line 416 of file `framed_multi_imp.h`.

References `_GLUCAT_HASH_SIZE_T`, `crd_of_mult()`, `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.100** `operator^()` [4/4]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator^ (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_↵
T,LO,HI,Tune_P> [inline]
```

Outer product.

Definition at line 292 of file `clifford_algebra_imp.h`.

References `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.101 operator" | ()** [1/4]

```
template<const index_t LO, const index_t HI>
auto glucat::operator| (
    const index_set< LO, HI > & lhs,
    const index_set< LO, HI > & rhs ) -> const index_set<LO,HI> [inline]
```

Set union: or.

Definition at line 211 of file index\_set\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.102 operator" | ()** [2/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator| (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const matrix_multi<Scalar↵
_T,LO,HI,Tune_P> [inline]
```

Transformation via twisted adjoint action.

Definition at line 717 of file matrix\_multi\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.103 operator" | ()** [3/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator| (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> const framed_multi<Scalar↵
_T,LO,HI,Tune_P> [inline]
```

Transformation via twisted adjoint action.

Definition at line 760 of file framed\_multi\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.104 operator" | () [4/4]**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::operator| (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Transformation via twisted adjoint action.

Definition at line 389 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.105 outer\_pow()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::outer_pow (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    int rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
```

Outer product power of multivector.

Definition at line 470 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.106 pade\_approx()**

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P , const size_t Size>
static auto glucat::pade_approx (
    const std::array< Scalar_T, Size > & numer,
    const std::array< Scalar_T, Size > & denom,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & X ) -> const matrix_multi<Scalar_T,LO,HI,Tune_P> [inline], [static]
```

Pade' approximation.

Definition at line 1245 of file matrix\_multi\_imp.h.

Referenced by matrix\_sqrt(), and pade\_log().

**5.2.2.107** `pade_log()`

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
static auto glucat::pade_log (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val ) -> const matrix_multi<Scalar_
_T,LO,HI,Tune_P>    [static]
```

Pade' approximation of log.

Definition at line 1900 of file `matrix_multi_imp.h`.

References `glucat::clifford_algebra< Scalar_T, index_set< LO, HI >, matrix_multi< Scalar_T, LO, HI, Tune_P >>::isnan()`, and `pade_approx()`.

Referenced by `cascade_log()`.

**5.2.2.108** `pos_mod()`

```
template<typename LHS_T , typename RHS_T >
auto glucat::pos_mod (
    LHS_T lhs,
    RHS_T rhs ) -> LHS_T    [inline]
```

Modulo function which works reliably for  $lhs < 0$ .

Definition at line 117 of file `global.h`.

References `PyClical::lhs`, and `PyClical::rhs`.

Referenced by `complexifier()`, `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::fast_framed_multi()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fast_matrix_multi()`, `glucat::gen::generator_table< Matrix_T >::gen_vector()`, `offset_level()`, and `glucat::gen::generator_table< Matrix_T >::operator()`.

**5.2.2.109** `pow()` [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::pow (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    int rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
```

Integer power of multivector.

Definition at line 407 of file `clifford_algebra_imp.h`.

References `abs()`, `PyClical::lhs`, and `PyClical::rhs`.

Referenced by `cascade_log()`, `clifford_exp()`, `matrix_sqrt()`, `operator<<()`, and `glucat::numeric_traits< Scalar_T >::pow()`.

**5.2.2.110 pow()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::pow (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Multivector power of multivector.

Definition at line 446 of file clifford\_algebra\_imp.h.

References `exp()`, `PyClical::lhs`, `log()`, and `PyClical::rhs`.

**5.2.2.111 pure()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::pure (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Pure part.

Definition at line 505 of file clifford\_algebra\_imp.h.

**5.2.2.112 quad()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::quad (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T [inline]
```

Scalar\_T quadratic form == (rev(x)\*x)(0)

Definition at line 561 of file clifford\_algebra\_imp.h.

**5.2.2.113 real()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::real (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T [inline]
```

Real part: synonym for scalar part.

Definition at line 486 of file clifford\_algebra\_imp.h.

Referenced by `glucat::matrix::classify_eigenvalues()`.

**5.2.2.114** `reframe()`

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::reframe (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs,
    matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs_reframed,
    matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs_reframed ) -> const index_set<LO,HI>
[inline]
```

Find a common frame for operands of a binary operator.

Definition at line 345 of file `matrix_multi_imp.h`.

References `PyClical::lhs`, and `PyClical::rhs`.

Referenced by operator`*`(), and operator`/`().

**5.2.2.115** `reverse()`

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::reverse (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar↵
_T,LO,HI,Tune_P> [inline]
```

Reversion, eg.  $\{1\}*\{2\} \rightarrow \{2\}*\{1\}$ .

Definition at line 545 of file `clifford_algebra_imp.h`.

**5.2.2.116** `scalar()`

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::scalar (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> Scalar_T [inline]
```

Scalar part.

Definition at line 478 of file `clifford_algebra_imp.h`.

Referenced by `exp()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fast()`, and operator`<<`().



**5.2.2.117 sign\_of\_square()**

```
auto glucat::sign_of_square (
    index_t j ) -> int [inline]
```

Square of generator {j}.

Square of generator index j.

Definition at line 963 of file index\_set\_imp.h.

**5.2.2.118 sin()** [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sin (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
```

Sine of multivector with specified complexifier.

Definition at line 956 of file clifford\_algebra\_imp.h.

References `check_complex()`, `exp()`, `PyClical::i`, and `PyClical::pi`.

Referenced by `glucat::numeric_traits< Scalar_T >::sin()`, `sin()`, and `tan()`.

**5.2.2.119 sin()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sin (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Sine of multivector.

Definition at line 979 of file clifford\_algebra\_imp.h.

References `complexifier()`, and `sin()`.

**5.2.2.120 sinh()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sinh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_T,LO,HI,Tune_P> [inline]
```

Hyperbolic sine of multivector.

Definition at line 911 of file `clifford_algebra_imp.h`.

References `exp()`.

Referenced by `glucat::numeric_traits< Scalar_T >::sinh()`, and `tanh()`.

**5.2.2.121 sqrt()** [1/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sqrt (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & val,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & i,
    bool prechecked ) -> const matrix_multi<Scalar_T,LO,HI,Tune_P>
```

Square root of multivector with specified complexifier.

Definition at line 1667 of file `matrix_multi_imp.h`.

References `check_complex()`, `PyClical::i`, `glucat::clifford_algebra< Scalar_T, index_set< LO, HI >, matrix_multi< Scalar_T, LO, HI, Tune_P > >::isnan()`, and `matrix_sqrt()`.

**5.2.2.122 sqrt()** [2/4]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sqrt (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & val,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & i,
    bool prechecked ) -> const framed_multi<Scalar_T,LO,HI,Tune_P>
```

Square root of multivector with specified complexifier.

Definition at line 1727 of file `framed_multi_imp.h`.

References `check_complex()`, `PyClical::i`, and `sqrt()`.

**5.2.2.123** `sqrt()` [3/4]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sqrt (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Square root of multivector with specified complexifier.

Definition at line 675 of file `clifford_algebra_imp.h`.

References `PyClical::i`.

Referenced by `acosh()`, `asinh()`, `matrix_sqrt()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::random()`, and `sqrt()`.

**5.2.2.124** `sqrt()` [4/4]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::sqrt (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_←
_T,LO,HI,Tune_P> [inline]
```

Square root of multivector.

Definition at line 683 of file `clifford_algebra_imp.h`.

References `complexifier()`, and `sqrt()`.

**5.2.2.125** `star()` [1/3]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::star (
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const matrix_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> Scalar_T [inline]
```

Hestenes scalar product.

Definition at line 600 of file `matrix_multi_imp.h`.

References `PyClical::lhs`, and `PyClical::rhs`.

**5.2.2.126 star()** [2/3]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::star (
    const framed_multi< Scalar_T, LO, HI, Tune_P > & lhs,
    const framed_multi< Scalar_T, LO, HI, Tune_P > & rhs ) -> Scalar_T
```

Hestenes scalar product.

Definition at line 684 of file framed\_multi\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.127 star()** [3/3]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T ,
const index_t LO, const index_t HI, typename Tune_P >
auto glucat::star (
    const Multivector< Scalar_T, LO, HI, Tune_P > & lhs,
    const RHS< Scalar_T, LO, HI, Tune_P > & rhs ) -> Scalar_T [inline]
```

Hestenes scalar product.

Definition at line 337 of file clifford\_algebra\_imp.h.

References PyClical::lhs, and PyClical::rhs.

**5.2.2.128 tan()** [1/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::tan (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val,
    const Multivector< Scalar_T, LO, HI, Tune_P > & i,
    const bool prechecked = false ) -> const Multivector<Scalar_T,LO,HI,Tune_P>
[inline]
```

Tangent of multivector with specified complexifier.

Definition at line 1060 of file clifford\_algebra\_imp.h.

References check\_complex(), cos(), PyClical::i, and sin().

Referenced by glucat::numeric\_traits< Scalar\_T >::tan(), and tan().

**5.2.2.129 tan()** [2/2]

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::tan (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Tangent of multivector.

Definition at line 1079 of file clifford\_algebra\_imp.h.

References `complexifier()`, and `tan()`.

**5.2.2.130 tanh()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::tanh (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const Multivector<Scalar_↵
_T,LO,HI,Tune_P> [inline]
```

Hyperbolic tangent of multivector.

Definition at line 1016 of file clifford\_algebra\_imp.h.

References `cosh()`, and `sinh()`.

Referenced by `glucat::numeric_traits< Scalar_T >::tanh()`.

**5.2.2.131 to\_demote()**

```
template<typename Scalar_T >
auto glucat::to_demote (
    const Scalar_T & val ) -> typename numeric_traits<Scalar_T>::demoted::type [inline]
```

Cast to demote.

Definition at line 135 of file scalar\_imp.h.

References `glucat::numeric_traits< Scalar_T >::to_scalar_t()`.

**5.2.2.132 to\_promote()**

```
template<typename Scalar_T >
auto glucat::to_promote (
    const Scalar_T & val ) -> typename numeric_traits<Scalar_T>::promoted::type
[inline]
```

Cast to promote.

Definition at line 125 of file scalar\_imp.h.

References glucat::numeric\_traits< Scalar\_T >::to\_scalar\_t().

**5.2.2.133 try\_catch()** [1/2]

```
int glucat::try_catch (
    intfn f )
```

Exception catching for functions returning int.

Definition at line 49 of file try\_catch.h.

References PyClical::e().

Referenced by glucat::control\_t::call().

**5.2.2.134 try\_catch()** [2/2]

```
int glucat::try_catch (
    intintfn f,
    int arg )
```

Exception catching for functions of int returning int.

Definition at line 64 of file try\_catch.h.

References PyClical::e().

**5.2.2.135 vector\_part()**

```
template<template< typename, const index_t, const index_t, typename > class Multivector,
typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::vector_part (
    const Multivector< Scalar_T, LO, HI, Tune_P > & val ) -> const std::vector<Scalar_↵
_T> [inline]
```

Vector part of multivector, as a vector\_t with respect to frame()

Definition at line 529 of file clifford\_algebra\_imp.h.

### 5.2.3 Variable Documentation

#### 5.2.3.1 BITS\_PER\_SET\_VALUE

```
const index_t glucat::BITS_PER_SET_VALUE = std::numeric_limits<set_value_t>::digits
```

Number of bits in set\_value\_t.

Definition at line 103 of file global.h.

#### 5.2.3.2 DEFAULT\_HI

```
const index_t glucat::DEFAULT_HI = index_t(BITS_PER_SET_VALUE / 2)
```

Default highest index in an index set.

Definition at line 111 of file global.h.

#### 5.2.3.3 l\_ln2

```
const long double glucat::l_ln2 = 0.6931471805599453094172321214581766L [static]
```

Definition at line 44 of file long\_double.h.

Referenced by glucat::numeric\_traits< Scalar\_T >::ln\_2().

#### 5.2.3.4 l\_pi

```
const long double glucat::l_pi = 3.1415926535897932384626433832795029L [static]
```

Definition at line 43 of file long\_double.h.

Referenced by glucat::numeric\_traits< Scalar\_T >::pi().

#### 5.2.3.5 MS\_PER\_S

```
const double glucat::MS_PER_S = 1000.0
```

Timing constant: deprecated here - moved to [test/timing.h](#).

Definition at line 83 of file global.h.

#### 5.2.3.6 Tuning\_Fast\_Basis\_Max\_Count

```
const unsigned int glucat::Tuning_Fast_Basis_Max_Count = 1
```

Definition at line 92 of file tuning.h.

#### 5.2.3.7 Tuning\_Fast\_CR\_Sqrt\_Max\_Steps

```
const unsigned int glucat::Tuning_Fast_CR_Sqrt_Max_Steps = 256
```

Definition at line 88 of file tuning.h.

#### 5.2.3.8 Tuning\_Fast\_DB\_Sqrt\_Max\_Steps

```
const unsigned int glucat::Tuning_Fast_DB_Sqrt_Max_Steps = 256
```

Definition at line 89 of file tuning.h.

#### 5.2.3.9 Tuning\_Fast\_Div\_Max\_Steps

```
const unsigned int glucat::Tuning_Fast_Div_Max_Steps = 0
```

Definition at line 87 of file tuning.h.

#### 5.2.3.10 Tuning\_Fast\_Fast\_Size\_Threshold

```
const unsigned int glucat::Tuning_Fast_Fast_Size_Threshold = 0
```

Definition at line 93 of file tuning.h.

#### 5.2.3.11 Tuning\_Fast\_Inv\_Fast\_Dim\_Threshold

```
const unsigned int glucat::Tuning_Fast_Inv_Fast_Dim_Threshold = 0
```

Definition at line 94 of file tuning.h.



#### 5.2.3.12 Tuning\_Fast\_Log\_Max\_Inner\_Steps

```
const unsigned int glucat::Tuning_Fast_Log_Max_Inner_Steps = 8
```

Definition at line 91 of file tuning.h.

#### 5.2.3.13 Tuning\_Fast\_Log\_Max\_Outer\_Steps

```
const unsigned int glucat::Tuning_Fast_Log_Max_Outer_Steps = 16
```

Definition at line 90 of file tuning.h.

#### 5.2.3.14 Tuning\_Fast\_Mult\_Matrix\_Threshold

```
const unsigned int glucat::Tuning_Fast_Mult_Matrix_Threshold = 0
```

Definition at line 86 of file tuning.h.

#### 5.2.3.15 Tuning\_Fast\_Products\_Size\_Threshold

```
const unsigned int glucat::Tuning_Fast_Products_Size_Threshold = 0
```

Definition at line 95 of file tuning.h.

#### 5.2.3.16 Tuning\_Int\_Digits

```
const unsigned int glucat::Tuning_Int_Digits = std::numeric_limits<int>::digits
```

Definition at line 36 of file tuning.h.

#### 5.2.3.17 Tuning\_Max\_Threshold

```
const unsigned int glucat::Tuning_Max_Threshold = 1 << Tuning\_Int\_Digits
```

Definition at line 37 of file tuning.h.

#### 5.2.3.18 Tuning\_Naive\_Basis\_Max\_Count

```
const unsigned int glucat::Tuning_Naive_Basis_Max_Count = Tuning_Max_Threshold
```

Definition at line 65 of file tuning.h.

#### 5.2.3.19 Tuning\_Naive\_Fast\_Size\_Threshold

```
const unsigned int glucat::Tuning_Naive_Fast_Size_Threshold = Tuning_Max_Threshold
```

Definition at line 66 of file tuning.h.

#### 5.2.3.20 Tuning\_Naive\_Inv\_Fast\_Dim\_Threshold

```
const unsigned int glucat::Tuning_Naive_Inv_Fast_Dim_Threshold = Tuning_Max_Threshold
```

Definition at line 67 of file tuning.h.

#### 5.2.3.21 Tuning\_Naive\_Mult\_Matrix\_Threshold

```
const unsigned int glucat::Tuning_Naive_Mult_Matrix_Threshold = 0
```

Definition at line 64 of file tuning.h.

#### 5.2.3.22 Tuning\_Slow\_Basis\_Max\_Count

```
const unsigned int glucat::Tuning_Slow_Basis_Max_Count = 0
```

Definition at line 42 of file tuning.h.

#### 5.2.3.23 Tuning\_Slow\_Fast\_Size\_Threshold

```
const unsigned int glucat::Tuning_Slow_Fast_Size_Threshold = Tuning_Max_Threshold
```

Definition at line 43 of file tuning.h.

## 5.2.3.24 Tuning\_Slow\_Inv\_Fast\_Dim\_Threshold

```
const unsigned int glucat::Tuning_Slow_Inv_Fast_Dim_Threshold = Tuning_Max_Threshold
```

Definition at line 44 of file tuning.h.

## 5.2.3.25 Tuning\_Slow\_Mult\_Matrix\_Threshold

```
const unsigned int glucat::Tuning_Slow_Mult_Matrix_Threshold = Tuning_Max_Threshold
```

Definition at line 41 of file tuning.h.

## 5.2.3.26 Tuning\_Slow\_Products\_Size\_Threshold

```
const unsigned int glucat::Tuning_Slow_Products_Size_Threshold = Tuning_Max_Threshold
```

Definition at line 45 of file tuning.h.

## 5.3 glucat::gen Namespace Reference

## Classes

- class [generator\\_table](#)  
*Table of generators for specific signatures.*

## Typedefs

- using [signature\\_t](#) = std::pair< [index\\_t](#), [index\\_t](#) >  
*A signature is a pair of indices, p, q, with p == frame.max(), q == -frame.min()*

## Variables

- static const std::array< [index\\_t](#), 8 > [offset\\_to\\_super](#) = {0,-1, 0,-1,-2, 3, 2, 1}  
*Offsets between the current signature and that of the real superalgebra.*

## 5.3.1 Typedef Documentation

### 5.3.1.1 signature\_t

```
using glucat::gen::signature_t = typedef std::pair<index_t, index_t>
```

A signature is a pair of indices, p, q, with  $p == \text{frame.max}()$ ,  $q == -\text{frame.min}()$

Definition at line 48 of file generation.h.

## 5.3.2 Variable Documentation

### 5.3.2.1 offset\_to\_super

```
const std::array<index_t, 8> glucat::gen::offset_to_super = {0,-1, 0,-1,-2, 3, 2, 1} [static]
```

Offsets between the current signature and that of the real superalgebra.

Definition at line 86 of file generation.h.

Referenced by `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::fast_framed_multi()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fast_matrix_multi()`, and `glucat::gen::generator_table< Matrix_T >::operator()`.

## 5.4 glucat::matrix Namespace Reference

### Classes

- struct `eig_genus`

*Structure containing classification of eigenvalues.*

### Typedefs

- using `eig_case_t` = enum { safe\_eigs, neg\_real\_eigs, both\_eigs }

*Classification of eigenvalues of a matrix.*

## Functions

- template<typename LHS\_T , typename RHS\_T >  
auto [kron](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> const RHS\_T  
*Kronecker tensor product of matrices - as per Matlab kron.*
- template<typename LHS\_T , typename RHS\_T >  
auto [mono\\_kron](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> const RHS\_T  
*Sparse Kronecker tensor product of monomial matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [nork](#) (const LHS\_T &lhs, const RHS\_T &rhs, const bool mono=true) -> const RHS\_T  
*Left inverse of Kronecker product.*
- template<typename LHS\_T , typename RHS\_T >  
auto [signed\\_perm\\_nork](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> const RHS\_T  
*Left inverse of Kronecker product where lhs is a signed permutation matrix.*
- template<typename Matrix\_T >  
auto [nnz](#) (const Matrix\_T &m) -> typename Matrix\_T::size\_type  
*Number of non-zeros.*
- template<typename Matrix\_T >  
auto [isinf](#) (const Matrix\_T &m) -> bool  
*Infinite.*
- template<typename Matrix\_T >  
auto [isnan](#) (const Matrix\_T &m) -> bool  
*Not a Number.*
- template<typename Matrix\_T >  
auto [unit](#) (const typename Matrix\_T::size\_type n) -> const Matrix\_T  
*Unit matrix - as per Matlab eye.*
- template<typename LHS\_T , typename RHS\_T >  
auto [mono\\_prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of monomial matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [sparse\\_prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of sparse matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of matrices.*
- template<typename Scalar\_T , typename LHS\_T , typename RHS\_T >  
auto [inner](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> Scalar\_T  
*Inner product:  $\sum(x(i,j)*y(i,j))/x.nrows()$*
- template<typename Matrix\_T >  
auto [norm\\_frob2](#) (const Matrix\_T &val) -> typename Matrix\_T::value\_type  
*Square of Frobenius norm.*
- template<typename Matrix\_T >  
auto [trace](#) (const Matrix\_T &val) -> typename Matrix\_T::value\_type  
*Matrix trace.*
- template<typename Matrix\_T >  
auto [eigenvalues](#) (const Matrix\_T &val) -> std::vector< std::complex< double > >  
*Eigenvalues of a matrix.*
- template<typename Matrix\_T >  
auto [classify\\_eigenvalues](#) (const Matrix\_T &val) -> [eig\\_genus](#)< Matrix\_T >  
*Classify the eigenvalues of a matrix.*

- `template<typename LHS_T, typename RHS_T >`  
`void nork\_range (RHS_T &result, const typename LHS_T::const_iterator2 lhs_it2, const RHS_T &rhs, const`  
`typename RHS_T::size_type res_s1, const typename RHS_T::size_type res_s2)`  
*Utility routine for nork: calculate result for a range of indices.*
- `template<typename Matrix_T >`  
`static auto to\_lapack (const Matrix_T &val) -> ublas::matrix< double, ublas::column_major >  
Convert matrix to LAPACK format.`

## 5.4.1 Typedef Documentation

### 5.4.1.1 eig\_case\_t

```
using glucat::matrix::eig\_case\_t = typedef enum { safe_eigs, neg_real_eigs, both_eigs}
```

Classification of eigenvalues of a matrix.

Definition at line 136 of file `matrix.h`.

## 5.4.2 Function Documentation

### 5.4.2.1 classify\_eigenvalues()

```
template<typename Matrix_T >
auto glucat::matrix::classify\_eigenvalues (
    const Matrix_T & val ) -> eig\_genus<Matrix_T>
```

Classify the eigenvalues of a matrix.

Definition at line 548 of file `matrix_imp.h`.

References `eigenvalues()`, `epsilon`, `glucat::imag()`, `glucat::matrix::eig_genus< Matrix_T >::m_eig_case`, `glucat::matrix::eig_genus< Matrix_T >::m_is_singular`, `glucat::matrix::eig_genus< Matrix_T >::m_safe_arg`, `glucat::norm()`, `glucat::numeric_traits< Scalar_T >::pi()`, `PyClical::pi`, and `glucat::real()`.

Referenced by `glucat::matrix_log()`, and `glucat::matrix_sqrt()`.

### 5.4.2.2 eigenvalues()

```
template<typename Matrix_T >
auto glucat::matrix::eigenvalues (
    const Matrix_T & val ) -> std::vector< std::complex<double> >
```

Eigenvalues of a matrix.

Definition at line 500 of file `matrix_imp.h`.

References `to_lapack()`.

Referenced by `classify_eigenvalues()`.

## 5.4.2.3 inner()

```
template<typename Scalar_T , typename LHS_T , typename RHS_T >
auto glucat::matrix::inner (
    const LHS_T & lhs,
    const RHS_T & rhs ) -> Scalar_T
```

Inner product:  $\text{sum}(x(i,j)*y(i,j))/x.\text{nrows}()$

Inner product:  $\text{sum}(lhs(i,j)*rhs(i,j))/lhs.\text{nrows}()$

Definition at line 373 of file matrix\_imp.h.

References PyClical::lhs, and PyClical::rhs.

## 5.4.2.4 isinf()

```
template<typename Matrix_T >
auto glucat::matrix::isinf (
    const Matrix_T & m ) -> bool
```

Infinite.

Definition at line 275 of file matrix\_imp.h.

## 5.4.2.5 isnan()

```
template<typename Matrix_T >
auto glucat::matrix::isnan (
    const Matrix_T & m ) -> bool
```

Not a Number.

Definition at line 292 of file matrix\_imp.h.

Referenced by glucat::matrix\_log(), and glucat::operator/().

## 5.4.2.6 kron()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::kron (
    const LHS_T & lhs,
    const RHS_T & rhs ) -> const RHS_T
```

Kronecker tensor product of matrices - as per Matlab kron.

Definition at line 83 of file matrix\_imp.h.

References PyClical::lhs, and PyClical::rhs.

Referenced by glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::fast().

#### 5.4.2.7 mono\_kron()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::mono_kron (
    const LHS_T & lhs,
    const RHS_T & rhs ) -> const RHS_T
```

Sparse Kronecker tensor product of monomial matrices.

Definition at line 119 of file matrix\_imp.h.

References `PyClical::lhs`, and `PyClical::rhs`.

Referenced by `glucat::gen::generator_table< Matrix_T >::gen_from_pm1_qm1()`.

#### 5.4.2.8 mono\_prod()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::mono_prod (
    const ublas::matrix_expression< LHS_T > & lhs,
    const ublas::matrix_expression< RHS_T > & rhs ) -> const typename RHS_T::expression←
_type
```

Product of monomial matrices.

Definition at line 320 of file matrix\_imp.h.

References `PyClical::lhs`, and `PyClical::rhs`.

Referenced by `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::basis_element()`, `glucat::gen::generator_table< Matrix_T >::gen_from_pm4_qp4()`, `glucat::gen::generator_table< Matrix_T >::gen_from_pp4_qm4()`, and `glucat::gen::generator_table< Matrix_T >::gen_from_qp1_pm1()`.

#### 5.4.2.9 nnz()

```
template<typename Matrix_T >
auto glucat::matrix::nnz (
    const Matrix_T & m ) -> typename Matrix_T::size_type
```

Number of non-zeros.

Definition at line 258 of file matrix\_imp.h.

Referenced by `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi()`.



## 5.4.2.10 nork()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::nork (
    const LHS_T & lhs,
    const RHS_T & rhs,
    const bool mono = true ) -> const RHS_T
```

Left inverse of Kronecker product.

Definition at line 182 of file matrix\_imp.h.

References PyClical::lhs, norm\_frob2(), and PyClical::rhs.

## 5.4.2.11 nork\_range()

```
template<typename LHS_T , typename RHS_T >
void glucat::matrix::nork_range (
    RHS_T & result,
    const typename LHS_T::const_iterator2 lhs_it2,
    const RHS_T & rhs,
    const typename RHS_T::size_type res_s1,
    const typename RHS_T::size_type res_s2 )
```

Utility routine for nork: calculate result for a range of indices.

Definition at line 152 of file matrix\_imp.h.

References PyClical::rhs, and glucat::numeric\_traits< Scalar\_T >::to\_scalar\_t().

## 5.4.2.12 norm\_frob2()

```
template<typename Matrix_T >
auto glucat::matrix::norm_frob2 (
    const Matrix_T & val ) -> typename Matrix_T::value_type
```

Square of Frobenius norm.

Definition at line 395 of file matrix\_imp.h.

Referenced by nork().

#### 5.4.2.13 prod()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::prod (
    const ublas::matrix_expression< LHS_T > & lhs,
    const ublas::matrix_expression< RHS_T > & rhs ) -> const typename RHS_T::expression←
_type [inline]
```

Product of matrices.

Definition at line 361 of file matrix\_imp.h.

References PyClical::lhs, and PyClical::rhs.

#### 5.4.2.14 signed\_perm\_nork()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::signed_perm_nork (
    const LHS_T & lhs,
    const RHS_T & rhs ) -> const RHS_T
```

Left inverse of Kronecker product where lhs is a signed permutation matrix.

Definition at line 228 of file matrix\_imp.h.

References PyClical::lhs, and PyClical::rhs.

Referenced by glucat::fast().

#### 5.4.2.15 sparse\_prod()

```
template<typename LHS_T , typename RHS_T >
auto glucat::matrix::sparse_prod (
    const ublas::matrix_expression< LHS_T > & lhs,
    const ublas::matrix_expression< RHS_T > & rhs ) -> const typename RHS_T::expression←
_type [inline]
```

Product of sparse matrices.

Definition at line 350 of file matrix\_imp.h.

References PyClical::lhs, and PyClical::rhs.

## 5.4.2.16 to\_lapack()

```
template<typename Matrix_T >
static auto glucat::matrix::to_lapack (
    const Matrix_T & val ) -> ublas::matrix<double, ublas::column_major> [static]
```

Convert matrix to LAPACK format.

Definition at line 440 of file matrix\_imp.h.

Referenced by eigenvalues().

## 5.4.2.17 trace()

```
template<typename Matrix_T >
auto glucat::matrix::trace (
    const Matrix_T & val ) -> typename Matrix_T::value_type
```

Matrix trace.

Definition at line 416 of file matrix\_imp.h.

References glucat::numeric\_traits< Scalar\_T >::NaN().

## 5.4.2.18 unit()

```
template<typename Matrix_T >
auto glucat::matrix::unit (
    const typename Matrix_T::size_type n ) -> const Matrix_T [inline]
```

Unit matrix - as per Matlab eye.

Definition at line 310 of file matrix\_imp.h.

## 5.5 glucat::timing Namespace Reference

## Functions

- static double [elapsed](#) (clock\_t cpu\_time)  
*Elapsed time in milliseconds.*

## Variables

- const double [MS\\_PER\\_SEC](#) = 1000.0  
*Timing constant: milliseconds per second.*
- const double [MS\\_PER\\_CLOCK](#) = [MS\\_PER\\_SEC](#) / double(CLOCKS\_PER\_SEC)  
*Timing constant: milliseconds per clock.*
- const int [EXTRA\\_TRIALS](#) = 2  
*Timing constant: trial expansion factor.*

## 5.5.1 Function Documentation

### 5.5.1.1 elapsed()

```
static double glucat::timing::elapsed (  
    clock_t cpu_time ) [inline], [static]
```

Elapsed time in milliseconds.

Definition at line 51 of file timing.h.

References `MS_PER_CLOCK`.

## 5.5.2 Variable Documentation

### 5.5.2.1 EXTRA\_TRIALS

```
const int glucat::timing::EXTRA_TRIALS = 2
```

Timing constant: trial expansion factor.

Definition at line 45 of file timing.h.

### 5.5.2.2 MS\_PER\_CLOCK

```
const double glucat::timing::MS_PER_CLOCK = MS\_PER\_SEC / double(CLOCKS_PER_SEC)
```

Timing constant: milliseconds per clock.

Definition at line 42 of file timing.h.

Referenced by `elapsed()`.

### 5.5.2.3 MS\_PER\_SEC

```
const double glucat::timing::MS_PER_SEC = 1000.0
```

Timing constant: milliseconds per second.

Definition at line 39 of file timing.h.

## 5.6 pade Namespace Reference

### Classes

- struct [pade\\_log\\_denom](#)

*Coefficients of denominator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)*

- struct [pade\\_log\\_denom](#)< dd\_real >
- struct [pade\\_log\\_denom](#)< float >
- struct [pade\\_log\\_denom](#)< long double >
- struct [pade\\_log\\_denom](#)< qd\_real >
- struct [pade\\_log\\_numer](#)

*Coefficients of numerator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)*

- struct [pade\\_log\\_numer](#)< dd\_real >
- struct [pade\\_log\\_numer](#)< float >
- struct [pade\\_log\\_numer](#)< long double >
- struct [pade\\_log\\_numer](#)< qd\_real >
- struct [pade\\_sqrt\\_denom](#)

*Coefficients of denominator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)*

- struct [pade\\_sqrt\\_denom](#)< dd\_real >
- struct [pade\\_sqrt\\_denom](#)< float >
- struct [pade\\_sqrt\\_denom](#)< long double >
- struct [pade\\_sqrt\\_denom](#)< qd\_real >
- struct [pade\\_sqrt\\_numer](#)

*Coefficients of numerator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)*

- struct [pade\\_sqrt\\_numer](#)< dd\_real >
- struct [pade\\_sqrt\\_numer](#)< float >
- struct [pade\\_sqrt\\_numer](#)< long double >
- struct [pade\\_sqrt\\_numer](#)< qd\_real >

## 5.7 PyClical Namespace Reference

### Classes

- class [clifford](#)
- class [index\\_set](#)

### Functions

- def [index\\_set\\_hidden\\_doctests](#) ()
- def [clifford\\_hidden\\_doctests](#) ()
- def [e](#) (obj)
- def [istpq](#) (p, q)
- def [\\_test](#) ()

## Variables

- `__version__` = str(`glucat_package_version`, 'utf-8')
- `lhs`
- `rhs`
- `threshold` = error\_squared\_tol(`rhs`) if threshold is `None` else threshold
- `None`
- `tol` = error\_squared\_tol(`rhs`) if tol is `None` else tol
- `obj`
- `i`
- `ixt`
- `fill`
- `scalar_epsilon` = `epsilon`
- float `pi` = atan(`clifford`(1.0)) \* 4.0
- float `tau` = atan(`clifford`(1.0)) \* 8.0
- `cl` = `clifford`
- `ist` = `index_set`
- def `ninf3` = `e`(4) + `e`(-1)
- def `nbar3` = `e`(4) - `e`(-1)

## 5.7.1 Function Documentation

### 5.7.1.1 `_test()`

```
def PyClical._test ( ) [private]
```

Definition at line 1962 of file PyClical.pyx.

### 5.7.1.2 `clifford_hidden_doctests()`

```
def PyClical.clifford_hidden_doctests ( )
```

Tests for functions that Doctest cannot see.

For `clifford.__cinit__`: Construct an object of type `clifford`.

```
>>> print(clifford(2))
2
>>> print(clifford(2.0))
2
>>> print(clifford(1.0e-1))
0.1
>>> print(clifford("2"))
2
>>> print(clifford("2{1,2,3}"))
2{1,2,3}
>>> print(clifford(clifford("2{1,2,3}")))
2{1,2,3}
>>> print(clifford("-{1}"))
-{1}
>>> print(clifford(2, index_set({1,2})))
```

```

2{1,2}
>>> print(clifford([2,3],index_set({1,2})))
2{1}+3{2}
>>> print(clifford([1,2]))
Traceback (most recent call last):
...
TypeError: Cannot initialize clifford object from <class 'list'>.
>>> print(clifford(None))
Traceback (most recent call last):
...
TypeError: Cannot initialize clifford object from <class 'NoneType'>.
>>> print(clifford(None,[1,2]))
Traceback (most recent call last):
...
TypeError: Cannot initialize clifford object from (<class 'NoneType'>, <class 'list'>).
>>> print(clifford([1,2],[1,2]))
Traceback (most recent call last):
...
TypeError: Cannot initialize clifford object from (<class 'list'>, <class 'list'>).
>>> print(clifford(""))
Traceback (most recent call last):
...
ValueError: Cannot initialize clifford object from invalid string ''.
>>> print(clifford("{")
Traceback (most recent call last):
...
ValueError: Cannot initialize clifford object from invalid string '{'.
>>> print(clifford("{1")
Traceback (most recent call last):
...
ValueError: Cannot initialize clifford object from invalid string '{1'.
>>> print(clifford("{+")
Traceback (most recent call last):
...
ValueError: Cannot initialize clifford object from invalid string '+'.
>>> print(clifford("-")
Traceback (most recent call last):
...
ValueError: Cannot initialize clifford object from invalid string '-'.
>>> print(clifford("{1}+")
Traceback (most recent call last):
...
ValueError: Cannot initialize clifford object from invalid string '{1}+'.

For clifford.__richcmp__: Compare objects of type clifford.

>>> clifford("{1}") == clifford("1{1}")
True
>>> clifford("{1}") != clifford("1.0{1}")
False
>>> clifford("{1}") != clifford("1.0")
True
>>> clifford("{1,2}") == None
False
>>> clifford("{1,2}") != None
True
>>> None == clifford("{1,2}")
False
>>> None != clifford("{1,2}")
True

```

Definition at line 1253 of file PyClical.pyx.

### 5.7.1.3 e()

```

def PyClical.e (
    obj )

```

Abbreviation for `clifford(index_set(obj))`.

```
>>> print(e(1))
{1}
>>> print(e(-1))
{-1}
>>> print(e(0))
1
```

Definition at line 1936 of file `PyClical.pyx`.

Referenced by `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >.basis_element()`, `clifford_to_str()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.framed_multi()`, `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >.matrix_multi()`, and `glucat.try_catch()`.

#### 5.7.1.4 index\_set\_hidden\_doctests()

```
def PyClical.index_set_hidden_doctests ( )
```

Tests for functions that Doctest cannot see.

For `index_set.__cinit__`: Construct `index_set`.

```
>>> print(index_set(1))
{1}
>>> print(index_set({1,2}))
{1,2}
>>> print(index_set(index_set({1,2})))
{1,2}
>>> print(index_set({1,2}))
{1,2}
>>> print(index_set({1,2,1}))
{1,2}
>>> print(index_set({1,2,1}))
{1,2}
>>> print(index_set(""))
{}
>>> print(index_set("{}"))
Traceback (most recent call last):
...
ValueError: Cannot initialize index_set object from invalid string '{}'.
>>> print(index_set("{1}"))
Traceback (most recent call last):
...
ValueError: Cannot initialize index_set object from invalid string '{1}'.
>>> print(index_set("{1,2,100}"))
Traceback (most recent call last):
...
ValueError: Cannot initialize index_set object from invalid string '{1,2,100}'.
>>> print(index_set({1,2,100}))
Traceback (most recent call last):
...
IndexError: Cannot initialize index_set object from invalid {1, 2, 100}.
>>> print(index_set([1,2]))
Traceback (most recent call last):
...
TypeError: Cannot initialize index_set object from <class 'list'>.
```

For `index_set.__richcmp__`: Compare two objects of class `index_set`.

```
>>> index_set(1) == index_set({1})
True
>>> index_set({1}) != index_set({1})
False
```



```

>>> index_set({1}) != index_set({2})
True
>>> index_set({1}) == index_set({2})
False
>>> index_set({1}) < index_set({2})
True
>>> index_set({1}) <= index_set({2})
True
>>> index_set({1}) > index_set({2})
False
>>> index_set({1}) >= index_set({2})
False
>>> None == index_set({1,2})
False
>>> None != index_set({1,2})
True
>>> None < index_set({1,2})
False
>>> None <= index_set({1,2})
False
>>> None > index_set({1,2})
False
>>> None >= index_set({1,2})
False
>>> index_set({1,2}) == None
False
>>> index_set({1,2}) != None
True
>>> index_set({1,2}) < None
False
>>> index_set({1,2}) <= None
False
>>> index_set({1,2}) > None
False
>>> index_set({1,2}) >= None
False

```

Definition at line 406 of file PyClical.pyx.

#### 5.7.1.5 istpq()

```

def PyClical.istpq (
    p,
    q )

```

Abbreviation for `index_set({-q,...p})`.

```

>>> print(istpq(2,3))
{-3,-2,-1,1,2}

```

Definition at line 1949 of file PyClical.pyx.

## 5.7.2 Variable Documentation

### 5.7.2.1 `__version__`

```
PyClical.__version__ = str(glucat_package_version, 'utf-8') [private]
```

Definition at line 35 of file PyClical.pyx.

### 5.7.2.2 `cl`

```
PyClical.cl = clifford
```

Definition at line 1910 of file PyClical.pyx.

Referenced by `cga3.agc3()`, `cga3.cga3()`, and `cga3.cga3std()`.

### 5.7.2.3 `fill`

```
PyClical.fill
```

Definition at line 1864 of file PyClical.pyx.

Referenced by `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.random()`, and `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >.random()`.

### 5.7.2.4 `i`

```
PyClical.i
```

Definition at line 1591 of file PyClical.pyx.

Referenced by `glucat.acos()`, `glucat.acosh()`, `glucat.asin()`, `glucat.asinh()`, `glucat.atan()`, `glucat.atanh()`, `glucat.check_complex()`, `glucat.cos()`, `glucat.log()`, `glucat.matrix_log()`, `glucat.matrix_sqrt()`, `glucat.operator<<()`, `glucat.operator>>()`, `glucat.sin()`, `glucat.sqrt()`, and `glucat.tan()`.

### 5.7.2.5 `ist`

```
PyClical.ist = index_set
```

Definition at line 1928 of file PyClical.pyx.

Referenced by `cga3.agc3()`, `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >.basis_element()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.centre_pm4_qp4()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.centre_pp4_qm4()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.centre_qp1_pm1()`, `cga3.cga3()`, `cga3.cga3std()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.divide()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >.framed_multi()`, `index_set_to_repr()`, `index_set_to_str()`, `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >.matrix_multi()`, `glucat.max_pos()`, `glucat.min_neg()`, `glucat.operator<<()`, and `glucat.operator>>()`.

### 5.7.2.6 ixt

`PyClical.ixt`

Definition at line 1864 of file `PyClical.pyx`.

### 5.7.2.7 lhs

`PyClical.lhs`

Definition at line 1359 of file `PyClical.pyx`.

Referenced by `glucat.approx_equal()`, `glucat.crd_of_mult()`, `glucat.error_squared()`, `glucat::numeric_traits<Scalar_T >.fmod()`, `glucat::matrix.inner()`, `glucat::matrix.kron()`, `glucat::matrix.mono_kron()`, `glucat::matrix.↔mono_prod()`, `glucat::matrix.nork()`, `glucat.operator &()`, `glucat.operator!=()`, `glucat.operator%()`, `glucat.operator*()`, `glucat.operator+()`, `glucat.operator-()`, `glucat.operator/()`, `glucat.operator^()`, `glucat.operator|()`, `glucat.outer_↔pow()`, `glucat.pos_mod()`, `glucat.pow()`, `glucat::matrix.prod()`, `glucat.reframe()`, `glucat::matrix.signed_perm_nork()`, `glucat::matrix.sparse_prod()`, and `glucat.star()`.

### 5.7.2.8 nbar3

```
def PyClical.nbar3 = e(4) - e(-1)
```

Definition at line 1959 of file `PyClical.pyx`.

### 5.7.2.9 ninf3

```
def PyClical.ninf3 = e(4) + e(-1)
```

Definition at line 1958 of file `PyClical.pyx`.

Referenced by `cga3.cga3()`, and `cga3.cga3std()`.

### 5.7.2.10 None

`PyClical.None`

Definition at line 1359 of file `PyClical.pyx`.

### 5.7.2.11 obj

PyClical.obj

Definition at line 1591 of file PyClical.pyx.

### 5.7.2.12 pi

```
float PyClical.pi = atan(clifford(1.0)) * 4.0
```

Definition at line 1907 of file PyClical.pyx.

Referenced by `glucat::matrix.classify_eigenvalues()`, `glucat.cos()`, `glucat.log()`, `glucat.matrix_log()`, and `glucat.sin()`.

### 5.7.2.13 rhs

PyClical.rhs

Definition at line 1359 of file PyClical.pyx.

Referenced by `glucat.approx_equal()`, `glucat.crd_of_mult()`, `glucat.error_squared()`, `glucat::numeric_traits< Scalar_T >.fmod()`, `glucat::matrix.inner()`, `glucat::matrix.kron()`, `glucat::index_set< LO, HI >.lex_less_than()`, `glucat::matrix.mono_kron()`, `glucat::matrix.mono_prod()`, `glucat::matrix.nork()`, `glucat::matrix.nork_range()`, `glucat.operator &()`, `glucat::index_set< LO, HI >.operator!=()`, `glucat.operator!=()`, `glucat.operator%()`, `glucat.operator*()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term.operator*=()`, `glucat.operator+()`, `glucat.operator-()`, `glucat.operator/()`, `glucat::index_set< LO, HI >.operator<()`, `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >.operator=()`, `glucat::index_set< LO, HI >.operator==()`, `glucat.operator^()`, `glucat::index_set< LO, HI >.operator^=()`, `glucat.operator|()`, `glucat::index_set< LO, HI >.operator|=()`, `glucat.outer_pow()`, `glucat.pos_mod()`, `glucat.pow()`, `glucat::matrix.prod()`, `glucat.reframe()`, `glucat::index_set< LO, HI >.sign_of_mult()`, `glucat::matrix.signed_perm_nork()`, `glucat::matrix.sparse_prod()`, and `glucat.star()`.

### 5.7.2.14 scalar\_epsilon

```
PyClical.scalar_epsilon = epsilon
```

Definition at line 1905 of file PyClical.pyx.

### 5.7.2.15 tau

```
float PyClical.tau = atan(clifford(1.0)) * 8.0
```

Definition at line 1908 of file PyClical.pyx.

#### 5.7.2.16 threshold

```
PyClical.threshold = error_squared_tol(rhs) if threshold is None else threshold
```

Definition at line 1359 of file PyClical.pyx.

Referenced by `glucat.approx_equal()`, and `glucat.error_squared()`.

#### 5.7.2.17 tol

```
PyClical.tol = error_squared_tol(rhs) if tol is None else tol
```

Definition at line 1359 of file PyClical.pyx.

Referenced by `glucat.operator<<()`.

## 5.8 std Namespace Reference

### Classes

- struct `numeric_limits< glucat::framed_multi< Scalar_T, LO, HI, Tune_P > >`  
*Numeric limits for framed\_multi inherit limits for the corresponding scalar type.*
- struct `numeric_limits< glucat::matrix_multi< Scalar_T, LO, HI, Tune_P > >`  
*Numeric limits for matrix\_multi inherit limits for the corresponding scalar type.*



## Chapter 6

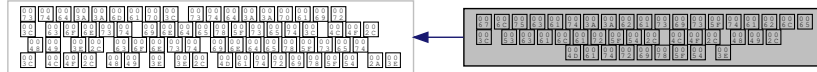
# Class Documentation

### 6.1 `glucat::basis_table< Scalar_T, LO, HI, Matrix_T >` Class Template Reference

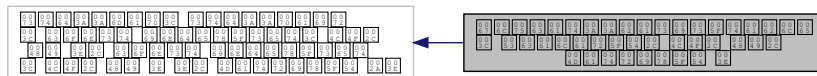
Table of basis elements used as a cache by `basis_element()`

```
#include <matrix_multi_imp.h>
```

Inheritance diagram for `glucat::basis_table< Scalar_T, LO, HI, Matrix_T >`:



Collaboration diagram for `glucat::basis_table< Scalar_T, LO, HI, Matrix_T >`:



#### Public Member Functions

- `basis_table` (const `basis_table` &)=delete
- auto `operator=` (const `basis_table` &) -> `basis_table` &=delete

#### Static Public Member Functions

- static auto `basis` () -> `basis_table` &  
*Single instance of basis table.*

## Private Member Functions

- [basis\\_table\(\)](#)=default
- [~basis\\_table\(\)](#)=default

## Friends

- class [friend\\_for\\_private\\_destructor](#)

### 6.1.1 Detailed Description

```
template<typename Scalar_T, const index_t LO, const index_t HI, typename Matrix_T>
class glucat::basis_table< Scalar_T, LO, HI, Matrix_T >
```

Table of basis elements used as a cache by [basis\\_element\(\)](#)

Definition at line 1162 of file [matrix\\_multi\\_imp.h](#).

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 [basis\\_table\(\)](#) [1/2]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Matrix_T >
glucat::basis_table< Scalar_T, LO, HI, Matrix_T >::basis_table ( ) [private], [default]
```

#### 6.1.2.2 [~basis\\_table\(\)](#)

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Matrix_T >
glucat::basis_table< Scalar_T, LO, HI, Matrix_T >::~~basis_table ( ) [private], [default]
```

#### 6.1.2.3 [basis\\_table\(\)](#) [2/2]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Matrix_T >
glucat::basis_table< Scalar_T, LO, HI, Matrix_T >::basis_table (
    const basis\_table< Scalar_T, LO, HI, Matrix_T > & ) [delete]
```

### 6.1.3 Member Function Documentation



### 6.1.3.1 basis()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Matrix_T >
static auto glucat::basis_table< Scalar_T, LO, HI, Matrix_T >::basis ( ) -> basis_table&
[inline], [static]
```

Single instance of basis table.

Definition at line 1168 of file matrix\_multi\_imp.h.

### 6.1.3.2 operator=()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Matrix_T >
auto glucat::basis_table< Scalar_T, LO, HI, Matrix_T >::operator= (
    const basis_table< Scalar_T, LO, HI, Matrix_T > & ) -> basis_table &=delete
[delete]
```

## 6.1.4 Friends And Related Function Documentation

### 6.1.4.1 friend\_for\_private\_destructor

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Matrix_T >
friend class friend_for_private_destructor [friend]
```

Friend declaration to avoid compiler warning: "... only defines a private destructor and has no friends" Ref: Carlos O'Ryan, ACE <http://doc.ece.uci.edu>

Definition at line 1173 of file matrix\_multi\_imp.h.

The documentation for this class was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)

## 6.2 glucat::bool\_to\_type< truth\_value > Class Template Reference

Bool to type.

```
#include <global.h>
```

### Private Types

- enum { [value](#) = truth\_value }

### 6.2.1 Detailed Description

```
template<bool truth_value>
class glucat::bool_to_type< truth_value >
```

Bool to type.

Definition at line 69 of file global.h.

### 6.2.2 Member Enumeration Documentation

#### 6.2.2.1 anonymous enum

```
template<bool truth_value>
anonymous enum [private]
```

Enumerator

value	
-------	--

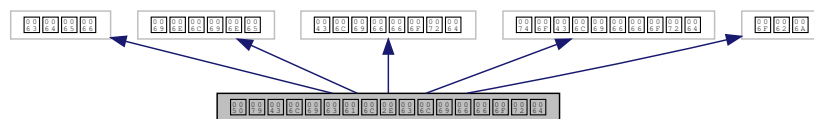
Definition at line 72 of file global.h.

The documentation for this class was generated from the following file:

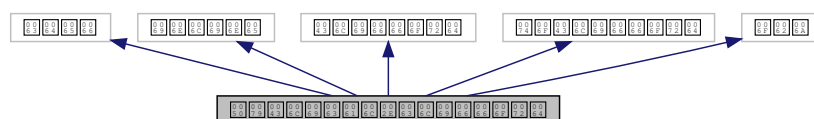
- glucat/[global.h](#)

## 6.3 PyClical.clifford Class Reference

Inheritance diagram for PyClical.clifford:



Collaboration diagram for PyClical.clifford:



## Public Member Functions

- `def __cinit__ (self, other=0, ixt=None)`
- `def __dealloc__ (self)`
- `def __contains__ (self, x)`
- `def __iter__ (self)`
- `def reframe (self, ixt)`
- `def __richcmp__ (lhs, rhs, int, op)`
- `def __getitem__ (self, ixt)`
- `def __neg__ (self)`
- `def __pos__ (self)`
- `def __add__ (lhs, rhs)`
- `def __iadd__ (self, rhs)`
- `def __sub__ (lhs, rhs)`
- `def __isub__ (self, rhs)`
- `def __mul__ (lhs, rhs)`
- `def __imul__ (self, rhs)`
- `def __mod__ (lhs, rhs)`
- `def __imod__ (self, rhs)`
- `def __and__ (lhs, rhs)`
- `def __iand__ (self, rhs)`
- `def __xor__ (lhs, rhs)`
- `def __ixor__ (self, rhs)`
- `def __truediv__ (lhs, rhs)`
- `def __idiv__ (self, rhs)`
- `def inv (self)`
- `def __or__ (lhs, rhs)`
- `def __ior__ (self, rhs)`
- `def __pow__ (self, m, dummy)`
- `def pow (self, m)`
- `def outer_pow (self, m)`
- `def __call__ (self, grade)`
- `def scalar (self)`
- `def pure (self)`
- `def even (self)`
- `def odd (self)`
- `def vector_part (self, frm=None)`
- `def involute (self)`
- `def reverse (self)`
- `def conj (self)`
- `def quad (self)`
- `def norm (self)`
- `def abs (self)`
- `def max_abs (self)`
- `def truncated (self, limit)`
- `def isinf (self)`
- `def isnan (self)`
- `def frame (self)`
- `def __repr__ (self)`
- `def __str__ (self)`

## Public Attributes

- `instance`

### 6.3.1 Detailed Description

Python class `clifford` wraps C++ class `Clifford`.

Definition at line 534 of file `PyClical.pyx`.

### 6.3.2 Member Function Documentation

#### 6.3.2.1 `__add__()`

```
def PyClical.clifford.__add__ (
    lhs,
    rhs )
```

Geometric sum.

```
>>> print(clifford(1) + clifford("{2}"))
1+{2}
>>> print(clifford("{1}") + clifford("{2}"))
{1}+{2}
```

Definition at line 740 of file `PyClical.pyx`.

#### 6.3.2.2 `__and__()`

```
def PyClical.clifford.__and__ (
    lhs,
    rhs )
```

Inner product.

```
>>> print(clifford("{1}") & clifford("{2}"))
0
>>> print(clifford(2) & clifford("{2}"))
0
>>> print(clifford("{1}") & clifford("{1}"))
1
>>> print(clifford("{1}") & clifford("{1,2}"))
{2}
```

Definition at line 836 of file `PyClical.pyx`.

6.3.2.3 `__call__()`

```
def PyClical.clifford.__call__ (
    self,
    grade )
```

Pure grade-vector part.

```
>>> print(clifford("{1}") (1))
{1}
>>> print(clifford("{1}") (0))
0
>>> print(clifford("1+{1}+{1,2}") (0))
1
>>> print(clifford("1+{1}+{1,2}") (1))
{1}
>>> print(clifford("1+{1}+{1,2}") (2))
{1,2}
>>> print(clifford("1+{1}+{1,2}") (3))
0
```

Definition at line 1020 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

6.3.2.4 `__cinit__()`

```
def PyClical.clifford.__cinit__ (
    self,
    other = 0,
    ixt = None )
```

Construct an object of type clifford.

```
>>> print(clifford(2))
2
>>> print(clifford(2.0))
2
>>> print(clifford(1.0e-1))
0.1
>>> print(clifford("2"))
2
>>> print(clifford("2{1,2,3}"))
2{1,2,3}
>>> print(clifford(clifford("2{1,2,3}")))
2{1,2,3}
>>> print(clifford("-{1}"))
-{1}
>>> print(clifford(2, index_set ({1,2})))
2{1,2}
>>> print(clifford([2,3], index_set ({1,2})))
2{1}+3{2}
```

Definition at line 565 of file PyClical.pyx.

**6.3.2.5 \_\_contains\_\_()**

```
def PyClical.clifford.__contains__ (
    self,
    x )
```

Not applicable.

```
>>> x=clifford(index_set({-3,4,7})); -3 in x
Traceback (most recent call last):
...
TypeError: Not applicable.
```

Definition at line 627 of file PyClical.pyx.

**6.3.2.6 \_\_dealloc\_\_()**

```
def PyClical.clifford.__dealloc__ (
    self )
```

Clean up by deallocating the instance of C++ class Clifford.

Definition at line 621 of file PyClical.pyx.

References PyClical.index\_set.instance, and PyClical.clifford.instance.

**6.3.2.7 \_\_getitem\_\_()**

```
def PyClical.clifford.__getitem__ (
    self,
    ixt )
```

Subscripting: map from index set to scalar coordinate.

```
>>> clifford("{1}")[index_set(1)]
1.0
>>> clifford("{1}")[index_set({1})]
1.0
>>> clifford("{1}")[index_set({1,2})]
0.0
>>> clifford("2{1,2}")[index_set({1,2})]
2.0
```

Definition at line 707 of file PyClical.pyx.

References PyClical.index\_set.instance, and PyClical.clifford.instance.

**6.3.2.8 \_\_iadd\_\_()**

```
def PyClical.clifford.__iadd__ (
    self,
    rhs )
```

Geometric sum.

```
>>> x = clifford(1); x += clifford("{2}"); print(x)
1+{2}
```

Definition at line 751 of file PyClical.pyx.

**6.3.2.9 \_\_iand\_\_()**

```
def PyClical.clifford.__iand__ (
    self,
    rhs )
```

Inner product.

```
>>> x = clifford("{1}"); x &= clifford("{2}"); print(x)
0
>>> x = clifford(2); x &= clifford("{2}"); print(x)
0
>>> x = clifford("{1}"); x &= clifford("{1}"); print(x)
1
>>> x = clifford("{1}"); x &= clifford("{1,2}"); print(x)
{2}
```

Definition at line 851 of file PyClical.pyx.

**6.3.2.10 \_\_idiv\_\_()**

```
def PyClical.clifford.__idiv__ (
    self,
    rhs )
```

Geometric quotient.

```
>>> x = clifford("{1}"); x /= clifford("{2}"); print(x)
{1,2}
>>> x = clifford(2); x /= clifford("{2}"); print(x)
2{2}
>>> x = clifford("{1}"); x /= clifford("{1}"); print(x)
1
>>> x = clifford("{1}"); x /= clifford("{1,2}"); print(x)
-{2}
```

Definition at line 911 of file PyClical.pyx.

**6.3.2.11 `__imod__()`**

```
def PyClical.clifford.__imod__ (
    self,
    rhs )
```

Contraction.

```
>>> x = clifford("{1}"); x %= clifford("{2}"); print(x)
0
>>> x = clifford(2); x %= clifford("{2}"); print(x)
2{2}
>>> x = clifford("{1}"); x %= clifford("{1}"); print(x)
1
>>> x = clifford("{1}"); x %= clifford("{1,2}"); print(x)
{2}
```

Definition at line 821 of file PyClical.pyx.

**6.3.2.12 `__imul__()`**

```
def PyClical.clifford.__imul__ (
    self,
    rhs )
```

Geometric product.

```
>>> x = clifford(2); x *= clifford("{2}"); print(x)
2{2}
>>> x = clifford("{1}"); x *= clifford("{2}"); print(x)
{1,2}
>>> x = clifford("{1}"); x *= clifford("{1,2}"); print(x)
{2}
```

Definition at line 793 of file PyClical.pyx.

**6.3.2.13 `__ior__()`**

```
def PyClical.clifford.__ior__ (
    self,
    rhs )
```

Transform left hand side, using right hand side as a transformation.

```
>>> x=clifford("{1,2}") * pi/2; y=clifford("{1}"); y|=x; print(y)
-{1}
>>> x=clifford("{1,2}") * pi/2; y=clifford("{1}"); y|=exp(x); print(y)
-{1}
```

Definition at line 950 of file PyClical.pyx.



**6.3.2.14** `__isub__()`

```
def PyClical.clifford.__isub__ (
    self,
    rhs )
```

Geometric difference.

```
>>> x = clifford(1); x -= clifford("{2}"); print(x)
1-{2}
```

Definition at line 771 of file PyClical.pyx.

**6.3.2.15** `__iter__()`

```
def PyClical.clifford.__iter__ (
    self )
```

Not applicable.

```
>>> for a in clifford(index_set({-3,4,7})):print(a, end=", ")
Traceback (most recent call last):
...
TypeError: Not applicable.
```

Definition at line 638 of file PyClical.pyx.

**6.3.2.16** `__ixor__()`

```
def PyClical.clifford.__ixor__ (
    self,
    rhs )
```

Outer product.

```
>>> x = clifford("{1}"); x ^= clifford("{2}"); print(x)
{1,2}
>>> x = clifford(2); x ^= clifford("{2}"); print(x)
2{2}
>>> x = clifford("{1}"); x ^= clifford("{1}"); print(x)
0
>>> x = clifford("{1}"); x ^= clifford("{1,2}"); print(x)
0
```

Definition at line 881 of file PyClical.pyx.

### 6.3.2.17 `__mod__()`

```
def PyClical.clifford.__mod__ (
    lhs,
    rhs )
```

Contraction.

```
>>> print(clifford("{1}") % clifford("{2}"))
0
>>> print(clifford(2) % clifford("{2}"))
2{2}
>>> print(clifford("{1}") % clifford("{1}"))
1
>>> print(clifford("{1}") % clifford("{1,2}"))
{2}
```

Definition at line 806 of file PyClical.pyx.

### 6.3.2.18 `__mul__()`

```
def PyClical.clifford.__mul__ (
    lhs,
    rhs )
```

Geometric product.

```
>>> print(clifford("{1}") * clifford("{2}"))
{1,2}
>>> print(clifford(2) * clifford("{2}"))
2{2}
>>> print(clifford("{1}") * clifford("{1,2}"))
{2}
```

Definition at line 780 of file PyClical.pyx.

### 6.3.2.19 `__neg__()`

```
def PyClical.clifford.__neg__ (
    self )
```

Unary `-`.

```
>>> print(-clifford("{1}"))
-{1}
```

Definition at line 722 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.20 `__or__()`**

```
def PyClical.clifford.__or__ (
    lhs,
    rhs )
```

Transform left hand side, using right hand side as a transformation.

```
>>> x=clifford("{1,2}") * pi/2; y=clifford("{1}"); print(y|x)
-{1}
>>> x=clifford("{1,2}") * pi/2; y=clifford("{1}"); print(y|exp(x))
-{1}
```

Definition at line 939 of file PyClical.pyx.

**6.3.2.21 `__pos__()`**

```
def PyClical.clifford.__pos__ (
    self )
```

Unary +.

```
>>> print(+clifford("{1}"))
{1}
```

Definition at line 731 of file PyClical.pyx.

**6.3.2.22 `__pow__()`**

```
def PyClical.clifford.__pow__ (
    self,
    m,
    dummy )
```

Power: self to the m.

```
>>> x=clifford("{1}"); print(x ** 2)
1
>>> x=clifford("2"); print(x ** 2)
4
>>> x=clifford("2+{1}"); print(x ** 0)
1
>>> x=clifford("2+{1}"); print(x ** 1)
2+{1}
>>> x=clifford("2+{1}"); print(x ** 2)
5+4{1}
>>> i=clifford("{1,2}"); print(exp(pi/2) * (i ** i))
1
```

Definition at line 961 of file PyClical.pyx.

References `PyClical.clifford.pow()`.

### 6.3.2.23 `__repr__()`

```
def PyClical.clifford.__repr__ (
    self )
```

The "official" string representation of self.

```
>>> clifford("1+3{-1}+2{1,2}+4{-2,7}").__repr__()
'clifford("1+3{-1}+2{1,2}+4{-2,7}")'
```

Definition at line 1235 of file PyClical.pyx.

References `clifford_to_repr()`.

### 6.3.2.24 `__richcmp__()`

```
def PyClical.clifford.__richcmp__ (
    lhs,
    rhs,
    int,
    op )
```

Compare objects of type clifford.

```
>>> clifford("{1}") == clifford("1{1}")
True
>>> clifford("{1}") != clifford("1.0{1}")
False
>>> clifford("{1}") != clifford("1.0")
True
>>> clifford("{1,2}") == None
False
>>> clifford("{1,2}") != None
True
>>> None == clifford("{1,2}")
False
>>> None != clifford("{1,2}")
True
```

Definition at line 672 of file PyClical.pyx.

### 6.3.2.25 `__str__()`

```
def PyClical.clifford.__str__ (
    self )
```

The "informal" string representation of self.

```
>>> clifford("1+3{-1}+2{1,2}+4{-2,7}").__str__()
'1+3{-1}+2{1,2}+4{-2,7}'
```

Definition at line 1244 of file PyClical.pyx.

References `clifford_to_str()`.

**6.3.2.26** `__sub__()`

```
def PyClical.clifford.__sub__ (
    lhs,
    rhs )
```

Geometric difference.

```
>>> print(clifford(1) - clifford("{2}"))
1-{2}
>>> print(clifford("{1}") - clifford("{2}"))
{1}-{2}
```

Definition at line 760 of file PyClical.pyx.

**6.3.2.27** `__truediv__()`

```
def PyClical.clifford.__truediv__ (
    lhs,
    rhs )
```

Geometric quotient.

```
>>> print(clifford("{1}") / clifford("{2}"))
{1,2}
>>> print(clifford(2) / clifford("{2}"))
2{2}
>>> print(clifford("{1}") / clifford("{1}"))
1
>>> print(clifford("{1}") / clifford("{1,2}"))
-{2}
```

Definition at line 896 of file PyClical.pyx.

**6.3.2.28** `__xor__()`

```
def PyClical.clifford.__xor__ (
    lhs,
    rhs )
```

Outer product.

```
>>> print(clifford("{1}") ^ clifford("{2}"))
{1,2}
>>> print(clifford(2) ^ clifford("{2}"))
2{2}
>>> print(clifford("{1}") ^ clifford("{1}"))
0
>>> print(clifford("{1}") ^ clifford("{1,2}"))
0
```

Definition at line 866 of file PyClical.pyx.

### 6.3.2.29 abs()

```
def PyClicl.clifford.abs (
    self )
```

Absolute value: square root of norm.

```
>>> clifford("1+{-1}+{1,2}+{1,2,3}").abs()
2.0
```

Definition at line 1175 of file PyClicl.pyx.

References [glucat.abs\(\)](#).

### 6.3.2.30 conj()

```
def PyClicl.clifford.conj (
    self )
```

Conjugation, reverse o involute == involute o reverse.

```
>>> print((clifford("{1}")).conj())
-{1}
>>> print((clifford("{2}") * clifford("{1}")).conj())
{1,2}
>>> print((clifford("{1}") * clifford("{2}")).conj())
-{1,2}
>>> print(clifford("1+{1}+{1,2}").conj())
1-{1}-{1,2}
```

Definition at line 1138 of file PyClicl.pyx.

References [PyClicl.index\\_set.instance](#), and [PyClicl.clifford.instance](#).

### 6.3.2.31 even()

```
def PyClicl.clifford.even (
    self )
```

Even part of multivector, sum of even grade terms.

```
>>> print(clifford("1+{1}+{1,2}").even())
1+{1,2}
```

Definition at line 1061 of file PyClicl.pyx.

References [PyClicl.index\\_set.instance](#), and [PyClicl.clifford.instance](#).

**6.3.2.32 frame()**

```
def PyClical.clifford.frame (
    self )
```

Subalgebra generated by all generators of terms of given multivector.

```
>>> print(clifford("1+3{-1}+2{1,2}+4{-2,7}").frame())
{-2,-1,1,2,7}
>>> s=clifford("1+3{-1}+2{1,2}+4{-2,7}").frame(); type(s)
<class 'PyClical.index_set'>
```

Definition at line 1224 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.33 inv()**

```
def PyClical.clifford.inv (
    self )
```

Geometric multiplicative inverse.

```
>>> x = clifford("{1}"); print(x.inv())
{1}
>>> x = clifford(2); print(x.inv())
0.5
>>> x = clifford("{1,2}"); print(x.inv())
-1,2}
```

Definition at line 926 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.34 involute()**

```
def PyClical.clifford.involute (
    self )
```

Main involution, each  $\{i\}$  is replaced by  $-\{i\}$  in each term, eg. `clifford("{1}") -> -clifford("{1}")`.

```
>>> print(clifford("{1}").involute())
-1
>>> print((clifford("{2}") * clifford("{1}")).involute())
-1,2
>>> print((clifford("{1}") * clifford("{2}")).involute())
1,2
>>> print(clifford("1+{1}+{1,2}").involute())
1-1+1,2}
```

Definition at line 1107 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

### 6.3.2.35 isinf()

```
def PyClical.clifford.isinf (
    self )
```

Check if a multivector contains any infinite values.

```
>>> clifford().isinf()
False
```

Definition at line 1206 of file PyClical.pyx.

References `PyClical.index_set.instance`, `PyClical.clifford.instance`, and `PyClical.clifford.isnan()`.

### 6.3.2.36 isnan()

```
def PyClical.clifford.isnan (
    self )
```

Check if a multivector contains any IEEE NaN values.

```
>>> clifford().isnan()
False
```

Definition at line 1215 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

Referenced by `PyClical.clifford.isinf()`.

### 6.3.2.37 max\_abs()

```
def PyClical.clifford.max_abs (
    self )
```

Maximum of absolute values of components of multivector: multivector infinity norm.

```
>>> clifford("1+{-1}+{1,2}+{1,2,3}").max_abs()
1.0
>>> clifford("3+2{1}+{1,2}").max_abs()
3.0
```

Definition at line 1184 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.



**6.3.2.38 norm()**

```
def PyClical.clifford.norm (
    self )

Norm == sum of squares of coordinates.

>>> clifford("1+{1}+{1,2}").norm()
3.0
>>> clifford("1+{-1}+{1,2}+{1,2,3}").norm()
4.0
```

Definition at line 1164 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.39 odd()**

```
def PyClical.clifford.odd (
    self )

Odd part of multivector, sum of odd grade terms.

>>> print(clifford("1+{1}+{1,2}").odd())
{1}
```

Definition at line 1070 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.40 outer\_pow()**

```
def PyClical.clifford.outer_pow (
    self,
    m )

Outer product power.

>>> x=clifford("2+{1}"); print(x.outer_pow(0))
1
>>> x=clifford("2+{1}"); print(x.outer_pow(1))
2+{1}
>>> x=clifford("2+{1}"); print(x.outer_pow(2))
4+4{1}
>>> print(clifford("1+{1}+{1,2}").outer_pow(3))
1+3{1}+3{1,2}
```

Definition at line 1004 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.41 pow()**

```
def PyClical.clifford.pow (
    self,
    m )

Power: self to the m.

>>> x=clifford("{1}"); print(x.pow(2))
1
>>> x=clifford("2"); print(x.pow(2))
4
>>> x=clifford("2+{1}"); print(x.pow(0))
1
>>> x=clifford("2+{1}"); print(x.pow(1))
2+{1}
>>> x=clifford("2+{1}"); print(x.pow(2))
5+4{1}
>>> print(clifford("1+{1}+{1,2}").pow(3))
1+3{1}+3{1,2}
>>> i=clifford("{1,2}"); print(exp(pi/2) * i.pow(i))
1
```

Definition at line 980 of file PyClical.pyx.

References `glucat.exp()`, `PyClical.index_set.instance`, `PyClical.clifford.instance`, and `glucat.log()`.

Referenced by `PyClical.clifford.__pow__()`.

**6.3.2.42 pure()**

```
def PyClical.clifford.pure (
    self )

Pure part.

>>> print(clifford("1+{1}+{1,2}").pure())
{1}+{1,2}
>>> print(clifford("{1,2}").pure())
{1,2}
```

Definition at line 1050 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.43 quad()**

```
def PyClical.clifford.quad (
    self )

Quadratic form == (rev(x)*x)(0).

>>> print(clifford("1+{1}+{1,2}").quad())
3.0
>>> print(clifford("1+{-1}+{1,2}+{1,2,3}").quad())
2.0
```

Definition at line 1153 of file PyClical.pyx.

References `PyClical.index_set.instance`, and `PyClical.clifford.instance`.

**6.3.2.44 reframe()**

```
def PyClical.clifford.reframe (
    self,
    ixt )
```

Put self into a larger frame, containing the union of self.frame() and index set ixt. This can be used to make multiplication faster, by multiplying within a common frame.

```
>>> clifford("2+3{1}").reframe(index_set({1,2,3}))
clifford("2+3{1}")
>>> s=index_set({1,2,3});t=index_set({-3,-2,-1});x=random_clifford(s); x.reframe(t).frame() == (s|t);
True
```

Definition at line 649 of file PyClical.pyx.

**6.3.2.45 reverse()**

```
def PyClical.clifford.reverse (
    self )
```

Reversion, eg. clifford("{1}")\*clifford("{2}") -> clifford("{2}")\*clifford("{1}").

```
>>> print(clifford("{1}").reverse())
{1}
>>> print((clifford("{2}") * clifford("{1}")).reverse())
{1,2}
>>> print((clifford("{1}") * clifford("{2}")).reverse())
-{1,2}
>>> print(clifford("1+{1}+{1,2}").reverse())
1+{1}-{1,2}
```

Definition at line 1123 of file PyClical.pyx.

References PyClical.index\_set.instance, and PyClical.clifford.instance.

**6.3.2.46 scalar()**

```
def PyClical.clifford.scalar (
    self )
```

Scalar part.

```
>>> clifford("1+{1}+{1,2}").scalar()
1.0
>>> clifford("{1,2}").scalar()
0.0
```

Definition at line 1039 of file PyClical.pyx.

References PyClical.index\_set.instance, and PyClical.clifford.instance.

### 6.3.2.47 truncated()

```
def PyClical.clifford.truncated (
    self,
    limit )
```

Remove all terms of self with relative size smaller than limit.

```
>>> clifford("1e8+{1}+1e-8{1,2}").truncated(1.0e-6)
clifford("100000000")
>>> clifford("1e4+{1}+1e-4{1,2}").truncated(1.0e-6)
clifford("10000+{1}")
```

Definition at line 1195 of file PyClical.pyx.

References PyClical.index\_set.instance, and PyClical.clifford.instance.

### 6.3.2.48 vector\_part()

```
def PyClical.clifford.vector_part (
    self,
    frm = None )
```

Vector part of multivector, as a Python list, with respect to frm.

```
>>> print(clifford("1+2{1}+3{2}+4{1,2}").vector_part())
[2.0, 3.0]
>>> print(clifford("1+2{1}+3{2}+4{1,2}").vector_part(index_set({-1,1,2})))
[0.0, 2.0, 3.0]
```

Definition at line 1079 of file PyClical.pyx.

References PyClical.index\_set.instance, and PyClical.clifford.instance.

## 6.3.3 Member Data Documentation

### 6.3.3.1 instance

PyClical.clifford.instance

Definition at line 592 of file PyClical.pyx.

Referenced by PyClical.clifford.\_\_call\_\_(), PyClical.clifford.\_\_dealloc\_\_(), PyClical.clifford.\_\_getitem\_\_(), PyClical.clifford.\_\_neg\_\_(), PyClical.clifford.conj(), PyClical.clifford.even(), PyClical.clifford.frame(), PyClical.clifford.inv(), PyClical.clifford.involute(), PyClical.clifford.isinf(), PyClical.clifford.isnan(), PyClical.clifford.max\_abs(), PyClical.clifford.norm(), PyClical.clifford.odd(), PyClical.clifford.outer\_pow(), PyClical.clifford.pow(), PyClical.clifford.pure(), PyClical.clifford.quad(), PyClical.clifford.reverse(), PyClical.clifford.scalar(), PyClical.clifford.truncated(), and PyClical.clifford.vector\_part().

The documentation for this class was generated from the following file:

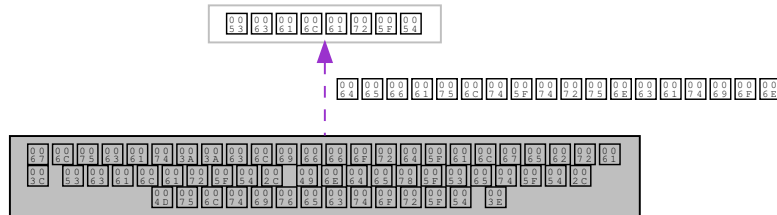
- [pyclical/PyClical.pyx](#)

## 6.4 glucat::clifford\_algebra< Scalar\_T, Index\_Set\_T, Multivector\_T > Class Template Reference

clifford\_algebra<> declares the operations of a Clifford algebra

```
#include <clifford_algebra.h>
```

Collaboration diagram for glucat::clifford\_algebra< Scalar\_T, Index\_Set\_T, Multivector\_T >:



### Public Types

- using [scalar\\_t](#) = Scalar\_T
- using [index\\_set\\_t](#) = Index\_Set\_T
- using [multivector\\_t](#) = Multivector\_T
- using [pair\\_t](#) = std::pair< const [index\\_set\\_t](#), Scalar\_T >
- using [vector\\_t](#) = std::vector< Scalar\_T >

### Public Member Functions

- virtual [~clifford\\_algebra](#) ()=default
- virtual auto [operator==](#) (const [multivector\\_t](#) &val) const -> bool=0  
*Test for equality of multivectors.*
- virtual auto [operator==](#) (const Scalar\_T &scr) const -> bool=0  
*Test for equality of multivector and scalar.*
- virtual auto [operator+=](#) (const [multivector\\_t](#) &rhs) -> [multivector\\_t](#) &=0  
*Geometric sum.*
- virtual auto [operator+=](#) (const Scalar\_T &scr) -> [multivector\\_t](#) &=0  
*Geometric sum of multivector and scalar.*
- virtual auto [operator-=](#) (const [multivector\\_t](#) &rhs) -> [multivector\\_t](#) &=0  
*Geometric difference.*
- virtual auto [operator-=](#) (const Scalar\_T &scr) -> [multivector\\_t](#) &=0  
*Geometric difference of multivector and scalar.*
- virtual auto [operator-](#) () const -> const [multivector\\_t](#)=0  
*Unary -.*
- virtual auto [operator\\*=](#) (const Scalar\_T &scr) -> [multivector\\_t](#) &=0  
*Product of multivector and scalar.*
- virtual auto [operator\\*=](#) (const [multivector\\_t](#) &rhs) -> [multivector\\_t](#) &=0  
*Geometric product.*

- virtual auto `operator%=(const multivector_t &rhs) -> multivector_t &=0`  
*Contraction.*
- virtual auto `operator &=(const multivector_t &rhs) -> multivector_t &=0`  
*Inner product.*
- virtual auto `operator^=(const multivector_t &rhs) -> multivector_t &=0`  
*Outer product.*
- virtual auto `operator/=(const Scalar_T &scr) -> multivector_t &=0`  
*Quotient of multivector and scalar.*
- virtual auto `operator/=(const multivector_t &rhs) -> multivector_t &=0`  
*Geometric quotient.*
- virtual auto `operator|=(const multivector_t &rhs) -> multivector_t &=0`  
*Transformation via twisted adjoint action.*
- virtual auto `inv () const -> const multivector_t=0`  
*Geometric multiplicative inverse.*
- virtual auto `pow (int m) const -> const multivector_t=0`  
*\*this to the m*
- virtual auto `outer_pow (int m) const -> const multivector_t=0`  
*Outer product power.*
- virtual auto `frame () const -> const index_set_t=0`  
*Subalgebra generated by all generators of terms of given multivector.*
- virtual auto `grade () const -> index_t=0`  
*Maximum of the grades of each term.*
- virtual auto `operator[] (const index_set_t ist) const -> Scalar_T=0`  
*Subscripting: map from index set to scalar coordinate.*
- virtual auto `operator() (index_t grade) const -> const multivector_t=0`  
*Pure grade-vector part.*
- virtual auto `scalar () const -> Scalar_T=0`  
*Scalar part.*
- virtual auto `pure () const -> const multivector_t=0`  
*Pure part.*
- virtual auto `even () const -> const multivector_t=0`  
*Even part of multivector, sum of even grade terms.*
- virtual auto `odd () const -> const multivector_t=0`  
*Odd part of multivector, sum of odd grade terms.*
- virtual auto `vector_part () const -> const vector_t=0`  
*Vector part of multivector, as a vector\_t with respect to frame()*
- virtual auto `vector_part (const index_set_t frm, const bool prechecked) const -> const vector_t=0`  
*Vector part of multivector, as a vector\_t with respect to frm.*
- virtual auto `involute () const -> const multivector_t=0`  
*Main involution, each {i} is replaced by -{i} in each term, eg. {1} -> -{1}.*
- virtual auto `reverse () const -> const multivector_t=0`  
*Reversion, eg. {1}\*{2} -> {2}\*{1}.*
- virtual auto `conj () const -> const multivector_t=0`  
*Conjugation, reverse o involute == involute o reverse.*
- virtual auto `quad () const -> Scalar_T=0`  
*Scalar\_T quadratic form == (rev(x)\*x)(0)*
- virtual auto `norm () const -> Scalar_T=0`  
*Scalar\_T norm == sum of norm of coordinates.*
- virtual auto `max_abs () const -> Scalar_T=0`  
*Maximum of absolute values of components of multivector: multivector infinity norm.*
- virtual auto `truncated (const Scalar_T &limit=default_truncation) const -> const multivector_t=0`

- Remove all terms with relative size smaller than limit.*
  - virtual auto `isinf` () const -> bool=0
  - Check if a multivector contains any infinite values.*
  - virtual auto `isnan` () const -> bool=0
  - Check if a multivector contains any IEEE NaN values.*
  - virtual void `write` (const std::string &msg="") const =0
  - Write formatted multivector to output.*
  - virtual void `write` (std::ofstream &ofile, const std::string &msg="") const =0
  - Write formatted multivector to file.*

## Static Public Member Functions

- static auto `classname` () -> const std::string

## Static Public Attributes

- static const `index_t v_lo` = index\_set\_t::v\_lo
- static const `index_t v_hi` = index\_set\_t::v\_hi
- static const Scalar\_T `default_truncation`
- Default for truncation.*

### 6.4.1 Detailed Description

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
class glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >
```

`clifford_algebra<>` declares the operations of a Clifford algebra

Definition at line 45 of file `clifford_algebra.h`.

### 6.4.2 Member Typedef Documentation

#### 6.4.2.1 `index_set_t`

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
using glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::index_set_t = Index_↔
Set_T
```

Definition at line 49 of file `clifford_algebra.h`.

#### 6.4.2.2 multivector\_t

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
using glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::multivector_t = Multivector<
_T
```

Definition at line 52 of file clifford\_algebra.h.

#### 6.4.2.3 pair\_t

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
using glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::pair_t = std::pair<const
index_set_t, Scalar_T>
```

Definition at line 53 of file clifford\_algebra.h.

#### 6.4.2.4 scalar\_t

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
using glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::scalar_t = Scalar_T
```

Definition at line 48 of file clifford\_algebra.h.

#### 6.4.2.5 vector\_t

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
using glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::vector_t = std::vector<Scalar<
_T>
```

Definition at line 54 of file clifford\_algebra.h.

### 6.4.3 Constructor & Destructor Documentation

#### 6.4.3.1 ~clifford\_algebra()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::~~clifford_algebra (
) [virtual], [default]
```



## 6.4.4 Member Function Documentation

### 6.4.4.1 classname()

```
template<typename Scalar_T , typename Index_Set_T , typename Multivector_T >
auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::classname ( ) -> const
std::string [static]
```

Definition at line 67 of file clifford\_algebra\_imp.h.

### 6.4.4.2 conj()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::conj ( ) const
-> const multivector_t [pure virtual]
```

Conjugation, reverse o involute == involute o reverse.

### 6.4.4.3 even()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::even ( ) const
-> const multivector_t [pure virtual]
```

Even part of multivector, sum of even grade terms.

### 6.4.4.4 frame()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::frame ( ) const
-> const index_set_t [pure virtual]
```

Subalgebra generated by all generators of terms of given multivector.

Referenced by glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi().

#### 6.4.4.5 grade()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::grade ( ) const
-> index\_t [pure virtual]
```

Maximum of the grades of each term.

#### 6.4.4.6 inv()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::inv ( ) const
-> const multivector\_t [pure virtual]
```

Geometric multiplicative inverse.

#### 6.4.4.7 involute()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::involute ( )
const -> const multivector\_t [pure virtual]
```

Main involution, each {i} is replaced by -{i} in each term, eg. {1} -> -{1}.

#### 6.4.4.8 isinf()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::isinf ( ) const
-> bool [pure virtual]
```

Check if a multivector contains any infinite values.

#### 6.4.4.9 isnan()

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::isnan ( ) const
-> bool [pure virtual]
```

Check if a multivector contains any IEEE NaN values.

**6.4.4.10 max\_abs()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::max_abs ( )
const -> Scalar_T [pure virtual]
```

Maximum of absolute values of components of multivector: multivector infinity norm.

**6.4.4.11 norm()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::norm ( ) const
-> Scalar_T [pure virtual]
```

Scalar\_T norm == sum of norm of coordinates.

**6.4.4.12 odd()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::odd ( ) const
-> const multivector_t [pure virtual]
```

Odd part of multivector, sum of odd grade terms.

**6.4.4.13 operator &=()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator&= (
    const multivector_t & rhs ) -> multivector_t & [pure virtual]
```

Inner product.

**6.4.4.14 operator%=( )**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator%=(
    const multivector_t & rhs ) -> multivector_t & [pure virtual]
```

Contraction.

**6.4.4.15 operator>()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator() (
    index\_t grade ) const -> const multivector\_t [pure virtual]
```

Pure grade-vector part.

**6.4.4.16 operator\*=( )** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator*= (
    const Scalar_T & scr ) -> multivector\_t & [pure virtual]
```

Product of multivector and scalar.

**6.4.4.17 operator\*=( )** [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator*= (
    const multivector\_t & rhs ) -> multivector\_t & [pure virtual]
```

Geometric product.

**6.4.4.18 operator+=( )** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator+= (
    const multivector\_t & rhs ) -> multivector\_t & [pure virtual]
```

Geometric sum.

**6.4.4.19 operator+=( )** [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator+= (
    const Scalar_T & scr ) -> multivector\_t & [pure virtual]
```

Geometric sum of multivector and scalar.

**6.4.4.20 operator-()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator- ( )
const -> const multivector_t [pure virtual]
```

Unary -.

**6.4.4.21 operator-=()** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator-= (
    const multivector_t & rhs ) -> multivector_t & [pure virtual]
```

Geometric difference.

**6.4.4.22 operator-=()** [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator-= (
    const Scalar_T & scr ) -> multivector_t & [pure virtual]
```

Geometric difference of multivector and scalar.

**6.4.4.23 operator/=(** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator/= (
    const Scalar_T & scr ) -> multivector_t & [pure virtual]
```

Quotient of multivector and scalar.

**6.4.4.24 operator/=(** [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator/= (
    const multivector_t & rhs ) -> multivector_t & [pure virtual]
```

Geometric quotient.

**6.4.4.25 operator==( )** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator==(
    const multivector\_t & val ) const -> bool [pure virtual]
```

Test for equality of multivectors.

**6.4.4.26 operator==( )** [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator==(
    const Scalar_T & scr ) const -> bool [pure virtual]
```

Test for equality of multivector and scalar.

**6.4.4.27 operator[]()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator[] (
    const index\_set\_t ist ) const -> Scalar_T [pure virtual]
```

Subscripting: map from index set to scalar coordinate.

**6.4.4.28 operator^=()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator^= (
    const multivector\_t & rhs ) -> multivector\_t & [pure virtual]
```

Outer product.

**6.4.4.29 operator" |=()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::operator|= (
    const multivector\_t & rhs ) -> multivector\_t & [pure virtual]
```

Transformation via twisted adjoint action.

**6.4.4.30 outer\_pow()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::outer_pow (
    int m ) const -> const multivector_t [pure virtual]
```

Outer product power.

**6.4.4.31 pow()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::pow (
    int m ) const -> const multivector_t [pure virtual]
```

\*this to the m

**6.4.4.32 pure()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::pure ( ) const
-> const multivector_t [pure virtual]
```

Pure part.

**6.4.4.33 quad()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::quad ( ) const
-> Scalar_T [pure virtual]
```

Scalar\_T quadratic form == (rev(x)\*x)(0)

**6.4.4.34 reverse()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::reverse ( )
const -> const multivector_t [pure virtual]
```

Reversion, eg. {1}\*{2} -> {2}\*{1}.

**6.4.4.35 scalar()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::scalar ( )
const -> Scalar_T [pure virtual]
```

Scalar part.

**6.4.4.36 truncated()**

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::truncated (
    const Scalar_T & limit = default\_truncation ) const -> const multivector\_t [pure
virtual]
```

Remove all terms with relative size smaller than limit.

**6.4.4.37 vector\_part()** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::vector_part ( )
const -> const vector\_t [pure virtual]
```

Vector part of multivector, as a [vector\\_t](#) with respect to [frame\(\)](#)

**6.4.4.38 vector\_part()** [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual auto glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::vector_part (
    const index\_set\_t frm,
    const bool prechecked ) const -> const vector\_t [pure virtual]
```

Vector part of multivector, as a [vector\\_t](#) with respect to *frm*.

**6.4.4.39 write()** [1/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual void glucat::clifford\_algebra< Scalar_T, Index_Set_T, Multivector_T >::write (
    const std::string & msg = "" ) const [pure virtual]
```

Write formatted multivector to output.



#### 6.4.4.40 write() [2/2]

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
virtual void glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::write (
    std::ostream & ofile,
    const std::string & msg = "" ) const [pure virtual]
```

Write formatted multivector to file.

### 6.4.5 Member Data Documentation

#### 6.4.5.1 default\_truncation

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
const Scalar_T glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::default_↵
truncation [static]
```

Default for truncation.

Definition at line 59 of file clifford\_algebra.h.

#### 6.4.5.2 v\_hi

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
const index_t glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::v_hi = index↵
_set_t::v_hi [static]
```

Definition at line 51 of file clifford\_algebra.h.

#### 6.4.5.3 v\_lo

```
template<typename Scalar_T, typename Index_Set_T, typename Multivector_T>
const index_t glucat::clifford_algebra< Scalar_T, Index_Set_T, Multivector_T >::v_lo = index↵
_set_t::v_lo [static]
```

Definition at line 50 of file clifford\_algebra.h.

The documentation for this class was generated from the following files:

- glucat/clifford\_algebra.h
- glucat/clifford\_algebra\_imp.h

## 6.5 glucat::compare\_types< LHS\_T, RHS\_T > Class Template Reference

Type comparison.

```
#include <global.h>
```

### Public Types

- enum { [are\\_same](#) = false }

#### 6.5.1 Detailed Description

```
template<typename LHS_T, typename RHS_T>  
class glucat::compare_types< LHS_T, RHS_T >
```

Type comparison.

Definition at line 54 of file global.h.

#### 6.5.2 Member Enumeration Documentation

##### 6.5.2.1 anonymous enum

```
template<typename LHS_T , typename RHS_T >  
anonymous enum
```

##### Enumerator

<a href="#">are_same</a>	
--------------------------	--

Definition at line 57 of file global.h.

The documentation for this class was generated from the following file:

- glucat/[global.h](#)

## 6.6 glucat::compare\_types< T, T > Class Template Reference

```
#include <global.h>
```

### Public Types

- enum { [are\\_same](#) = true }

### 6.6.1 Detailed Description

```
template<typename T>
class glucat::compare_types< T, T >
```

Definition at line 60 of file global.h.

### 6.6.2 Member Enumeration Documentation

#### 6.6.2.1 anonymous enum

```
template<typename T >
anonymous enum
```

##### Enumerator

are_same	
----------	--

Definition at line 63 of file global.h.

The documentation for this class was generated from the following file:

- [glucat/global.h](#)

## 6.7 glucat::control\_t Class Reference

Parameters to control tests.

```
#include <control.h>
```

### Public Member Functions

- int [call](#) ([intfn](#) f) const  
*Call a function that returns int.*
- int [call](#) ([intintfn](#) f, int arg) const  
*Call a function of int that returns int.*

### Static Public Member Functions

- static const [control\\_t](#) & [control](#) (int argc, char \*\*argv)
- static bool [verbose](#) ()  
*Produce more detailed output from tests.*

## Private Member Functions

- bool [valid](#) () const
- bool [catch\\_exceptions](#) () const
- [control\\_t](#) (int argc, char \*\*argv)  
*Constructor from program arguments.*
- [control\\_t](#) ()=default
- [~control\\_t](#) ()=default
- [control\\_t](#) (const [control\\_t](#) &)=delete
- [control\\_t](#) & [operator=](#) (const [control\\_t](#) &)=delete

## Private Attributes

- bool [m\\_valid](#)  
*Test parameters are valid.*
- bool [m\\_catch\\_exceptions](#)  
*Catch exceptions.*

## Static Private Attributes

- static bool [m\\_verbose\\_output](#) = false  
*Produce more detailed output from tests.*

## Friends

- class [friend\\_for\\_private\\_destructor](#)

### 6.7.1 Detailed Description

Parameters to control tests.

Definition at line 39 of file control.h.

### 6.7.2 Constructor & Destructor Documentation

#### 6.7.2.1 [control\\_t](#)() [1/3]

```
glucat::control_t::control_t (
    int argc,
    char ** argv ) [private]
```

Constructor from program arguments.

Test control constructor from program arguments.

Definition at line 89 of file control.h.

References [GLUCAT\\_PACKAGE\\_NAME](#), [GLUCAT\\_VERSION](#), [m\\_catch\\_exceptions](#), [m\\_valid](#), [m\\_verbose\\_output](#), and [valid\(\)](#).

### 6.7.2.2 control\_t() [2/3]

```
glucat::control_t::control_t ( ) [private], [default]
```

### 6.7.2.3 ~control\_t()

```
glucat::control_t::~~control_t ( ) [private], [default]
```

### 6.7.2.4 control\_t() [3/3]

```
glucat::control_t::control_t (
    const control_t & ) [private], [delete]
```

## 6.7.3 Member Function Documentation

### 6.7.3.1 call() [1/2]

```
int glucat::control_t::call (
    intfn f ) const [inline]
```

Call a function that returns int.

Definition at line 137 of file control.h.

References [catch\\_exceptions\(\)](#), [glucat::try\\_catch\(\)](#), and [valid\(\)](#).

### 6.7.3.2 call() [2/2]

```
int glucat::control_t::call (
    intintfn f,
    int arg ) const [inline]
```

Call a function of int that returns int.

Definition at line 151 of file control.h.

References [catch\\_exceptions\(\)](#), [glucat::try\\_catch\(\)](#), and [valid\(\)](#).

#### 6.7.3.3 catch\_exceptions()

```
bool glucat::control_t::catch_exceptions ( ) const [inline], [private]
```

Definition at line 49 of file control.h.

References `m_catch_exceptions`.

Referenced by `call()`.

#### 6.7.3.4 control()

```
static const control_t& glucat::control_t::control (
    int argc,
    char ** argv ) [inline], [static]
```

Single instance Ref: Scott Meyers, "Effective C++" Second Edition, Addison-Wesley, 1998.

Definition at line 71 of file control.h.

#### 6.7.3.5 operator=()

```
control_t& glucat::control_t::operator= (
    const control_t & ) [private], [delete]
```

#### 6.7.3.6 valid()

```
bool glucat::control_t::valid ( ) const [inline], [private]
```

Definition at line 44 of file control.h.

References `m_valid`.

Referenced by `call()`, and `control_t()`.

#### 6.7.3.7 verbose()

```
static bool glucat::control_t::verbose ( ) [inline], [static]
```

Produce more detailed output from tests.

Definition at line 80 of file control.h.

References `m_verbose_output`.

## 6.7.4 Friends And Related Function Documentation

### 6.7.4.1 friend\_for\_private\_destructor

```
friend class friend_for_private_destructor [friend]
```

Friend declaration to avoid compiler warning: "... only defines a private destructor and has no friends" Ref: Carlos O'Ryan, ACE <http://doc.ece.uci.edu>

Definition at line 67 of file control.h.

## 6.7.5 Member Data Documentation

### 6.7.5.1 m\_catch\_exceptions

```
bool glucat::control_t::m_catch_exceptions [private]
```

Catch exceptions.

Definition at line 48 of file control.h.

Referenced by `catch_exceptions()`, and `control_t()`.

### 6.7.5.2 m\_valid

```
bool glucat::control_t::m_valid [private]
```

Test parameters are valid.

Definition at line 43 of file control.h.

Referenced by `control_t()`, and `valid()`.

### 6.7.5.3 m\_verbose\_output

```
bool glucat::control_t::m_verbose_output = false [static], [private]
```

Produce more detailed output from tests.

Definition at line 53 of file control.h.

Referenced by `control_t()`, and `verbose()`.

The documentation for this class was generated from the following file:

- [test/control.h](#)

## 6.8 `glucat::CTAssertion< bool >` Struct Template Reference

Compile time assertion.

```
#include <global.h>
```

### 6.8.1 Detailed Description

```
template<bool>
struct glucat::CTAssertion< bool >
```

Compile time assertion.

Definition at line 46 of file `global.h`.

The documentation for this struct was generated from the following file:

- [glucat/global.h](#)

## 6.9 `glucat::CTAssertion< true >` Struct Template Reference

```
#include <global.h>
```

### 6.9.1 Detailed Description

```
template<>
struct glucat::CTAssertion< true >
```

Definition at line 47 of file `global.h`.

The documentation for this struct was generated from the following file:

- [glucat/global.h](#)

## 6.10 `glucat::numeric_traits< Scalar_T >::demoted<>` Struct Template Reference

Demoted type for long double.

```
#include <promotion.h>
```

### Public Types

- using `type` = float
- using `type` = float



### 6.10.1 Detailed Description

```
template<typename Scalar_T>
template<>
struct glucat::numeric_traits< Scalar_T >::demoted<>
```

Demoted type for long double.

Demoted type.

Definition at line 76 of file promotion.h.

### 6.10.2 Member Typedef Documentation

#### 6.10.2.1 type [1/2]

```
template<typename Scalar_T >
using glucat::numeric_traits< Scalar_T >::demoted<>::type = float
```

Definition at line 78 of file promotion.h.

#### 6.10.2.2 type [2/2]

```
template<typename Scalar_T >
using glucat::numeric_traits< Scalar_T >::demoted<>::type = float
```

Definition at line 148 of file scalar.h.

The documentation for this struct was generated from the following files:

- glucat/[promotion.h](#)
- glucat/[scalar.h](#)

## 6.11 glucat::matrix::eig\_genus< Matrix\_T > Struct Template Reference

Structure containing classification of eigenvalues.

```
#include <matrix.h>
```

### Public Types

- using [Scalar\\_T](#) = typename Matrix\_T::value\_type

## Public Attributes

- bool `m_is_singular` = false  
*Is the matrix singular?*
- `eig_case_t` `m_eig_case` = `safe_eigs`  
*What kind of eigenvalues does the matrix contain?*
- `Scalar_T` `m_safe_arg` = `Scalar_T`(0)  
*Argument such that  $\exp(\pi - m\_safe\_arg)$  lies between arguments of eigenvalues.*

### 6.11.1 Detailed Description

```
template<typename Matrix_T>
struct glucat::matrix::eig_genus< Matrix_T >
```

Structure containing classification of eigenvalues.

Definition at line 140 of file matrix.h.

### 6.11.2 Member Typedef Documentation

#### 6.11.2.1 `Scalar_T`

```
template<typename Matrix_T>
using glucat::matrix::eig_genus< Matrix_T >::Scalar_T = typename Matrix_T::value_type
```

Definition at line 142 of file matrix.h.

### 6.11.3 Member Data Documentation

#### 6.11.3.1 `m_eig_case`

```
template<typename Matrix_T>
eig_case_t glucat::matrix::eig_genus< Matrix_T >::m_eig_case = safe_eigs
```

What kind of eigenvalues does the matrix contain?

Definition at line 146 of file matrix.h.

Referenced by `glucat::matrix::classify_eigenvalues()`.

## 6.11.3.2 m\_is\_singular

```
template<typename Matrix_T>
bool glucat::matrix::eig_genus< Matrix_T >::m_is_singular = false
```

Is the matrix singular?

Definition at line 144 of file matrix.h.

Referenced by glucat::matrix::classify\_eigenvalues().

## 6.11.3.3 m\_safe\_arg

```
template<typename Matrix_T>
Scalar_T glucat::matrix::eig_genus< Matrix_T >::m_safe_arg = Scalar_T(0)
```

Argument such that  $\exp(\pi \cdot m\_safe\_arg)$  lies between arguments of eigenvalues.

Definition at line 148 of file matrix.h.

Referenced by glucat::matrix::classify\_eigenvalues().

The documentation for this struct was generated from the following file:

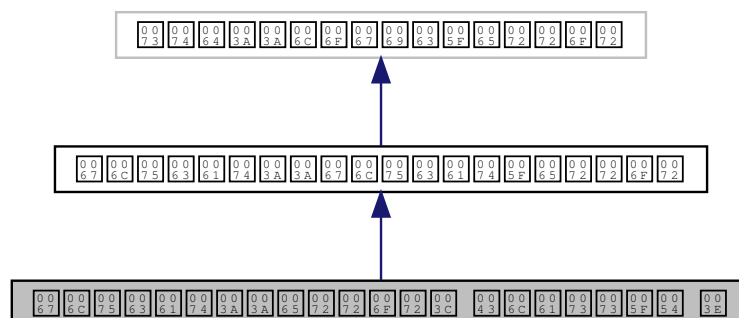
- glucat/[matrix.h](#)

## 6.12 glucat::error&lt; Class\_T &gt; Class Template Reference

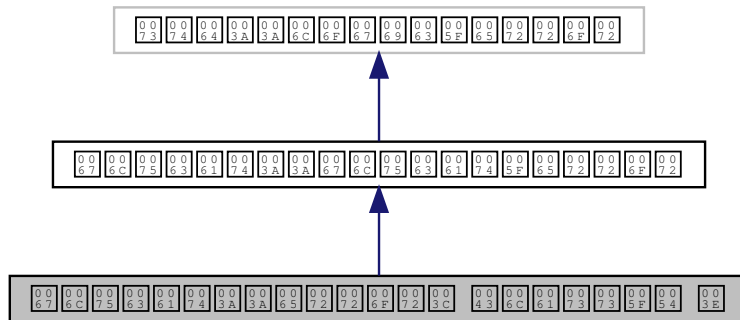
Specific exception class.

```
#include <errors.h>
```

Inheritance diagram for glucat::error< Class\_T >:



Collaboration diagram for `glucat::error< Class_T >`:



## Public Member Functions

- [error](#) (const std::string &msg)  
*Specific exception class.*
- [error](#) (const std::string &context, const std::string &msg)
- auto [heading](#) () const noexcept -> const std::string override
- auto [classname](#) () const noexcept -> const std::string override
- void [print\\_error\\_msg](#) () const override

## Additional Inherited Members

### 6.12.1 Detailed Description

```
template<class Class_T>
class glucat::error< Class_T >
```

Specific exception class.

Definition at line 56 of file errors.h.

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 [error\(\)](#) [1/2]

```
template<class Class_T >
glucat::error< Class_T >::error (
    const std::string & msg )
```

Specific exception class.

Definition at line 45 of file errors\_imp.h.

### 6.12.2.2 error() [2/2]

```
template<class Class_T >
glucat::error< Class_T >::error (
    const std::string & context,
    const std::string & msg )
```

Definition at line 51 of file errors\_imp.h.

## 6.12.3 Member Function Documentation

### 6.12.3.1 classname()

```
template<class Class_T >
auto glucat::error< Class_T >::classname ( ) const -> const std::string [override], [virtual],
[noexcept]
```

Implements [glucat::glucat\\_error](#).

Definition at line 64 of file errors\_imp.h.

### 6.12.3.2 heading()

```
template<class Class_T >
auto glucat::error< Class_T >::heading ( ) const -> const std::string [override], [virtual],
[noexcept]
```

Implements [glucat::glucat\\_error](#).

Definition at line 58 of file errors\_imp.h.

### 6.12.3.3 print\_error\_msg()

```
template<class Class_T >
void glucat::error< Class_T >::print_error_msg ( ) const [override], [virtual]
```

Implements [glucat::glucat\\_error](#).

Definition at line 70 of file errors\_imp.h.

The documentation for this class was generated from the following files:

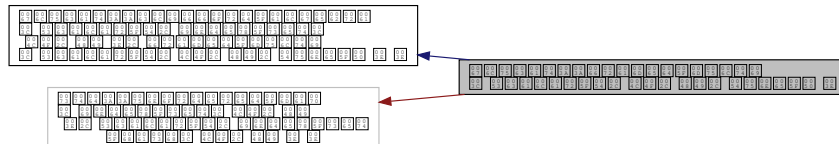
- [glucat/errors.h](#)
- [glucat/errors\\_imp.h](#)

## 6.13 glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P > Class Template Reference

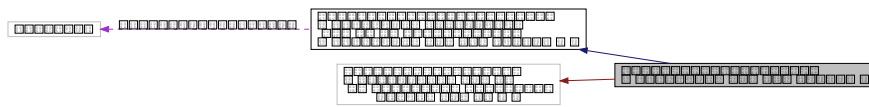
A framed\_multi<Scalar\_T,LO,HI,Tune\_P> is a framed approximation to a multivector.

```
#include <framed_multi.h>
```

Inheritance diagram for glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >:



Collaboration diagram for glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >:



### Classes

- class [hash\\_size\\_t](#)
- class [var\\_term](#)

*Variable term.*

### Public Types

- using [multivector\\_t](#) = [framed\\_multi](#)
- using [framed\\_multi\\_t](#) = [multivector\\_t](#)
- using [scalar\\_t](#) = [Scalar\\_T](#)
- using [tune\\_p](#) = [Tune\\_P](#)
- using [index\\_set\\_t](#) = [index\\_set](#)< [LO](#), [HI](#) >
- using [term\\_t](#) = std::pair< const [index\\_set\\_t](#), [Scalar\\_T](#) >
- using [vector\\_t](#) = std::vector< [Scalar\\_T](#) >
- using [error\\_t](#) = [error](#)< [multivector\\_t](#) >
- using [matrix\\_multi\\_t](#) = [matrix\\_multi](#)< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) >

## Public Member Functions

- `~framed_multi ()` override=default  
*Destructor.*
- `framed_multi ()`  
*Default constructor.*
- `template<typename Other_Scalar_T >`  
`framed_multi (const framed_multi< Other_Scalar_T, LO, HI, Tune_P > &val)`  
*Construct a multivector from a multivector with a different scalar type.*
- `template<typename Other_Scalar_T >`  
`framed_multi (const framed_multi< Other_Scalar_T, LO, HI, Tune_P > &val, const index_set_t frm, const bool prechecked=false)`  
*Construct a multivector, within a given frame, from a given multivector.*
- `framed_multi (const framed_multi_t &val, const index_set_t frm, const bool prechecked=false)`  
*Construct a multivector, within a given frame, from a given multivector.*
- `framed_multi (const index_set_t ist, const Scalar_T &crd=Scalar_T(1))`  
*Construct a multivector from an index set and a scalar coordinate.*
- `framed_multi (const index_set_t ist, const Scalar_T &crd, const index_set_t frm, const bool prechecked=false)`  
*Construct a multivector, within a given frame, from an index set and a scalar coordinate.*
- `framed_multi (const Scalar_T &scr, const index_set_t frm=index_set_t())`  
*Construct a multivector from a scalar (within a frame, if given)*
- `framed_multi (const int scr, const index_set_t frm=index_set_t())`  
*Construct a multivector from an int (within a frame, if given)*
- `framed_multi (const vector_t &vec, const index_set_t frm, const bool prechecked=false)`  
*Construct a multivector, within a given frame, from a given vector.*
- `framed_multi (const std::string &str)`  
*Construct a multivector from a string: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `framed_multi (const std::string &str, const index_set_t frm, const bool prechecked=false)`  
*Construct a multivector, within a given frame, from a string: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `framed_multi (const char *str)`  
*Construct a multivector from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `framed_multi (const char *str, const index_set_t frm, const bool prechecked=false)`  
*Construct a multivector, within a given frame, from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `template<typename Other_Scalar_T >`  
`framed_multi (const matrix_multi< Other_Scalar_T, LO, HI, Tune_P > &val)`  
*Construct a multivector from a matrix\_multi\_t.*
- `template<typename Other_Scalar_T >`  
`auto fast_matrix_multi (const index_set_t frm) const -> const matrix_multi< Other_Scalar_T, LO, HI, Tune_P >`  
*Use generalized FFT to construct a matrix\_multi\_t.*
- `auto fast_framed_multi () const -> const framed_multi_t`  
*Use inverse generalized FFT to construct a framed\_multi\_t.*
- `_GLUCAT_CLIFFORD_ALGEBRA_OPERATIONS auto nbr_terms () const -> unsigned long`  
*Number of terms.*
- `auto operator+= (const term_t &term) -> multivector_t &`  
*Add a term, if non-zero.*

## Static Public Member Functions

- `static auto classname () -> const std::string`  
*Class name used in messages.*
- `static auto random (const index_set_t frm, Scalar_T fill=Scalar_T(1)) -> const multivector_t`  
*Random multivector within a frame.*

## Private Types

- using `var_term_t` = class `var_term`
- using `matrix_t` = typename `matrix_multi_t::matrix_t`
- using `sorted_map_t` = `std::map< index_set_t, Scalar_T, std::less< const index_set_t > >`
- using `map_t` = `std::unordered_map< index_set_t, Scalar_T, index_set_hash< LO, HI > >`
- using `framed_pair_t` = `std::pair< const multivector_t, const multivector_t >`
- using `size_type` = typename `map_t::size_type`
- using `iterator` = typename `map_t::iterator`
- using `const_iterator` = typename `map_t::const_iterator`

## Private Member Functions

- `framed_multi` (const `hash_size_t` &hash\_size)  
*Private constructor using hash\_size.*
- auto `fold` (const `index_set_t` frm) const -> `multivector_t`  
*Subalgebra isomorphism: fold each term within the given frame.*
- auto `unfold` (const `index_set_t` frm) const -> `multivector_t`  
*Subalgebra isomorphism: unfold each term within the given frame.*
- auto `centre_pm4_qp4` (`index_t` &p, `index_t` &q) -> `multivector_t` &  
*Subalgebra isomorphism:  $R_{\{p,q\}}$  to  $R_{\{p-4,q+4\}}$ .*
- auto `centre_pp4_qm4` (`index_t` &p, `index_t` &q) -> `multivector_t` &  
*Subalgebra isomorphism:  $R_{\{p,q\}}$  to  $R_{\{p+4,q-4\}}$ .*
- auto `centre_qp1_pm1` (`index_t` &p, `index_t` &q) -> `multivector_t` &  
*Subalgebra isomorphism:  $R_{\{p,q\}}$  to  $R_{\{q+1,p-1\}}$ .*
- auto `divide` (const `index_set_t` ist) const -> const `framed_pair_t`  
*Divide multivector into part divisible by `index_set` and remainder.*
- auto `fast` (const `index_t` level, const bool odd) const -> const `matrix_t`  
*Generalized FFT from `multivector_t` to `matrix_t`.*

## Friends

- template<typename Other\_Scalar\_T, const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
class `matrix_multi`
- template<typename Other\_Scalar\_T, const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
class `framed_multi`
- auto `operator*` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> const `multivector_t`
- auto `operator^` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> const `multivector_t`
- auto `operator &` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> const `multivector_t`
- auto `operator%` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> const `multivector_t`
- auto `star` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> `Scalar_T`
- auto `operator/` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> const `multivector_t`
- auto `operator|` (const `multivector_t` &lhs, const `multivector_t` &rhs) -> const `multivector_t`
- auto `operator>>` (std::istream &s, `multivector_t` &val) -> std::istream &
- auto `operator<<` (std::ostream &os, const `multivector_t` &val) -> std::ostream &
- auto `operator<<` (std::ostream &os, const `term_t` &term) -> std::ostream &
- auto `exp` (const `multivector_t` &val) -> const `multivector_t`



## Additional Inherited Members

### 6.13.1 Detailed Description

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P =
tuning<>>
class glucat::framed_multi< Scalar_T, LO, HI, Tune_P >
```

A framed\_multi<Scalar\_T,LO,HI,Tune\_P> is a framed approximation to a multivector.

Definition at line 56 of file framed\_multi.h.

### 6.13.2 Member Typedef Documentation

#### 6.13.2.1 const\_iterator

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::const_iterator = typename map_t::const_iterator [private]
```

Definition at line 167 of file framed\_multi.h.

#### 6.13.2.2 error\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::error_t = error<multivector_t>
```

Definition at line 138 of file framed\_multi.h.

#### 6.13.2.3 framed\_multi\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi_t = multivector_t
```

Definition at line 132 of file framed\_multi.h.

#### 6.13.2.4 framed\_pair\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_pair_t = std::pair<const multivector_t,
const multivector_t> [private]
```

Definition at line 164 of file framed\_multi.h.

#### 6.13.2.5 index\_set\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::index_set_t = index_set<LO, HI>
```

Definition at line 135 of file framed\_multi.h.

#### 6.13.2.6 iterator

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::iterator = typename map_t::iterator
[private]
```

Definition at line 166 of file framed\_multi.h.

#### 6.13.2.7 map\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::map_t = std::unordered_map<index_set_t,
Scalar_T, index_set_hash<LO, HI> > [private]
```

Definition at line 150 of file framed\_multi.h.

#### 6.13.2.8 matrix\_multi\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi_t = matrix_multi<Scalar↵
_T, LO, HI, Tune_P >
```

Definition at line 139 of file framed\_multi.h.

#### 6.13.2.9 matrix\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::matrix_t = typename matrix_multi_t::matrix_t
[private]
```

Definition at line 148 of file framed\_multi.h.

#### 6.13.2.10 multivector\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::multivector_t = framed_multi
```

Definition at line 131 of file framed\_multi.h.

#### 6.13.2.11 scalar\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::scalar_t = Scalar_T
```

Definition at line 133 of file framed\_multi.h.

#### 6.13.2.12 size\_type

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::size_type = typename map_t::size_type
[private]
```

Definition at line 165 of file framed\_multi.h.

#### 6.13.2.13 sorted\_map\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::sorted_map_t = std::map< index_set_t,
Scalar_T, std::less<const index_set_t> > [private]
```

Definition at line 149 of file framed\_multi.h.

#### 6.13.2.14 term\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::term_t = std::pair<const index_set_t, Scalar_T>
```

Definition at line 136 of file framed\_multi.h.

#### 6.13.2.15 tune\_p

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::tune_p = Tune_P
```

Definition at line 134 of file framed\_multi.h.

#### 6.13.2.16 var\_term\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term_t = class var_term [private]
```

Definition at line 147 of file framed\_multi.h.

#### 6.13.2.17 vector\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::vector_t = std::vector<Scalar_T>
```

Definition at line 137 of file framed\_multi.h.

### 6.13.3 Constructor & Destructor Documentation

#### 6.13.3.1 ~framed\_multi()

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::~~framed_multi ( ) [override], [default]
```

Destructor.

**6.13.3.2 framed\_multi()** [1/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi ( )
```

Default constructor.

Definition at line 60 of file framed\_multi\_imp.h.

Referenced by glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi().

**6.13.3.3 framed\_multi()** [2/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const hash_size_t & hash_size ) [private]
```

Private constructor using hash\_size.

Definition at line 67 of file framed\_multi\_imp.h.

**6.13.3.4 framed\_multi()** [3/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const framed_multi< Other_Scalar_T, LO, HI, Tune_P > & val )
```

Construct a multivector from a multivector with a different scalar type.

Definition at line 75 of file framed\_multi\_imp.h.

**6.13.3.5 framed\_multi()** [4/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const framed_multi< Other_Scalar_T, LO, HI, Tune_P > & val,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a given multivector.

Definition at line 86 of file framed\_multi\_imp.h.

References glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, framed\_multi< Scalar\_T, LO, HI, Tune\_P > >::frame().

**6.13.3.6 framed\_multi()** [5/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const framed_multi_t & val,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a given multivector.

Definition at line 99 of file framed\_multi\_imp.h.

References glucat::clifford\_algebra< Scalar\_T, Index\_Set\_T, Multivector\_T >::frame().

**6.13.3.7 framed\_multi()** [6/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const index_set_t ist,
    const Scalar_T & crd = Scalar_T(1) )
```

Construct a multivector from an index set and a scalar coordinate.

Definition at line 112 of file framed\_multi\_imp.h.

References PyClical::ist.

**6.13.3.8 framed\_multi()** [7/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const index_set_t ist,
    const Scalar_T & crd,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from an index set and a scalar coordinate.

Definition at line 122 of file framed\_multi\_imp.h.

References PyClical::ist.

**6.13.3.9 framed\_multi()** [8/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const Scalar_T & scr,
    const index_set_t frm = index_set_t() )
```

Construct a multivector from a scalar (within a frame, if given)

Definition at line 135 of file framed\_multi\_imp.h.

**6.13.3.10 framed\_multi()** [9/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const int scr,
    const index_set_t frm = index_set_t() )
```

Construct a multivector from an int (within a frame, if given)

Definition at line 145 of file framed\_multi\_imp.h.

**6.13.3.11 framed\_multi()** [10/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const vector_t & vec,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a given vector.

Definition at line 155 of file framed\_multi\_imp.h.

References glucat::index\_set< LO, HI >::count(), glucat::index\_set< LO, HI >::max(), and glucat::index\_set< LO, HI >::min().

**6.13.3.12 framed\_multi()** [11/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const std::string & str )
```

Construct a multivector from a string: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 177 of file framed\_multi\_imp.h.

**6.13.3.13 framed\_multi()** [12/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const std::string & str,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a string: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 193 of file framed\_multi\_imp.h.

**6.13.3.14 framed\_multi()** [13/15]

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const char * str ) [inline]
```

Construct a multivector from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 209 of file framed\_multi.h.

References glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi().

**6.13.3.15 framed\_multi()** [14/15]

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const char * str,
    const index_set_t frm,
    const bool prechecked = false ) [inline]
```

Construct a multivector, within a given frame, from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 212 of file framed\_multi.h.

References glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi().

**6.13.3.16 framed\_multi()** [15/15]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi (
    const matrix_multi< Other_Scalar_T, LO, HI, Tune_P > & val )
```

Construct a multivector from a matrix\_multi\_t.

Definition at line 206 of file framed\_multi\_imp.h.

References \_GLUCAT\_HASH\_SIZE\_T, glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::basis\_element(), PyClical::e(), glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::frame(), PyClical::ist, glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix, glucat::matrix::nnz(), and glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >::norm().



### 6.13.4 Member Function Documentation

#### 6.13.4.1 centre\_pm4\_qp4()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::centre_pm4_qp4 (
    index_t & p,
    index_t & q ) -> multivector_t& [private]
```

Subalgebra isomorphism:  $R_{\{p,q\}}$  to  $R_{\{p-4,q+4\}}$ .

Definition at line 1470 of file framed\_multi\_imp.h.

References PyClical::ist.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::fast\_framed\_multi().

#### 6.13.4.2 centre\_pp4\_qm4()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::centre_pp4_qm4 (
    index_t & p,
    index_t & q ) -> multivector_t& [private]
```

Subalgebra isomorphism:  $R_{\{p,q\}}$  to  $R_{\{p+4,q-4\}}$ .

Definition at line 1512 of file framed\_multi\_imp.h.

References PyClical::ist.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::fast\_framed\_multi().

#### 6.13.4.3 centre\_qp1\_pm1()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::centre_qp1_pm1 (
    index_t & p,
    index_t & q ) -> multivector_t& [private]
```

Subalgebra isomorphism:  $R_{\{p,q\}}$  to  $R_{\{q+1,p-1\}}$ .

Definition at line 1554 of file framed\_multi\_imp.h.

References PyClical::ist.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::fast\_framed\_multi().

#### 6.13.4.4 classname()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::classname ( ) -> const std::string
[static]
```

Class name used in messages.

Definition at line 51 of file framed\_multi\_imp.h.

#### 6.13.4.5 divide()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::divide (
    const index_set_t ist ) const -> const framed_pair_t [private]
```

Divide multivector into part divisible by [index\\_set](#) and remainder.

Divide multivector into quotient with terms divisible by index set, and remainder.

Definition at line 1587 of file framed\_multi\_imp.h.

References `PyClical::ist`.

#### 6.13.4.6 fast()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fast (
    const index_t level,
    const bool odd ) const -> const matrix_t [private]
```

Generalized FFT from multivector\_t to matrix\_t.

Definition at line 1603 of file framed\_multi\_imp.h.

References `glucat::matrix::kron()`, `glucat::odd()`, and `glucat::scalar()`.

#### 6.13.4.7 fast\_framed\_multi()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fast_framed_multi ( ) const -> const
framed_multi_t [inline]
```

Use inverse generalized FFT to construct a framed\_multi\_t.

Definition at line 1701 of file framed\_multi\_imp.h.

## 6.13.4.8 fast\_matrix\_multi()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fast_matrix_multi (
    const index_set_t frm ) const -> const matrix_multi<Other_Scalar_T,LO,HI,Tune_P
>
```

Use generalized FFT to construct a matrix\_multi\_t.

Definition at line 1669 of file framed\_multi\_imp.h.

References glucat::gen::offset\_to\_super, and glucat::pos\_mod().

## 6.13.4.9 fold()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::fold (
    const index_set_t frm ) const -> multivector_t [private]
```

Subalgebra isomorphism: fold each term within the given frame.

Definition at line 1435 of file framed\_multi\_imp.h.

## 6.13.4.10 nbr\_terms()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::nbr_terms ( ) const -> unsigned long
```

Number of terms.

Definition at line 1357 of file framed\_multi\_imp.h.

## 6.13.4.11 operator+=( )

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::operator+= (
    const term_t & term ) -> multivector_t& [inline]
```

Add a term, if non-zero.

Insert a term into a multivector, add terms with same index set.

Geometric sum.

Geometric sum of multivector and scalar.

Definition at line 296 of file framed\_multi\_imp.h.

#### 6.13.4.12 random()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::random (
    const index_set_t frm,
    Scalar_T fill = Scalar_T(1) ) -> const multivector_t [static]
```

Random multivector within a frame.

Definition at line 1059 of file framed\_multi\_imp.h.

References PyClical::fill, and glucat::sqrt().

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::random().

#### 6.13.4.13 unfold()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::unfold (
    const index_set_t frm ) const -> multivector_t [private]
```

Subalgebra isomorphism: unfold each term within the given frame.

Definition at line 1452 of file framed\_multi\_imp.h.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::fast\_framed\_multi().

### 6.13.5 Friends And Related Function Documentation

#### 6.13.5.1 exp

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI,
typename Tune_P = tuning<>>
auto exp (
    const multivector_t & val ) -> const multivector_t [friend]
```

#### 6.13.5.2 framed\_multi

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI,
typename Tune_P = tuning<>>
template<typename Other_Scalar_T , const index_t Other_LO, const index_t Other_HI, typename
Other_Tune_P >
friend class framed_multi [friend]
```

Definition at line 143 of file framed\_multi.h.

## 6.13.5.3 matrix\_multi

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
template<typename Other_Scalar_T , const index_t Other_LO, const index_t Other_HI, typename Other_Tune_P >
friend class matrix_multi [friend]
```

Definition at line 141 of file framed\_multi.h.

## 6.13.5.4 operator &amp;

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator& (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> const multivector_t [friend]
```

## 6.13.5.5 operator%

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator% (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> const multivector_t [friend]
```

## 6.13.5.6 operator\*

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator* (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> const multivector_t [friend]
```

## 6.13.5.7 operator/

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator/ (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> const multivector_t [friend]
```

**6.13.5.8 operator<< [1/2]**

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator<< (
    std::ostream & os,
    const multivector_t & val ) -> std::ostream & [friend]
```

**6.13.5.9 operator<< [2/2]**

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator<< (
    std::ostream & os,
    const term_t & term ) -> std::ostream & [friend]
```

**6.13.5.10 operator>>**

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator>> (
    std::istream & s,
    multivector_t & val ) -> std::istream & [friend]
```

**6.13.5.11 operator^**

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator^ (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> const multivector_t [friend]
```

**6.13.5.12 operator" |**

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator| (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> const multivector_t [friend]
```

## 6.13.5.13 star

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto star (
    const multivector_t & lhs,
    const multivector_t & rhs ) -> Scalar_T [friend]
```

The documentation for this class was generated from the following files:

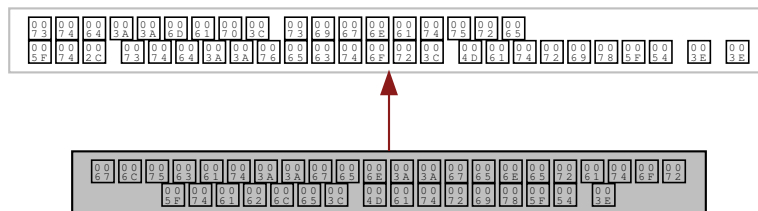
- glucat/framed\_multi.h
- glucat/framed\_multi\_imp.h

## 6.14 glucat::gen::generator\_table&lt; Matrix\_T &gt; Class Template Reference

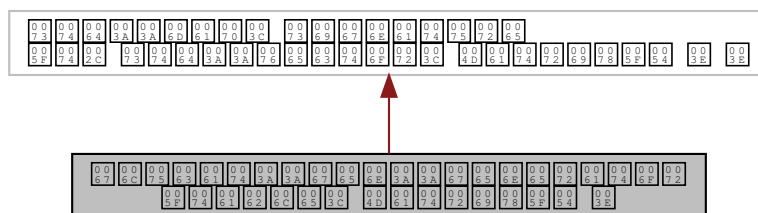
Table of generators for specific signatures.

```
#include <generation.h>
```

Inheritance diagram for glucat::gen::generator\_table< Matrix\_T >:



Collaboration diagram for glucat::gen::generator\_table< Matrix\_T >:



## Public Member Functions

- auto [operator\(\)](#) (const [index\\_t](#) p, const [index\\_t](#) q) -> const Matrix\_T \*  
*Pointer to generators for a specific signature.*
- [generator\\_table](#) (const [generator\\_table](#) &)=delete
- auto [operator=](#) (const [generator\\_table](#) &) -> [generator\\_table](#) &=delete

## Static Public Member Functions

- static auto [generator](#) () -> [generator\\_table](#)< Matrix\_T > &  
*Single instance of generator table.*

## Private Member Functions

- auto [gen\\_vector](#) (const [index\\_t](#) p, const [index\\_t](#) q) -> const std::vector< Matrix\_T > &  
*Construct a vector of generators for a specific signature.*
- void [gen\\_from\\_pm1\\_qm1](#) (const std::vector< Matrix\_T > &old, const [signature\\_t](#) sig)  
*Construct generators for p,q given generators for p-1,q-1.*
- void [gen\\_from\\_pm4\\_qp4](#) (const std::vector< Matrix\_T > &old, const [signature\\_t](#) sig)  
*Construct generators for p,q given generators for p-4,q+4.*
- void [gen\\_from\\_pp4\\_qm4](#) (const std::vector< Matrix\_T > &old, const [signature\\_t](#) sig)  
*Construct generators for p,q given generators for p+4,q-4.*
- void [gen\\_from\\_qp1\\_pm1](#) (const std::vector< Matrix\_T > &old, const [signature\\_t](#) sig)  
*Construct generators for p,q given generators for q+1,p-1.*
- [generator\\_table](#) ()=default
- [~generator\\_table](#) ()=default

## Friends

- class [friend\\_for\\_private\\_destructor](#)

### 6.14.1 Detailed Description

```
template<class Matrix_T>
class glucat::gen::generator_table< Matrix_T >
```

Table of generators for specific signatures.

Definition at line 52 of file generation.h.

### 6.14.2 Constructor & Destructor Documentation



## 6.14.2.1 generator\_table() [1/2]

```
template<class Matrix_T>
glucat::gen::generator_table< Matrix_T >::generator_table ( ) [private], [default]
```

## 6.14.2.2 ~generator\_table()

```
template<class Matrix_T>
glucat::gen::generator_table< Matrix_T >::~~generator_table ( ) [private], [default]
```

## 6.14.2.3 generator\_table() [2/2]

```
template<class Matrix_T>
glucat::gen::generator_table< Matrix_T >::generator_table (
    const generator_table< Matrix_T > & ) [delete]
```

## 6.14.3 Member Function Documentation

## 6.14.3.1 gen\_from\_pm1\_qm1()

```
template<class Matrix_T >
void glucat::gen::generator_table< Matrix_T >::gen_from_pm1_qm1 (
    const std::vector< Matrix_T > & old,
    const signature_t sig ) [private]
```

Construct generators for p,q given generators for p-1,q-1.

Definition at line 128 of file generation\_imp.h.

References glucat::matrix::mono\_kron().

## 6.14.3.2 gen\_from\_pm4\_qp4()

```
template<class Matrix_T >
void glucat::gen::generator_table< Matrix_T >::gen_from_pm4_qp4 (
    const std::vector< Matrix_T > & old,
    const signature_t sig ) [private]
```

Construct generators for p,q given generators for p-4,q+4.

Definition at line 166 of file generation\_imp.h.

References glucat::matrix::mono\_prod().

#### 6.14.3.3 gen\_from\_pp4\_qm4()

```
template<class Matrix_T >
void glucat::gen::generator_table< Matrix_T >::gen_from_pp4_qm4 (
    const std::vector< Matrix_T > & old,
    const signature_t sig ) [private]
```

Construct generators for p,q given generators for p+4,q-4.

Definition at line 199 of file generation\_imp.h.

References glucat::matrix::mono\_prod().

#### 6.14.3.4 gen\_from\_qp1\_pm1()

```
template<class Matrix_T >
void glucat::gen::generator_table< Matrix_T >::gen_from_qp1_pm1 (
    const std::vector< Matrix_T > & old,
    const signature_t sig ) [private]
```

Construct generators for p,q given generators for q+1,p-1.

Definition at line 232 of file generation\_imp.h.

References glucat::matrix::mono\_prod().

#### 6.14.3.5 gen\_vector()

```
template<class Matrix_T >
auto glucat::gen::generator_table< Matrix_T >::gen_vector (
    const index_t p,
    const index_t q ) -> const std::vector<Matrix_T>& [private]
```

Construct a vector of generators for a specific signature.

Definition at line 80 of file generation\_imp.h.

References glucat::pos\_mod().

#### 6.14.3.6 generator()

```
template<class Matrix_T >
auto glucat::gen::generator_table< Matrix_T >::generator ( ) -> generator_table<Matrix_T>&
[static]
```

Single instance of generator table.

Definition at line 50 of file generation\_imp.h.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::basis\_element().

#### 6.14.3.7 operator>()

```
template<class Matrix_T >
auto glucat::gen::generator_table< Matrix_T >::operator() (
    const index_t p,
    const index_t q ) -> const Matrix_T* [inline]
```

Pointer to generators for a specific signature.

Definition at line 59 of file generation\_imp.h.

References glucat::gen::offset\_to\_super, and glucat::pos\_mod().

#### 6.14.3.8 operator=()

```
template<class Matrix_T>
auto glucat::gen::generator_table< Matrix_T >::operator= (
    const generator_table< Matrix_T > & ) -> generator_table &=delete [delete]
```

### 6.14.4 Friends And Related Function Documentation

#### 6.14.4.1 friend\_for\_private\_destructor

```
template<class Matrix_T>
friend class friend_for_private_destructor [friend]
```

Friend declaration to avoid compiler warning: "... only defines a private destructor and has no friends" Ref: Carlos O'Ryan, ACE <http://doc.ece.uci.edu>

Definition at line 75 of file generation.h.

The documentation for this class was generated from the following files:

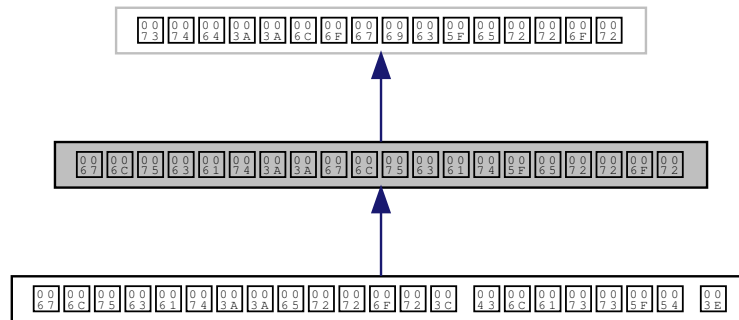
- glucat/[generation.h](#)
- glucat/[generation\\_imp.h](#)

## 6.15 glucat::glucat\_error Class Reference

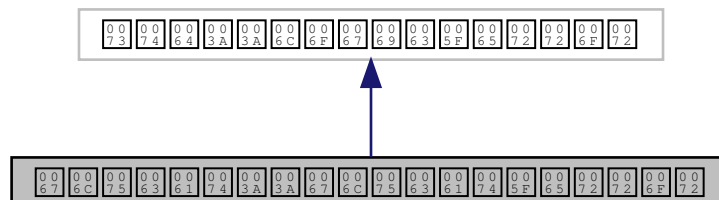
Abstract exception class.

```
#include <errors.h>
```

Inheritance diagram for glucat::glucat\_error:



Collaboration diagram for glucat::glucat\_error:



### Public Member Functions

- [glucat\\_error](#) (const std::string &context, const std::string &msg)
- [~glucat\\_error](#) () noexcept override=default
- virtual auto [heading](#) () const noexcept -> const std::string=0
- virtual auto [classname](#) () const noexcept -> const std::string=0
- virtual void [print\\_error\\_msg](#) () const=0

### Public Attributes

- std::string [name](#)

### 6.15.1 Detailed Description

Abstract exception class.

Definition at line 41 of file errors.h.

### 6.15.2 Constructor & Destructor Documentation

#### 6.15.2.1 glucat\_error()

```
glucat::glucat_error::glucat_error (
    const std::string & context,
    const std::string & msg ) [inline]
```

Definition at line 44 of file errors.h.

#### 6.15.2.2 ~glucat\_error()

```
glucat::glucat_error::~~glucat_error ( ) [override], [default], [noexcept]
```

### 6.15.3 Member Function Documentation

#### 6.15.3.1 classname()

```
virtual auto glucat::glucat_error::classname ( ) const -> const std::string [pure virtual],
[noexcept]
```

Implemented in [glucat::error< Class\\_T >](#).

#### 6.15.3.2 heading()

```
virtual auto glucat::glucat_error::heading ( ) const -> const std::string [pure virtual],
[noexcept]
```

Implemented in [glucat::error< Class\\_T >](#).

### 6.15.3.3 print\_error\_msg()

```
virtual void glucat::glucat_error::print_error_msg ( ) const [pure virtual]
```

Implemented in [glucat::error< Class\\_T >](#).

## 6.15.4 Member Data Documentation

### 6.15.4.1 name

```
std::string glucat::glucat_error::name
```

Definition at line 51 of file errors.h.

The documentation for this class was generated from the following file:

- [glucat/errors.h](#)

## 6.16 glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::hash\_size\_t Class Reference

### Public Member Functions

- [hash\\_size\\_t](#) (size\_t hash\_size)
- auto [operator\(\)](#) () const -> size\_t

### Private Attributes

- size\_t [n](#)

### 6.16.1 Detailed Description

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P =  
tuning<>>  
class glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::hash_size_t
```

Definition at line 152 of file framed\_multi.h.

### 6.16.2 Constructor & Destructor Documentation

### 6.16.2.1 hash\_size\_t()

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::hash_size_t::hash_size_t (
    size_t hash_size ) [inline]
```

Definition at line 155 of file framed\_multi.h.

## 6.16.3 Member Function Documentation

### 6.16.3.1 operator>()()

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::hash_size_t::operator() ( ) const ->
size_t [inline]
```

Definition at line 158 of file framed\_multi.h.

References glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::hash\_size\_t::n.

## 6.16.4 Member Data Documentation

### 6.16.4.1 n

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
size_t glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::hash_size_t::n [private]
```

Definition at line 161 of file framed\_multi.h.

Referenced by glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::hash\_size\_t::operator>()().

The documentation for this class was generated from the following file:

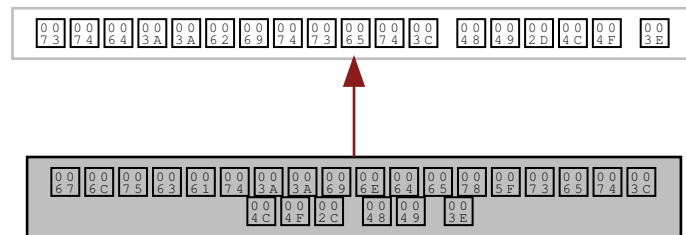
- [glucat/framed\\_multi.h](#)

## 6.17 glucat::index\_set< LO, HI > Class Template Reference

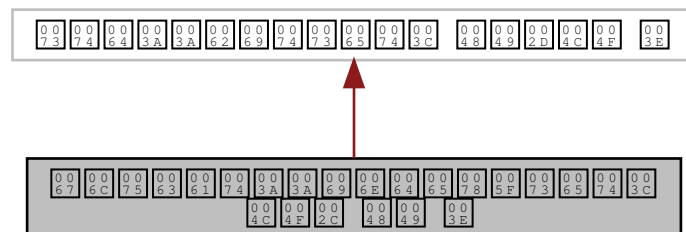
Index set class based on `std::bitset<>` in Gnu standard C++ library.

```
#include <index_set.h>
```

Inheritance diagram for `glucat::index_set< LO, HI >`:



Collaboration diagram for `glucat::index_set< LO, HI >`:



### Classes

- class [reference](#)  
*Index set member reference.*

### Public Types

- using [index\\_set\\_t](#) = [index\\_set](#)
- using [index\\_pair\\_t](#) = `std::pair< index\_t, index\_t >`



## Public Member Functions

- `index_set()` = default  
*Default constructor creates an empty set.*
- `index_set(const bitset_t bst)`  
*Constructor from bitset\_t.*
- `index_set(const index_t idx)`  
*Constructor from index.*
- `index_set(const set_value_t folded_val, const index_set_t frm, const bool prechecked=false)`  
*Constructor from set value of an index set folded within the given frame.*
- `index_set(const index_pair_t &range, const bool prechecked=false)`  
*Constructor from range of indices from range.first to range.second.*
- `index_set(const std::string &str)`  
*Constructor from string.*
- `auto operator== (const index_set_t rhs) const -> bool`  
*Equality.*
- `auto operator!= (const index_set_t rhs) const -> bool`  
*Inequality.*
- `auto operator~ () const -> index_set_t`  
*Set complement: not.*
- `auto operator^= (const index_set_t rhs) -> index_set_t &`  
*Symmetric set difference: exclusive or.*
- `auto operator &= (const index_set_t rhs) -> index_set_t &`  
*Set intersection: and.*
- `auto operator|= (const index_set_t rhs) -> index_set_t &`  
*Set union: or.*
- `auto operator[] (const index_t idx) const -> bool`  
*Subscripting: Test idx for membership: test value of bit idx.*
- `auto test (const index_t idx) const -> bool`  
*Test idx for membership: test value of bit idx.*
- `auto set () -> index_set_t &`  
*Include all indices except 0: set all bits except 0.*
- `auto set (const index_t idx) -> index_set_t &`  
*Include idx: Set bit at idx if idx != 0.*
- `auto set (const index_t idx, const int val) -> index_set_t &`  
*Set membership of idx to val if idx != 0: Set bit at idx to val if idx != 0.*
- `auto reset () -> index_set_t &`  
*Make set empty: Set all bits to 0.*
- `auto reset (const index_t idx) -> index_set_t &`  
*Exclude idx: Set bit at idx to 0.*
- `auto flip () -> index_set_t &`  
*Set complement, except 0: flip all bits, except 0.*
- `auto flip (const index_t idx) -> index_set_t &`  
*Complement membership of idx if idx != 0: flip bit at idx if idx != 0.*
- `auto count () const -> index_t`  
*Cardinality: Number of indices included in set.*
- `auto count_neg () const -> index_t`  
*Number of negative indices included in set.*
- `auto count_pos () const -> index_t`  
*Number of positive indices included in set.*
- `auto min () const -> index_t`

*Minimum member.*

- auto `max` () const -> `index_t`

*Maximum member.*

- auto `operator<` (const `index_set_t` rhs) const -> bool

*Less than operator used for comparisons, map, etc.*

- auto `is_contiguous` () const -> bool

*Determine if the index set is contiguous, ie. has no gaps.*

- auto `fold` () const -> const `index_set_t`

*Fold this index set within itself as a frame.*

- auto `fold` (const `index_set_t` frm, const bool prechecked=false) const -> const `index_set_t`

*Fold this index set within the given frame.*

- auto `unfold` (const `index_set_t` frm, const bool prechecked=false) const -> const `index_set_t`

*Unfold this index set within the given frame.*

- auto `value_of_fold` (const `index_set_t` frm) const -> `set_value_t`

*The set value of the fold of this index set within the given frame.*

- auto `sign_of_mult` (const `index_set_t` ist) const -> int

*Sign of geometric product of two Clifford basis elements.*

- auto `sign_of_square` () const -> int

*Sign of geometric square of a Clifford basis element.*

- auto `hash_fn` () const -> `size_t`

*Hash function.*

- auto `operator[]` (`index_t` idx) -> `reference`

*Subscripting: Element access.*

## Static Public Member Functions

- static auto `classname` () -> const std::string

## Static Public Attributes

- static const `index_t v_lo` = LO
- static const `index_t v_hi` = HI

## Private Types

- using `bitset_t` = std::bitset< HI - LO >
- using `error_t` = `error`< `index_set` >

## Private Member Functions

- `BOOST_STATIC_ASSERT` ((LO<=0) &&(0<=HI) &&(LO< HI) &&(-LO< \_GLUCAT\_BITS\_PER\_ULONG) &&(HI< \_GLUCAT\_BITS\_PER\_ULONG) &&(HI-LO<=\_GLUCAT\_BITS\_PER\_ULONG))
- auto `lex_less_than` (const `index_set_t` rhs) const -> bool

*Lexicographic ordering of two sets: \*this < rhs.*

## Friends

- class [reference](#)
- auto [operator^](#) (const [index\\_set\\_t](#) &lhs, const [index\\_set\\_t](#) &rhs) -> const [index\\_set\\_t](#)
- auto [operator &](#) (const [index\\_set\\_t](#) &lhs, const [index\\_set\\_t](#) &rhs) -> const [index\\_set\\_t](#)
- auto [operator|](#) (const [index\\_set\\_t](#) &lhs, const [index\\_set\\_t](#) &rhs) -> const [index\\_set\\_t](#)
- auto [compare](#) (const [index\\_set\\_t](#) &lhs, const [index\\_set\\_t](#) &rhs) -> int

## 6.17.1 Detailed Description

```
template<const index_t LO, const index_t HI>
class glucat::index_set< LO, HI >
```

Index set class based on std::bitset<> in Gnu standard C++ library.

Definition at line 45 of file index\_set.h.

## 6.17.2 Member Typedef Documentation

### 6.17.2.1 bitset\_t

```
template<const index_t LO, const index_t HI>
using glucat::index_set< LO, HI >::bitset_t = std::bitset<HI - LO> [private]
```

Definition at line 81 of file index\_set.h.

### 6.17.2.2 error\_t

```
template<const index_t LO, const index_t HI>
using glucat::index_set< LO, HI >::error_t = error<index_set> [private]
```

Definition at line 82 of file index\_set.h.

### 6.17.2.3 index\_pair\_t

```
template<const index_t LO, const index_t HI>
using glucat::index_set< LO, HI >::index_pair_t = std::pair<index_t, index_t>
```

Definition at line 85 of file index\_set.h.

#### 6.17.2.4 index\_set\_t

```
template<const index_t LO, const index_t HI>
using glucat::index_set< LO, HI >::index_set_t = index_set
```

Definition at line 84 of file index\_set.h.

### 6.17.3 Constructor & Destructor Documentation

#### 6.17.3.1 index\_set() [1/6]

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::index_set ( ) [default]
```

Default constructor creates an empty set.

#### 6.17.3.2 index\_set() [2/6]

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::index_set (
    const bitset_t bst )
```

Constructor from bitset\_t.

Definition at line 62 of file index\_set\_imp.h.

#### 6.17.3.3 index\_set() [3/6]

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::index_set (
    const index_t idx )
```

Constructor from index.

Constructor from index value.

Definition at line 56 of file index\_set\_imp.h.

**6.17.3.4 index\_set()** [4/6]

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::index_set (
    const set_value_t folded_val,
    const index_set_t frm,
    const bool prechecked = false )
```

Constructor from set value of an index set folded within the given frame.

Definition at line 69 of file index\_set\_imp.h.

References glucat::index\_set< LO, HI >::count(), glucat::index\_set< LO, HI >::fold(), glucat::index\_set< LO, HI >::min(), and glucat::index\_set< LO, HI >::unfold().

**6.17.3.5 index\_set()** [5/6]

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::index_set (
    const index_pair_t & range,
    const bool prechecked = false )
```

Constructor from range of indices from range.first to range.second.

Definition at line 83 of file index\_set\_imp.h.

**6.17.3.6 index\_set()** [6/6]

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::index_set (
    const std::string & str )
```

Constructor from string.

Definition at line 103 of file index\_set\_imp.h.

**6.17.4 Member Function Documentation****6.17.4.1 BOOST\_STATIC\_ASSERT()**

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::BOOST_STATIC_ASSERT (
    (LO<=0) && (0<=HI) && (LO< HI) && (-LO< _GLUCAT_BITS_PER_ULON→
TS_PER_ULONG) && (HI-LO<=_GLUCAT_BITS_PER_ULONG) ) [private]
```

#### 6.17.4.2 classname()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::classname ( ) -> const std::string [inline], [static]
```

Definition at line 50 of file index\_set\_imp.h.

#### 6.17.4.3 count()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::count ( ) const -> index_t [inline]
```

Cardinality: Number of indices included in set.

Definition at line 345 of file index\_set\_imp.h.

Referenced by glucat::index\_set< LO, HI >::count\_neg(), glucat::index\_set< LO, HI >::count\_pos(), glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi(), glucat::index\_set< LO, HI >::index\_set(), and glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().

#### 6.17.4.4 count\_neg()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::count_neg ( ) const -> index_t [inline]
```

Number of negative indices included in set.

Definition at line 365 of file index\_set\_imp.h.

References glucat::index\_set< LO, HI >::count().

#### 6.17.4.5 count\_pos()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::count_pos ( ) const -> index_t [inline]
```

Number of positive indices included in set.

Definition at line 377 of file index\_set\_imp.h.

References glucat::index\_set< LO, HI >::count().

**6.17.4.6 flip()** [1/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::flip ( ) -> index_set_t& [inline]
```

Set complement, except 0: flip all bits, except 0.

Definition at line 320 of file index\_set\_imp.h.

**6.17.4.7 flip()** [2/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::flip (
    const index_t idx ) -> index_set_t& [inline]
```

Complement membership of idx if idx != 0: flip bit at idx if idx != 0.

Definition at line 331 of file index\_set\_imp.h.

**6.17.4.8 fold()** [1/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::fold ( ) const -> const index_set_t [inline]
```

Fold this index set within itself as a frame.

Definition at line 748 of file index\_set\_imp.h.

Referenced by glucat::index\_set< LO, HI >::index\_set().

**6.17.4.9 fold()** [2/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::fold (
    const index_set_t frm,
    const bool prechecked = false ) const -> const index_set_t
```

Fold this index set within the given frame.

Definition at line 756 of file index\_set\_imp.h.

**6.17.4.10 hash\_fn()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::hash_fn ( ) const -> size_t [inline]
```

Hash function.

Definition at line 951 of file index\_set\_imp.h.

**6.17.4.11 is\_contiguous()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::is_contiguous ( ) const -> bool [inline]
```

Determine if the index set is contiguous, ie. has no gaps.

Determine if the index set is contiguous, ie. has no gaps when 0 is included.

Definition at line 733 of file index\_set\_imp.h.

**6.17.4.12 lex\_less\_than()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::lex_less_than (
    const index_set_t rhs ) const -> bool [inline], [private]
```

Lexicographic ordering of two sets: \*this < rhs.

Definition at line 589 of file index\_set\_imp.h.

References PyClical::rhs.

**6.17.4.13 max()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::max ( ) const -> index_t
```

Maximum member.

Maximum member, or 0 if none.

Definition at line 551 of file index\_set\_imp.h.

Referenced by PyClical.index\_set::\_\_iter\_\_(), glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi(), and glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().



## 6.17.4.14 min()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::min ( ) const -> index_t
```

Minimum member.

Minimum member, or 0 if none.

Definition at line 462 of file index\_set\_imp.h.

Referenced by PyClical.index\_set::\_\_iter\_\_(), glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi(), glucat::index\_set< LO, HI >::index\_set(), glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi(), and glucat::index\_set< LO, HI >::unfold().

## 6.17.4.15 operator &amp;=()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator&= (
    const index_set_t rhs ) -> index_set_t &
```

Set intersection: and.

## 6.17.4.16 operator !=()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator!= (
    const index_set_t rhs ) const -> bool [inline]
```

Inequality.

Definition at line 131 of file index\_set\_imp.h.

References PyClical::rhs.

## 6.17.4.17 operator &lt;()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator< (
    const index_set_t rhs ) const -> bool [inline]
```

Less than operator used for comparisons, map, etc.

Definition at line 598 of file index\_set\_imp.h.

References PyClical::rhs.

**6.17.4.18 operator==( )**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator==(
    const index_set_t rhs ) const -> bool [inline]
```

Equality.

Definition at line 120 of file index\_set\_imp.h.

References PyClical::rhs.

**6.17.4.19 operator[]() [1/2]**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator[] (
    const index_t idx ) const -> bool [inline]
```

Subscripting: Test idx for membership: test value of bit idx.

Definition at line 233 of file index\_set\_imp.h.

**6.17.4.20 operator[]() [2/2]**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator[] (
    index_t idx ) -> reference [inline]
```

Subscripting: Element access.

Definition at line 225 of file index\_set\_imp.h.

**6.17.4.21 operator^=( )**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator^= (
    const index_set_t rhs ) -> index_set_t& [inline]
```

Symmetric set difference: exclusive or.

Definition at line 150 of file index\_set\_imp.h.

References PyClical::rhs.

#### 6.17.4.22 operator" |=()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator|= (
    const index_set_t rhs ) -> index_set_t& [inline]
```

Set union: or.

Definition at line 200 of file index\_set\_imp.h.

References PyClical::rhs.

#### 6.17.4.23 operator~()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::operator~ ( ) const -> index_set_t [inline]
```

Set complement: not.

Definition at line 142 of file index\_set\_imp.h.

#### 6.17.4.24 reset() [1/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reset ( ) -> index_set_t& [inline]
```

Make set empty: Set all bits to 0.

Definition at line 295 of file index\_set\_imp.h.

#### 6.17.4.25 reset() [2/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reset (
    const index_t idx ) -> index_set_t& [inline]
```

Exclude idx: Set bit at idx to 0.

Definition at line 306 of file index\_set\_imp.h.

**6.17.4.26 set()** [1/3]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::set ( ) -> index_set_t& [inline]
```

Include all indices except 0: set all bits except 0.

Definition at line 256 of file index\_set\_imp.h.

**6.17.4.27 set()** [2/3]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::set (
    const index_t idx ) -> index_set_t& [inline]
```

Include idx: Set bit at idx if idx != 0.

Definition at line 267 of file index\_set\_imp.h.

**6.17.4.28 set()** [3/3]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::set (
    const index_t idx,
    const int val ) -> index_set_t& [inline]
```

Set membership of idx to val if idx != 0: Set bit at idx to val if idx != 0.

Definition at line 281 of file index\_set\_imp.h.

**6.17.4.29 sign\_of\_mult()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::sign_of_mult (
    const index_set_t ist ) const -> int
```

Sign of geometric product of two Clifford basis elements.

Definition at line 881 of file index\_set\_imp.h.

References glucat::inverse\_gray(), glucat::inverse\_reversed\_gray(), and PyClical::rhs.

**6.17.4.30 sign\_of\_square()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::sign_of_square ( ) const -> int [inline]
```

Sign of geometric square of a Clifford basis element.

Definition at line 931 of file index\_set\_imp.h.

**6.17.4.31 test()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::test (
    const index_t idx ) const -> bool [inline]
```

Test idx for membership: test value of bit idx.

Definition at line 241 of file index\_set\_imp.h.

**6.17.4.32 unfold()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::unfold (
    const index_set_t frm,
    const bool prechecked = false ) const -> const index_set_t
```

Unfold this index set within the given frame.

Definition at line 795 of file index\_set\_imp.h.

References glucat::index\_set< LO, HI >::min().

Referenced by glucat::index\_set< LO, HI >::index\_set().

**6.17.4.33 value\_of\_fold()**

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::value_of_fold (
    const index_set_t frm ) const -> set_value_t [inline]
```

The set value of the fold of this index set within the given frame.

Definition at line 830 of file index\_set\_imp.h.

## 6.17.5 Friends And Related Function Documentation

### 6.17.5.1 compare

```
template<const index_t LO, const index_t HI>
auto compare (
    const index_set_t & lhs,
    const index_set_t & rhs ) -> int [friend]
```

### 6.17.5.2 operator &

```
template<const index_t LO, const index_t HI>
auto operator& (
    const index_set_t & lhs,
    const index_set_t & rhs ) -> const index_set_t [friend]
```

### 6.17.5.3 operator^

```
template<const index_t LO, const index_t HI>
auto operator^ (
    const index_set_t & lhs,
    const index_set_t & rhs ) -> const index_set_t [friend]
```

### 6.17.5.4 operator" |

```
template<const index_t LO, const index_t HI>
auto operator| (
    const index_set_t & lhs,
    const index_set_t & rhs ) -> const index_set_t [friend]
```

### 6.17.5.5 reference

```
template<const index_t LO, const index_t HI>
friend class reference [friend]
```

Definition at line 173 of file index\_set.h.

### 6.17.6 Member Data Documentation

#### 6.17.6.1 v\_hi

```
template<const index_t LO, const index_t HI>
const index_t glucat::index_set< LO, HI >::v_hi = HI [static]
```

Definition at line 88 of file index\_set.h.

#### 6.17.6.2 v\_lo

```
template<const index_t LO, const index_t HI>
const index_t glucat::index_set< LO, HI >::v_lo = LO [static]
```

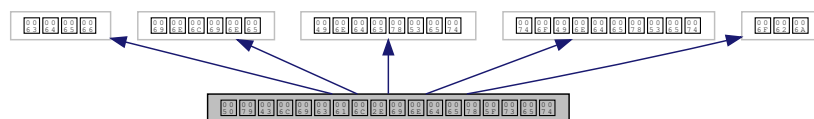
Definition at line 87 of file index\_set.h.

The documentation for this class was generated from the following files:

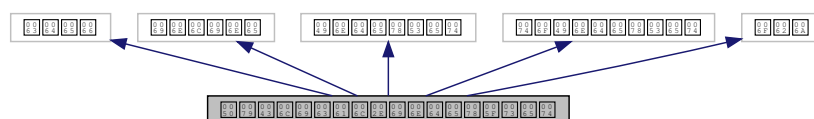
- [glucat/index\\_set.h](#)
- [glucat/index\\_set\\_imp.h](#)

## 6.18 PyClical.index\_set Class Reference

Inheritance diagram for PyClical.index\_set:



Collaboration diagram for PyClical.index\_set:



## Public Member Functions

- `def __cinit__(self, other=0)`
- `def __dealloc__(self)`
- `def __richcmp__(lhs, rhs, int, op)`
- `def __setitem__(self, idx, val)`
- `def __getitem__(self, idx)`
- `def __contains__(self, idx)`
- `def __iter__(self)`
- `def __invert__(self)`
- `def __xor__(lhs, rhs)`
- `def __ixor__(self, rhs)`
- `def __and__(lhs, rhs)`
- `def __iand__(self, rhs)`
- `def __or__(lhs, rhs)`
- `def __ior__(self, rhs)`
- `def count(self)`
- `def count_neg(self)`
- `def count_pos(self)`
- `def min(self)`
- `def max(self)`
- `def hash_fn(self)`
- `def sign_of_mult(self, rhs)`
- `def sign_of_square(self)`
- `def __repr__(self)`
- `def __str__(self)`

## Public Attributes

- `instance`

### 6.18.1 Detailed Description

Return the C++ `IndexSet` instance wrapped by `index_set(obj)`.

Python class `index_set` wraps C++ class `IndexSet`.

Definition at line 40 of file `PyClical.pyx`.

### 6.18.2 Member Function Documentation



### 6.18.2.1 `__and__()`

```
def PyClical.index_set.__and__ (
    lhs,
    rhs )
```

Set intersection: `and`.

```
>>> print(index_set({1}) & index_set({2}))
{}
>>> print(index_set({1,2}) & index_set({2}))
{2}
```

Definition at line 271 of file `PyClical.pyx`.

### 6.18.2.2 `__cinit__()`

```
def PyClical.index_set.__cinit__ (
    self,
    other = 0 )
```

Construct an object of type `index_set`.

```
>>> print(index_set(1))
{1}
>>> print(index_set({1,2}))
{1,2}
>>> print(index_set(index_set({1,2})))
{1,2}
>>> print(index_set({1,2}))
{1,2}
>>> print(index_set({1,2,1}))
{1,2}
>>> print(index_set("{1,2,1}"))
{1,2}
>>> print(index_set(""))
{}
```

Definition at line 74 of file `PyClical.pyx`.

### 6.18.2.3 `__contains__()`

```
def PyClical.index_set.__contains__ (
    self,
    idx )
```

Check that an `index_set` object contains the index `idx`: `idx in self`.

```
>>> 1 in index_set({1})
True
>>> 2 in index_set({1})
False
>>> -1 in index_set({2})
False
>>> 1 in index_set({2})
False
>>> 2 in index_set({2})
True
>>> 33 in index_set({2})
False
```

Definition at line 210 of file `PyClicl.pyx`.

References `PyClicl.index_set.instance`.

#### 6.18.2.4 `__dealloc__()`

```
def PyClicl.index_set.__dealloc__ (
    self )
```

Clean up by deallocating the instance of C++ class `IndexSet`.

Definition at line 116 of file `PyClicl.pyx`.

References `PyClicl.index_set.instance`.

#### 6.18.2.5 `__getitem__()`

```
def PyClicl.index_set.__getitem__ (
    self,
    idx )
```

Get the value of an `index_set` object at an index.

```
>>> index_set({1})[1]
True
>>> index_set({1})[2]
False
>>> index_set({2})[-1]
False
>>> index_set({2})[1]
False
>>> index_set({2})[2]
True
>>> index_set({2})[33]
False
```

Definition at line 191 of file `PyClicl.pyx`.

References `PyClicl.index_set.instance`.

#### 6.18.2.6 \_\_iand\_\_()

```
def PyClical.index_set.__iand__ (
    self,
    rhs )

Set intersection: and.

>>> x = index_set({1}); x &= index_set({2}); print(x)
{}
>>> x = index_set({1,2}); x &= index_set({2}); print(x)
{2}
```

Definition at line 282 of file PyClical.pyx.

#### 6.18.2.7 \_\_invert\_\_()

```
def PyClical.index_set.__invert__ (
    self )

Set complement: not.

>>> print(~index_set({-16,-15,-14,-13,-12,-11,-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
{-32,-31,-30,-29,-28,-27,-26,-25,-24,-23,-22,-21,-20,-19,-18,-17,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,
```

Definition at line 240 of file PyClical.pyx.

References `PyClical.index_set.instance`.

#### 6.18.2.8 \_\_ior\_\_()

```
def PyClical.index_set.__ior__ (
    self,
    rhs )

Set union: or.

>>> x = index_set({1}); x |= index_set({2}); print(x)
{1,2}
>>> x = index_set({1,2}); x |= index_set({2}); print(x)
{1,2}
```

Definition at line 304 of file PyClical.pyx.

#### 6.18.2.9 `__iter__()`

```
def PyClical.index_set.__iter__ (
    self )
```

Iterate over the indices of an `index_set`.

```
>>> for i in index_set({-3,4,7}):print(i, end=", ")
-3,4,7,
```

Definition at line 229 of file `PyClical.pyx`.

References `glucat::index_set< LO, HI >.max()`, `PyClical.index_set.max()`, `glucat::index_set< LO, HI >.min()`, and `PyClical.index_set.min()`.

#### 6.18.2.10 `__ixor__()`

```
def PyClical.index_set.__ixor__ (
    self,
    rhs )
```

Symmetric set difference: exclusive or.

```
>>> x = index_set({1}); x ^= index_set({2}); print(x)
{1,2}
>>> x = index_set({1,2}); x ^= index_set({2}); print(x)
{1}
```

Definition at line 260 of file `PyClical.pyx`.

#### 6.18.2.11 `__or__()`

```
def PyClical.index_set.__or__ (
    lhs,
    rhs )
```

Set union: or.

```
>>> print(index_set({1}) | index_set({2}))
{1,2}
>>> print(index_set({1,2}) | index_set({2}))
{1,2}
```

Definition at line 293 of file `PyClical.pyx`.

#### 6.18.2.12 `__repr__()`

```
def PyClical.index_set.__repr__ (
    self )
```

The “official” string representation of self.

```
>>> index_set({1,2}).__repr__()
'index_set({1,2})'
>>> repr(index_set({1,2}))
'index_set({1,2})'
```

Definition at line 384 of file `PyClical.pyx`.

References `index_set_to_repr()`.

#### 6.18.2.13 `__richcmp__()`

```
def PyClical.index_set.__richcmp__ (
    lhs,
    rhs,
    int,
    op )
```

Compare two objects of class `index_set`.

```
>>> index_set(1) == index_set({1})
True
>>> index_set({1}) != index_set({1})
False
>>> index_set({1}) != index_set({2})
True
>>> index_set({1}) == index_set({2})
False
>>> index_set({1}) < index_set({2})
True
>>> index_set({1}) <= index_set({2})
True
>>> index_set({1}) > index_set({2})
False
>>> index_set({1}) >= index_set({2})
False
```

Definition at line 122 of file `PyClical.pyx`.

#### 6.18.2.14 `__setitem__()`

```
def PyClical.index_set.__setitem__ (
    self,
    idx,
    val )
```

Set the value of an `index_set` object at index `idx` to value `val`.

```
>>> s=index_set({1}); s[2] = True; print(s)
{1,2}
>>> s=index_set({1,2}); s[1] = False; print(s)
{2}
```

Definition at line 179 of file `PyClical.pyx`.

References `PyClical.index_set.instance`.

#### 6.18.2.15 `__str__()`

```
def PyClical.index_set.__str__ (
    self )
```

The “informal” string representation of self.

```
>>> index_set({1,2}).__str__()
' {1,2} '
>>> str(index_set({1,2}))
' {1,2} '
```

Definition at line 395 of file PyClical.pyx.

References `index_set_to_str()`.

#### 6.18.2.16 `__xor__()`

```
def PyClical.index_set.__xor__ (
    lhs,
    rhs )
```

Symmetric set difference: exclusive or.

```
>>> print(index_set({1}) ^ index_set({2}))
{1,2}
>>> print(index_set({1,2}) ^ index_set({2}))
{1}
```

Definition at line 249 of file PyClical.pyx.

#### 6.18.2.17 `count()`

```
def PyClical.index_set.count (
    self )
```

Cardinality: Number of indices included in set.

```
>>> index_set({-1,1,2}).count()
3
```

Definition at line 315 of file PyClical.pyx.

References `PyClical.index_set.instance`.

#### 6.18.2.18 count\_neg()

```
def PyClical.index_set.count_neg (
    self )
```

Number of negative indices included in set.

```
>>> index_set({-1,1,2}).count_neg()
1
```

Definition at line 324 of file PyClical.pyx.

References `PyClical.index_set.instance`.

#### 6.18.2.19 count\_pos()

```
def PyClical.index_set.count_pos (
    self )
```

Number of positive indices included in set.

```
>>> index_set({-1,1,2}).count_pos()
2
```

Definition at line 333 of file PyClical.pyx.

References `PyClical.index_set.instance`.

#### 6.18.2.20 hash\_fn()

```
def PyClical.index_set.hash_fn (
    self )
```

Hash function.

Definition at line 360 of file PyClical.pyx.

References `PyClical.index_set.instance`.

#### 6.18.2.21 max()

```
def PyClical.index_set.max (
    self )
```

Maximum member.

```
>>> index_set({-1,1,2}).max()
2
```

Definition at line 351 of file PyClical.pyx.

References PyClical.index\_set.instance.

Referenced by PyClical.index\_set.\_\_iter\_\_().

#### 6.18.2.22 min()

```
def PyClical.index_set.min (
    self )
```

Minimum member.

```
>>> index_set({-1,1,2}).min()
-1
```

Definition at line 342 of file PyClical.pyx.

References PyClical.index\_set.instance.

Referenced by PyClical.index\_set.\_\_iter\_\_().

#### 6.18.2.23 sign\_of\_mult()

```
def PyClical.index_set.sign_of_mult (
    self,
    rhs )
```

Sign of geometric product of two Clifford basis elements.

```
>>> s = index_set({1,2}); t=index_set({-1}); s.sign_of_mult(t)
1
```

Definition at line 366 of file PyClical.pyx.

References PyClical.index\_set.instance.



6.18.2.24 `sign_of_square()`

```
def PyClical.index_set.sign_of_square (
    self )
```

Sign of geometric square of a Clifford basis element.

```
>>> s = index_set({1,2}); s.sign_of_square()
-1
```

Definition at line 375 of file `PyClical.pyx`.

References `PyClical.index_set.instance`.

## 6.18.3 Member Data Documentation

6.18.3.1 `instance`

`PyClical.index_set.instance`

Definition at line 95 of file `PyClical.pyx`.

Referenced by `PyClical.clifford.__call__()`, `PyClical.index_set.__contains__()`, `PyClical.index_set.__dealloc__`↵  
`()`, `PyClical.clifford.__dealloc__()`, `PyClical.index_set.__getitem__()`, `PyClical.clifford.__getitem__()`, `PyClical`↵  
`index_set.__invert__()`, `PyClical.clifford.__neg__()`, `PyClical.index_set.__setitem__()`, `PyClical.clifford.conj()`, `Py`↵  
`Clical.index_set.count()`, `PyClical.index_set.count_neg()`, `PyClical.index_set.count_pos()`, `PyClical.clifford.even()`,  
`PyClical.clifford.frame()`, `PyClical.index_set.hash_fn()`, `PyClical.clifford.inv()`, `PyClical.clifford.involute()`, `PyClical`↵  
`clifford.isinf()`, `PyClical.clifford.isnan()`, `PyClical.index_set.max()`, `PyClical.clifford.max_abs()`, `PyClical.index_set`↵  
`min()`, `PyClical.clifford.norm()`, `PyClical.clifford.odd()`, `PyClical.clifford.outer_pow()`, `PyClical.clifford.pow()`, `Py`↵  
`Clical.clifford.pure()`, `PyClical.clifford.quad()`, `PyClical.clifford.reverse()`, `PyClical.clifford.scalar()`, `PyClical.index`↵  
`set.sign_of_mult()`, `PyClical.index_set.sign_of_square()`, `PyClical.clifford.truncated()`, and `PyClical.clifford.vector`↵  
`_part()`.

The documentation for this class was generated from the following file:

- `pyclical/PyClical.pyx`

6.19 `glucat::index_set_hash< LO, HI >` Class Template Reference

```
#include <framed_multi.h>
```

## Public Types

- using `index_set_t = index_set< LO, HI >`

## Public Member Functions

- auto [operator\(\)](#) ([index\\_set\\_t](#) val) const -> [size\\_t](#)

### 6.19.1 Detailed Description

```
template<const index_t LO, const index_t HI>  
class glucat::index_set_hash< LO, HI >
```

Definition at line 117 of file [framed\\_multi.h](#).

### 6.19.2 Member Typedef Documentation

#### 6.19.2.1 [index\\_set\\_t](#)

```
template<const index_t LO, const index_t HI>  
using glucat::index\_set\_hash< LO, HI >::index\_set\_t = index\_set<LO, HI>
```

Definition at line 120 of file [framed\\_multi.h](#).

### 6.19.3 Member Function Documentation

#### 6.19.3.1 [operator\(\)](#)()

```
template<const index_t LO, const index_t HI>  
auto glucat::index\_set\_hash< LO, HI >::operator\(\) (  
    index\_set\_t val ) const -> size\_t    [inline]
```

Definition at line 121 of file [framed\\_multi.h](#).

The documentation for this class was generated from the following file:

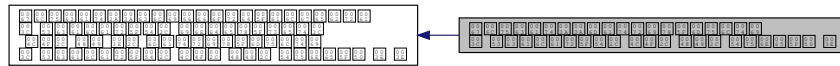
- [glucat/framed\\_multi.h](#)

## 6.20 glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P > Class Template Reference

A matrix\_multi<Scalar\_T,LO,HI,Tune\_P> is a matrix approximation to a multivector.

```
#include <framed_multi.h>
```

Inheritance diagram for glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >:



Collaboration diagram for glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >:



### Public Types

- using [multivector\\_t](#) = [matrix\\_multi](#)
- using [matrix\\_multi\\_t](#) = [multivector\\_t](#)
- using [scalar\\_t](#) = [Scalar\\_T](#)
- using [tune\\_p](#) = [Tune\\_P](#)
- using [index\\_set\\_t](#) = [index\\_set](#)< [LO](#), [HI](#) >
- using [term\\_t](#) = std::pair< const [index\\_set\\_t](#), [Scalar\\_T](#) >
- using [vector\\_t](#) = std::vector< [Scalar\\_T](#) >
- using [error\\_t](#) = [error](#)< [multivector\\_t](#) >
- using [framed\\_multi\\_t](#) = [framed\\_multi](#)< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) >

### Public Member Functions

- [~matrix\\_multi](#) () override=default  
*Destructor.*
- [matrix\\_multi](#) ()  
*Default constructor.*
- template<typename Other\_Scalar\_T >  
[matrix\\_multi](#) (const [matrix\\_multi](#)< Other\_Scalar\_T, [LO](#), [HI](#), [Tune\\_P](#) > &val)  
*Construct a multivector from a multivector with a different scalar type.*
- template<typename Other\_Scalar\_T >  
[matrix\\_multi](#) (const [matrix\\_multi](#)< Other\_Scalar\_T, [LO](#), [HI](#), [Tune\\_P](#) > &val, const [index\\_set\\_t](#) frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from a given multivector.*
- [matrix\\_multi](#) (const [multivector\\_t](#) &val, const [index\\_set\\_t](#) frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from a given multivector.*
- [matrix\\_multi](#) (const [index\\_set\\_t](#) ist, const [Scalar\\_T](#) &crd=[Scalar\\_T](#)(1))  
*Construct a multivector from an index set and a scalar coordinate.*

- `matrix_multi` (const `index_set_t` ist, const `Scalar_T` &crd, const `index_set_t` frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from an index set and a scalar coordinate.*
- `matrix_multi` (const `Scalar_T` &scr, const `index_set_t` frm=`index_set_t`())  
*Construct a multivector from a scalar (within a frame, if given)*
- `matrix_multi` (const int scr, const `index_set_t` frm=`index_set_t`())  
*Construct a multivector from an int (within a frame, if given)*
- `matrix_multi` (const `vector_t` &vec, const `index_set_t` frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from a given vector.*
- `matrix_multi` (const std::string &str)  
*Construct a multivector from a string: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `matrix_multi` (const std::string &str, const `index_set_t` frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from a string: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `matrix_multi` (const char \*str)  
*Construct a multivector from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".*
- `matrix_multi` (const char \*str, const `index_set_t` frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".*
- template<typename Other\_Scalar\_T >  
`matrix_multi` (const `framed_multi`< Other\_Scalar\_T, LO, HI, Tune\_P > &val)  
*Construct a multivector from a framed\_multi\_t.*
- template<typename Other\_Scalar\_T >  
`matrix_multi` (const `framed_multi`< Other\_Scalar\_T, LO, HI, Tune\_P > &val, const `index_set_t` frm, const bool prechecked=false)  
*Construct a multivector, within a given frame, from a framed\_multi\_t.*
- auto `fast_matrix_multi` (const `index_set_t` frm) const -> const `matrix_multi_t`  
*Use generalized FFT to construct a matrix\_multi\_t.*
- template<typename Other\_Scalar\_T >  
auto `fast_framed_multi` () const -> const `framed_multi`< Other\_Scalar\_T, LO, HI, Tune\_P >  
*Use inverse generalized FFT to construct a framed\_multi\_t.*
- `_GLUCAT_CLIFFORD_ALGEBRA_OPERATIONS` auto `operator=` (const `multivector_t` &rhs) -> `multivector_t` &  
*Assignment operator.*
- auto `operator+=` (const `term_t` &rhs) -> `multivector_t` &  
*Add a term, if non-zero.*

## Static Public Member Functions

- static auto `classname` () -> const std::string  
*Class name used in messages.*
- static auto `random` (const `index_set_t` frm, `Scalar_T` fill=`Scalar_T`(1)) -> const `matrix_multi_t`  
*Random multivector within a frame.*

## Private Types

- using `orientation_t` = `ublas::row_major`
- using `basis_matrix_t` = `ublas::compressed_matrix`< int, `orientation_t` >
- using `matrix_t` = `ublas::matrix`< `Scalar_T`, `orientation_t` >
- using `matrix_index_t` = `typename matrix_t::size_type`

## Private Member Functions

- template<typename Matrix\_T >  
matrix\_multi (const Matrix\_T &mtx, const index\_set\_t frm)  
*Construct a multivector within a given frame from a given matrix.*
- matrix\_multi (const matrix\_t &mtx, const index\_set\_t frm)  
*Construct a multivector within a given frame from a given matrix.*
- auto basis\_element (const index\_set< LO, HI > &ist) const -> const basis\_matrix\_t  
*Create a basis element matrix within the current frame.*

## Private Attributes

- index\_set\_t m\_frame  
*Index set representing the frame for the subalgebra which contains the multivector.*
- matrix\_t m\_matrix  
*Matrix value representing the multivector within the folded frame.*

## Friends

- template<typename Other\_Scalar\_T , const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
class framed\_multi
- template<typename Other\_Scalar\_T , const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
class matrix\_multi
- auto operator\* (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> const matrix\_multi\_t
- auto operator^ (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> const matrix\_multi\_t
- auto operator & (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> const matrix\_multi\_t
- auto operator% (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> const matrix\_multi\_t
- auto star (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> Scalar\_T
- auto operator/ (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> const matrix\_multi\_t
- auto operator| (const matrix\_multi\_t &lhs, const matrix\_multi\_t &rhs) -> const matrix\_multi\_t
- auto operator>> (std::istream &s, multivector\_t &val) -> std::istream &
- auto operator<< (std::ostream &os, const multivector\_t &val) -> std::ostream &
- template<typename Other\_Scalar\_T , const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
auto reframe (const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &lhs, const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &rhs, matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &lhs\_reframed, matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &rhs\_reframed) -> const index\_set< Other\_LO, Other\_HI >
- template<typename Other\_Scalar\_T , const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
auto matrix\_sqrt (const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &val, const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &i, const index\_t level) -> const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P >
- template<typename Other\_Scalar\_T , const index\_t Other\_LO, const index\_t Other\_HI, typename Other\_Tune\_P >  
auto matrix\_log (const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &val, const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P > &i, const index\_t level) -> const matrix\_multi< Other\_Scalar\_T, Other\_LO, Other\_HI, Other\_Tune\_P >

## Additional Inherited Members

### 6.20.1 Detailed Description

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P =
tuning<>>
class glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >
```

A matrix\_multi<Scalar\_T,LO,HI,Tune\_P> is a matrix approximation to a multivector.

Definition at line 59 of file framed\_multi.h.

## 6.20.2 Member Typedef Documentation

### 6.20.2.1 basis\_matrix\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::basis_matrix_t = ublas::compressed_↵
matrix<int, orientation_t> [private]
```

Definition at line 157 of file matrix\_multi.h.

### 6.20.2.2 error\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::error_t = error<multivector_t>
```

Definition at line 148 of file matrix\_multi.h.

### 6.20.2.3 framed\_multi\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::framed_multi_t = framed_multi<Scalar_↵
_T, LO, HI, Tune_P>
```

Definition at line 149 of file matrix\_multi.h.

### 6.20.2.4 index\_set\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::index_set_t = index_set<LO, HI>
```

Definition at line 145 of file matrix\_multi.h.

#### 6.20.2.5 matrix\_index\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_index_t = typename matrix_t< Scalar_T, LO, HI, Tune_P >::size_type [private]
```

Definition at line 159 of file matrix\_multi.h.

#### 6.20.2.6 matrix\_multi\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi_t = multivector_t< Scalar_T, LO, HI, Tune_P >
```

Definition at line 142 of file matrix\_multi.h.

#### 6.20.2.7 matrix\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_t = ublas::matrix<Scalar_T, orientation_t> [private]
```

Definition at line 158 of file matrix\_multi.h.

#### 6.20.2.8 multivector\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::multivector_t = matrix_multi< Scalar_T, LO, HI, Tune_P >
```

Definition at line 141 of file matrix\_multi.h.

#### 6.20.2.9 orientation\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::orientation_t = ublas::row_major [private]
```

Definition at line 156 of file matrix\_multi.h.

#### 6.20.2.10 scalar\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::scalar_t = Scalar_T
```

Definition at line 143 of file matrix\_multi.h.

#### 6.20.2.11 term\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::term_t = std::pair<const index_set_t, Scalar_T>
```

Definition at line 146 of file matrix\_multi.h.

#### 6.20.2.12 tune\_p

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::tune_p = Tune_P
```

Definition at line 144 of file matrix\_multi.h.

#### 6.20.2.13 vector\_t

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::vector_t = std::vector<Scalar_T>
```

Definition at line 147 of file matrix\_multi.h.

### 6.20.3 Constructor & Destructor Documentation

#### 6.20.3.1 ~matrix\_multi()

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::~matrix_multi ( ) [override], [default]
```

Destructor.



**6.20.3.2 matrix\_multi()** [1/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi ( )
```

Default constructor.

Definition at line 107 of file matrix\_multi\_imp.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().

**6.20.3.3 matrix\_multi()** [2/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const matrix_multi< Other_Scalar_T, LO, HI, Tune_P > & val )
```

Construct a multivector from a multivector with a different scalar type.

Definition at line 116 of file matrix\_multi\_imp.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix.

**6.20.3.4 matrix\_multi()** [3/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const matrix_multi< Other_Scalar_T, LO, HI, Tune_P > & val,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a given multivector.

Definition at line 135 of file matrix\_multi\_imp.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_frame, and glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix.

**6.20.3.5** `matrix_multi()` [4/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const multivector_t & val,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a given multivector.

Definition at line 160 of file `matrix_multi_imp.h`.

References `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_frame`, and `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_matrix`.

**6.20.3.6** `matrix_multi()` [5/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const index_set_t ist,
    const Scalar_T & crd = Scalar_T(1) )
```

Construct a multivector from an index set and a scalar coordinate.

Definition at line 172 of file `matrix_multi_imp.h`.

References `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_frame`, and `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_matrix`.

**6.20.3.7** `matrix_multi()` [6/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const index_set_t ist,
    const Scalar_T & crd,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from an index set and a scalar coordinate.

Definition at line 184 of file `matrix_multi_imp.h`.

References `PyClical::ist`, and `glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_matrix`.

**6.20.3.8 matrix\_multi()** [7/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const Scalar_T & scr,
    const index_set_t frm = index_set_t() )
```

Construct a multivector from a scalar (within a frame, if given)

Definition at line 198 of file matrix\_multi\_imp.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix.

**6.20.3.9 matrix\_multi()** [8/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const int scr,
    const index_set_t frm = index_set_t() )
```

Construct a multivector from an int (within a frame, if given)

Definition at line 210 of file matrix\_multi\_imp.h.

**6.20.3.10 matrix\_multi()** [9/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const vector_t & vec,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a given vector.

Definition at line 216 of file matrix\_multi\_imp.h.

References glucat::index\_set< LO, HI >::count(), glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix, glucat::index\_set< LO, HI >::max(), and glucat::index\_set< LO, HI >::min().

**6.20.3.11 matrix\_multi()** [10/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const std::string & str )
```

Construct a multivector from a string: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 241 of file matrix\_multi\_imp.h.

**6.20.3.12 matrix\_multi()** [11/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const std::string & str,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a string: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 247 of file matrix\_multi\_imp.h.

**6.20.3.13 matrix\_multi()** [12/17]

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const char * str ) [inline]
```

Construct a multivector from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 196 of file matrix\_multi.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().

**6.20.3.14 matrix\_multi()** [13/17]

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const char * str,
    const index_set_t frm,
    const bool prechecked = false ) [inline]
```

Construct a multivector, within a given frame, from a char\*: eg: "3+2{1,2}-6.1e-2{2,3}".

Definition at line 199 of file matrix\_multi.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().

**6.20.3.15 matrix\_multi()** [14/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const framed_multi< Other_Scalar_T, LO, HI, Tune_P > & val )
```

Construct a multivector from a framed\_multi\_t.

Definition at line 254 of file matrix\_multi\_imp.h.

References PyClical::e(), glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_frame, and glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix.

**6.20.3.16** matrix\_multi() [15/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const framed_multi< Other_Scalar_T, LO, HI, Tune_P > & val,
    const index_set_t frm,
    const bool prechecked = false )
```

Construct a multivector, within a given frame, from a framed\_multi\_t.

Definition at line 278 of file matrix\_multi\_imp.h.

References PyClical::e(), and glucat::clifford\_algebra< Scalar\_T, index\_set< LO, HI >, framed\_multi< Scalar\_T, LO, HI, Tune\_P > >::truncated().

**6.20.3.17** matrix\_multi() [16/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Matrix_T >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const Matrix_T & mtx,
    const index_set_t frm ) [private]
```

Construct a multivector within a given frame from a given matrix.

Definition at line 304 of file matrix\_multi\_imp.h.

References glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::m\_matrix.

**6.20.3.18** matrix\_multi() [17/17]

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::matrix_multi (
    const matrix_t & mtx,
    const index_set_t frm ) [private]
```

Construct a multivector within a given frame from a given matrix.

Definition at line 323 of file matrix\_multi\_imp.h.

**6.20.4** Member Function Documentation

#### 6.20.4.1 basis\_element()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::basis_element (
    const index_set< LO, HI > & ist ) const -> const basis_matrix_t [private]
```

Create a basis element matrix within the current frame.

Definition at line 1187 of file matrix\_multi\_imp.h.

References `PyClical::e()`, `glucat::gen::generator_table< Matrix_T >::generator()`, `PyClical::ist`, `glucat::matrix_multi::mono_prod()`, and `glucat::offset_level()`.

Referenced by `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi()`.

#### 6.20.4.2 classname()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::classname ( ) -> const std::string
[static]
```

Class name used in messages.

Definition at line 79 of file matrix\_multi\_imp.h.

#### 6.20.4.3 fast\_framed\_multi()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
template<typename Other_Scalar_T >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::fast_framed_multi ( ) const -> const
framed_multi<Other_Scalar_T,LO,HI,Tune_P>
```

Use inverse generalized FFT to construct a framed\_multi\_t.

Definition at line 1110 of file matrix\_multi\_imp.h.

References `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::centre_pm4_qp4()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::centre_pp4_qm4()`, `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::centre_qp1_pm1()`, `glucat::gen::offset_to_super`, `glucat::pos_mod()`, and `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::unfold()`.

#### 6.20.4.4 fast\_matrix\_multi()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::fast_matrix_multi (
    const index_set_t frm ) const -> const matrix_multi_t [inline]
```

Use generalized FFT to construct a matrix\_multi\_t.

Definition at line 1097 of file matrix\_multi\_imp.h.

## 6.20.4.5 operator+=()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::operator+= (
    const term_t & rhs ) -> multivector_t& [inline]
```

Add a term, if non-zero.

Geometric sum.

Geometric sum of multivector and scalar.

Definition at line 417 of file matrix\_multi\_imp.h.

## 6.20.4.6 operator=()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::operator= (
    const multivector_t & rhs ) -> multivector_t&
```

Assignment operator.

Definition at line 331 of file matrix\_multi\_imp.h.

References PyClical::rhs.

## 6.20.4.7 random()

```
template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >
auto glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::random (
    const index_set_t frm,
    Scalar_T fill = Scalar_T(1) ) -> const matrix_multi_t [static]
```

Random multivector within a frame.

Definition at line 927 of file matrix\_multi\_imp.h.

References PyClical::fill, and glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::random().

## 6.20.5 Friends And Related Function Documentation

### 6.20.5.1 framed\_multi

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
template<typename Other_Scalar_T , const index_t Other_LO, const index_t Other_HI, typename
Other_Tune_P >
friend class framed_multi [friend]
```

Definition at line 151 of file matrix\_multi.h.

### 6.20.5.2 matrix\_log

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
template<typename Other_Scalar_T , const index_t Other_LO, const index_t Other_HI, typename
Other_Tune_P >
auto matrix_log (
    const matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & val,
    const matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & i,
    const index_t level ) -> const matrix_multi< Other_Scalar_T, Other_LO, Other_HI,
Other_Tune_P > [friend]
```

### 6.20.5.3 matrix\_multi

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
template<typename Other_Scalar_T , const index_t Other_LO, const index_t Other_HI, typename
Other_Tune_P >
friend class matrix_multi [friend]
```

Definition at line 153 of file matrix\_multi.h.

### 6.20.5.4 matrix\_sqrt

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAU↵
LT_HI, typename Tune_P = tuning<>>
template<typename Other_Scalar_T , const index_t Other_LO, const index_t Other_HI, typename
Other_Tune_P >
auto matrix_sqrt (
    const matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & val,
    const matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & i,
    const index_t level ) -> const matrix_multi< Other_Scalar_T, Other_LO, Other_HI,
Other_Tune_P > [friend]
```



#### 6.20.5.5 operator &

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator& (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> const matrix_multi_t [friend]
```

#### 6.20.5.6 operator%

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator% (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> const matrix_multi_t [friend]
```

#### 6.20.5.7 operator\*

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator* (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> const matrix_multi_t [friend]
```

#### 6.20.5.8 operator/

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator/ (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> const matrix_multi_t [friend]
```

#### 6.20.5.9 operator<<

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator<< (
    std::ostream & os,
    const multivector_t & val ) -> std::ostream & [friend]
```

#### 6.20.5.10 operator>>

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator>> (
    std::istream & s,
    multivector_t & val ) -> std::istream & [friend]
```

#### 6.20.5.11 operator^

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator^ (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> const matrix_multi_t [friend]
```

#### 6.20.5.12 operator"|"

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto operator| (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> const matrix_multi_t [friend]
```

#### 6.20.5.13 reframe

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
template<typename Other_Scalar_T, const index_t Other_LO, const index_t Other_HI, typename Other_Tune_P >
auto reframe (
    const matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & lhs,
    const matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & rhs,
    matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & lhs_reframed,
    matrix_multi< Other_Scalar_T, Other_LO, Other_HI, Other_Tune_P > & rhs_reframed )
-> const index_set< Other_LO, Other_HI > [friend]
```

#### 6.20.5.14 star

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto star (
    const matrix_multi_t & lhs,
    const matrix_multi_t & rhs ) -> Scalar_T [friend]
```

## 6.20.6 Member Data Documentation

### 6.20.6.1 m\_frame

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
index_set_t glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_frame [private]
```

Index set representing the frame for the subalgebra which contains the multivector.

Definition at line 278 of file matrix\_multi.h.

Referenced by glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().

### 6.20.6.2 m\_matrix

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
matrix_t glucat::matrix_multi< Scalar_T, LO, HI, Tune_P >::m_matrix [private]
```

Matrix value representing the multivector within the folded frame.

Definition at line 280 of file matrix\_multi.h.

Referenced by glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::framed\_multi(), and glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >::matrix\_multi().

The documentation for this class was generated from the following files:

- glucat/framed\_multi.h
- glucat/matrix\_multi.h
- glucat/matrix\_multi\_imp.h

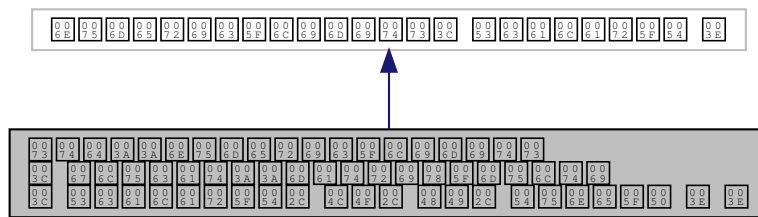


## 6.22 std::numeric\_limits< glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P > > Struct Template Reference

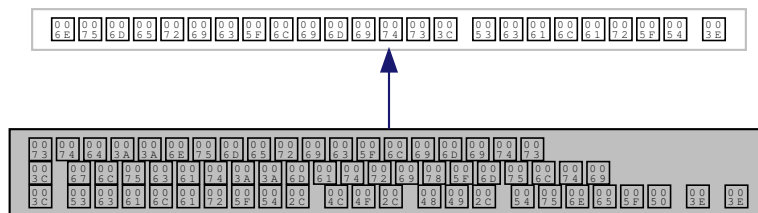
Numeric limits for matrix\_multi inherit limits for the corresponding scalar type.

```
#include <matrix_multi.h>
```

Inheritance diagram for std::numeric\_limits< glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >:



Collaboration diagram for std::numeric\_limits< glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P > >:



### 6.22.1 Detailed Description

```
template<typename Scalar_T, const glucat::index_t LO, const glucat::index_t HI, typename Tune_P>
struct std::numeric_limits< glucat::matrix_multi< Scalar_T, LO, HI, Tune_P > >
```

Numeric limits for matrix\_multi inherit limits for the corresponding scalar type.

Definition at line 296 of file `matrix_multi.h`.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi.h](#)

## 6.23 glucat::numeric\_traits< Scalar\_T > Class Template Reference

Extra traits which extend numeric limits.

```
#include <scalar.h>
```

### Classes

- struct [demoted](#)  
*Demoted type for long double.*
- struct [promoted](#)  
*Extra traits which extend numeric limits.*

### Public Member Functions

- `template<>`  
`auto pi () -> long double`  
*Pi for long double.*
- `template<>`  
`auto ln\_2 () -> long double`  
*log(2) for long double*
- `template<>`  
`auto to\_scalar\_t (const Other_Scalar_T &val) -> float`  
*Extra traits which extend numeric limits.*
- `template<>`  
`auto to\_scalar\_t (const Other_Scalar_T &val) -> double`  
*Cast to double.*
- `template<>`  
`auto to\_scalar\_t (const dd_real &val) -> long double`  
*Cast to long double.*
- `template<>`  
`auto to\_scalar\_t (const qd_real &val) -> long double`  
*Cast to long double.*
- `template<>`  
`auto to\_scalar\_t (const long double &val) -> dd_real`  
*Cast to dd\_real.*
- `template<>`  
`auto to\_scalar\_t (const qd_real &val) -> dd_real`  
*Cast to dd\_real.*
- `template<>`  
`auto to\_scalar\_t (const long double &val) -> qd_real`  
*Cast to qd\_real.*
- `template<>`  
`auto to\_scalar\_t (const dd_real &val) -> qd_real`  
*Cast to qd\_real.*

## Static Public Member Functions

- static auto `isInf` (const Scalar\_T &val) -> bool  
*Smart isinf.*
- static auto `isNaN` (const Scalar\_T &val) -> bool  
*Smart isnan.*
- static auto `isNaN_or_isInf` (const Scalar\_T &val) -> bool  
*Smart isnan or isinf.*
- static auto `NaN` () -> Scalar\_T  
*Smart NaN.*
- static auto `to_int` (const Scalar\_T &val) -> int  
*Cast to int.*
- static auto `to_double` (const Scalar\_T &val) -> double  
*Cast to double.*
- template<typename Other\_Scalar\_T >  
static auto `to_scalar_t` (const Other\_Scalar\_T &val) -> Scalar\_T  
*Cast to Scalar\_T.*
- static auto `fmod` (const Scalar\_T &lhs, const Scalar\_T &rhs) -> Scalar\_T  
*Modulo function for scalar.*
- static auto `conj` (const Scalar\_T &val) -> Scalar\_T  
*Complex conjugate of scalar.*
- static auto `real` (const Scalar\_T &val) -> Scalar\_T  
*Real part of scalar.*
- static auto `imag` (const Scalar\_T &val) -> Scalar\_T  
*Imaginary part of scalar.*
- static auto `abs` (const Scalar\_T &val) -> Scalar\_T  
*Absolute value of scalar.*
- static auto `pi` () -> Scalar\_T  
*Pi.*
- static auto `ln_2` () -> Scalar\_T  
*log(2)*
- static auto `pow` (const Scalar\_T &val, int n) -> Scalar\_T  
*Integer power.*
- static auto `sqrt` (const Scalar\_T &val) -> Scalar\_T  
*Square root of scalar.*
- static auto `exp` (const Scalar\_T &val) -> Scalar\_T  
*Exponential.*
- static auto `log` (const Scalar\_T &val) -> Scalar\_T  
*Logarithm of scalar.*
- static auto `log2` (const Scalar\_T &val) -> Scalar\_T  
*Log base 2.*
- static auto `cos` (const Scalar\_T &val) -> Scalar\_T  
*Cosine of scalar.*
- static auto `acos` (const Scalar\_T &val) -> Scalar\_T  
*Inverse cosine of scalar.*
- static auto `cosh` (const Scalar\_T &val) -> Scalar\_T  
*Hyperbolic cosine of scalar.*
- static auto `sin` (const Scalar\_T &val) -> Scalar\_T  
*Sine of scalar.*
- static auto `asin` (const Scalar\_T &val) -> Scalar\_T  
*Inverse sine of scalar.*

- static auto [sinh](#) (const Scalar\_T &val) -> Scalar\_T  
*Hyperbolic sine of scalar.*
- static auto [tan](#) (const Scalar\_T &val) -> Scalar\_T  
*Tangent of scalar.*
- static auto [atan](#) (const Scalar\_T &val) -> Scalar\_T  
*Inverse tangent of scalar.*
- static auto [tanh](#) (const Scalar\_T &val) -> Scalar\_T  
*Hyperbolic tangent of scalar.*

### Static Private Member Functions

- static auto [isInf](#) (const Scalar\_T &val, [bool\\_to\\_type](#)< false >) -> bool  
*Smart isinf specialised for Scalar\_T without infinity.*
- static auto [isInf](#) (const Scalar\_T &val, [bool\\_to\\_type](#)< true >) -> bool  
*Smart isinf specialised for Scalar\_T with infinity.*
- static auto [isNaN](#) (const Scalar\_T &val, [bool\\_to\\_type](#)< false >) -> bool  
*Smart isnan specialised for Scalar\_T without quiet NaN.*
- static auto [isNaN](#) (const Scalar\_T &val, [bool\\_to\\_type](#)< true >) -> bool  
*Smart isnan specialised for Scalar\_T with quiet NaN.*

### 6.23.1 Detailed Description

```
template<typename Scalar_T>
class glucat::numeric_traits< Scalar_T >
```

Extra traits which extend numeric limits.

Definition at line 47 of file scalar.h.

### 6.23.2 Member Function Documentation

#### 6.23.2.1 [abs\(\)](#)

```
template<typename Scalar_T >
static auto glucat::numeric\_traits< Scalar_T >::abs (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Absolute value of scalar.

Definition at line 182 of file scalar.h.

References [UBLAS\\_ABS](#).



#### 6.23.2.2 acos()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::acos (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Inverse cosine of scalar.

Definition at line 245 of file scalar.h.

References glucat::acos().

#### 6.23.2.3 asin()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::asin (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Inverse sine of scalar.

Definition at line 266 of file scalar.h.

References glucat::asin().

#### 6.23.2.4 atan()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::atan (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Inverse tangent of scalar.

Definition at line 287 of file scalar.h.

References glucat::atan().

#### 6.23.2.5 conj()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::conj (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Complex conjugate of scalar.

Definition at line 161 of file scalar.h.

#### 6.23.2.6 cos()

```
template<typename Scalar_T >
static auto glucat::numeric\_traits< Scalar_T >::cos (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Cosine of scalar.

Definition at line 238 of file scalar.h.

References [glucat::cos\(\)](#).

#### 6.23.2.7 cosh()

```
template<typename Scalar_T >
static auto glucat::numeric\_traits< Scalar_T >::cosh (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Hyperbolic cosine of scalar.

Definition at line 252 of file scalar.h.

References [glucat::cosh\(\)](#).

#### 6.23.2.8 exp()

```
template<typename Scalar_T >
static auto glucat::numeric\_traits< Scalar_T >::exp (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Exponential.

Definition at line 217 of file scalar.h.

References [glucat::exp\(\)](#).

#### 6.23.2.9 fmod()

```
template<typename Scalar_T >
static auto glucat::numeric\_traits< Scalar_T >::fmod (
    const Scalar_T & lhs,
    const Scalar_T & rhs ) -> Scalar_T    [inline], [static]
```

Modulo function for scalar.

Definition at line 154 of file scalar.h.

References [PyClical::lhs](#), and [PyClical::rhs](#).

## 6.23.2.10 imag()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::imag (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Imaginary part of scalar.

Definition at line 175 of file scalar.h.

## 6.23.2.11 isInf() [1/3]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isInf (
    const Scalar_T & val,
    bool_to_type< false > ) -> bool    [inline], [static], [private]
```

Smart isinf specialised for Scalar\_T without infinity.

Definition at line 54 of file scalar.h.

Referenced by glucat::numeric\_traits< Scalar\_T >::isInf(), and glucat::numeric\_traits< Scalar\_T >::isNaN\_or\_isInf().

## 6.23.2.12 isInf() [2/3]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isInf (
    const Scalar_T & val,
    bool_to_type< true > ) -> bool    [inline], [static], [private]
```

Smart isinf specialised for Scalar\_T with infinity.

Definition at line 61 of file scalar.h.

References \_GLUCAT\_ISINF.

## 6.23.2.13 isInf() [3/3]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isInf (
    const Scalar_T & val ) -> bool    [inline], [static]
```

Smart isinf.

Definition at line 83 of file scalar.h.

References glucat::numeric\_traits< Scalar\_T >::isInf().

**6.23.2.14 isNaN()** [1/3]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isNaN (
    const Scalar_T & val,
    bool_to_type< false > ) -> bool    [inline], [static], [private]
```

Smart isnan specialised for Scalar\_T without quiet NaN.

Definition at line 68 of file scalar.h.

Referenced by glucat::numeric\_traits< Scalar\_T >::isNaN(), and glucat::numeric\_traits< Scalar\_T >::isNaN\_or\_isInf().

**6.23.2.15 isNaN()** [2/3]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isNaN (
    const Scalar_T & val,
    bool_to_type< true > ) -> bool    [inline], [static], [private]
```

Smart isnan specialised for Scalar\_T with quiet NaN.

Definition at line 75 of file scalar.h.

References \_GLUCAT\_ISNAN.

**6.23.2.16 isNaN()** [3/3]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isNaN (
    const Scalar_T & val ) -> bool    [inline], [static]
```

Smart isnan.

Definition at line 93 of file scalar.h.

References glucat::numeric\_traits< Scalar\_T >::isNaN().

**6.23.2.17 isNaN\_or\_isInf()**

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::isNaN_or_isInf (
    const Scalar_T & val ) -> bool    [inline], [static]
```

Smart isnan or isinf.

Definition at line 103 of file scalar.h.

References glucat::numeric\_traits< Scalar\_T >::isInf(), and glucat::numeric\_traits< Scalar\_T >::isNaN().

**6.23.2.18** `ln_2()` [1/2]

```
template<>
auto glucat::numeric_traits< long double >::ln_2 ( ) -> long double    [inline]
```

`log(2)` for long double

Definition at line 60 of file `long_double.h`.

References `glucat::l_ln2`.

**6.23.2.19** `ln_2()` [2/2]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::ln_2 ( ) -> Scalar_T    [inline], [static]
```

`log(2)`

Definition at line 196 of file `scalar.h`.

Referenced by `glucat::numeric_traits< Scalar_T >::log2()`.

**6.23.2.20** `log()`

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::log (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Logarithm of scalar.

Definition at line 224 of file `scalar.h`.

References `glucat::log()`.

Referenced by `glucat::numeric_traits< Scalar_T >::log2()`.

**6.23.2.21** `log2()`

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::log2 (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Log base 2.

Definition at line 231 of file `scalar.h`.

References `glucat::numeric_traits< Scalar_T >::ln_2()`, and `glucat::numeric_traits< Scalar_T >::log()`.

Referenced by `glucat::log2()`.

**6.23.2.22 NaN()**

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::NaN ( ) -> Scalar_T    [inline], [static]
```

Smart NaN.

Definition at line 115 of file scalar.h.

References glucat::log().

Referenced by glucat::cr\_sqrt(), and glucat::matrix::trace().

**6.23.2.23 pi()** [1/2]

```
template<>
auto glucat::numeric_traits< long double >::pi ( ) -> long double    [inline]
```

Pi for long double.

Definition at line 52 of file long\_double.h.

References glucat::l\_pi.

**6.23.2.24 pi()** [2/2]

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::pi ( ) -> Scalar_T    [inline], [static]
```

Pi.

Definition at line 189 of file scalar.h.

Referenced by glucat::matrix::classify\_eigenvalues().

**6.23.2.25 pow()**

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::pow (
    const Scalar_T & val,
    int n ) -> Scalar_T    [inline], [static]
```

Integer power.

Definition at line 203 of file scalar.h.

References glucat::pow().

Referenced by glucat::error\_squared\_tol().

#### 6.23.2.26 real()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::real (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Real part of scalar.

Definition at line 168 of file scalar.h.

#### 6.23.2.27 sin()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::sin (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Sine of scalar.

Definition at line 259 of file scalar.h.

References glucat::sin().

#### 6.23.2.28 sinh()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::sinh (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Hyperbolic sine of scalar.

Definition at line 273 of file scalar.h.

References glucat::sinh().

#### 6.23.2.29 sqrt()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::sqrt (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Square root of scalar.

Definition at line 210 of file scalar.h.

References UBLAS\_SQRT.

Referenced by glucat::abs().

#### 6.23.2.30 tan()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::tan (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Tangent of scalar.

Definition at line 280 of file scalar.h.

References glucat::tan().

#### 6.23.2.31 tanh()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::tanh (
    const Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Hyperbolic tangent of scalar.

Definition at line 294 of file scalar.h.

References glucat::tanh().

#### 6.23.2.32 to\_double()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::to_double (
    const Scalar_T & val ) -> double    [inline], [static]
```

Cast to double.

Definition at line 133 of file scalar.h.

Referenced by glucat::operator<<(), and glucat::numeric\_traits< Scalar\_T >::to\_scalar\_t().

#### 6.23.2.33 to\_int()

```
template<typename Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::to_int (
    const Scalar_T & val ) -> int    [inline], [static]
```

Cast to int.

Definition at line 126 of file scalar.h.



**6.23.2.34** to\_scalar\_t() [1/9]

```
template<>
auto glucat::numeric_traits< float >::to_scalar_t (
    const Other_Scalar_T & val ) -> float    [inline]
```

Extra traits which extend numeric limits.

Cast to float

Definition at line 53 of file scalar\_imp.h.

References glucat::numeric\_traits< Scalar\_T >::to\_double().

**6.23.2.35** to\_scalar\_t() [2/9]

```
template<>
auto glucat::numeric_traits< double >::to_scalar_t (
    const Other_Scalar_T & val ) -> double    [inline]
```

Cast to double.

Definition at line 62 of file scalar\_imp.h.

References glucat::numeric\_traits< Scalar\_T >::to\_double().

**6.23.2.36** to\_scalar\_t() [3/9]

```
template<>
auto glucat::numeric_traits< long double >::to_scalar_t (
    const dd_real & val ) -> long double    [inline]
```

Cast to long double.

Definition at line 72 of file scalar\_imp.h.

**6.23.2.37** to\_scalar\_t() [4/9]

```
template<>
auto glucat::numeric_traits< long double >::to_scalar_t (
    const qd_real & val ) -> long double    [inline]
```

Cast to long double.

Definition at line 81 of file scalar\_imp.h.

**6.23.2.38 to\_scalar\_t()** [ 5/9]

```
template<>
auto glucat::numeric_traits< dd_real >::to_scalar_t (
    const long double & val ) -> dd_real    [inline]
```

Cast to dd\_real.

Definition at line 90 of file scalar\_imp.h.

**6.23.2.39 to\_scalar\_t()** [ 6/9]

```
template<>
auto glucat::numeric_traits< dd_real >::to_scalar_t (
    const qd_real & val ) -> dd_real    [inline]
```

Cast to dd\_real.

Definition at line 99 of file scalar\_imp.h.

**6.23.2.40 to\_scalar\_t()** [ 7/9]

```
template<>
auto glucat::numeric_traits< qd_real >::to_scalar_t (
    const long double & val ) -> qd_real    [inline]
```

Cast to qd\_real.

Definition at line 108 of file scalar\_imp.h.

**6.23.2.41 to\_scalar\_t()** [ 8/9]

```
template<>
auto glucat::numeric_traits< qd_real >::to_scalar_t (
    const dd_real & val ) -> qd_real    [inline]
```

Cast to qd\_real.

Definition at line 117 of file scalar\_imp.h.

## 6.23.2.42 to\_scalar\_t() [ 9/9]

```
template<typename Scalar_T >
template<typename Other_Scalar_T >
static auto glucat::numeric_traits< Scalar_T >::to_scalar_t (
    const Other_Scalar_T & val ) -> Scalar_T    [inline], [static]
```

Cast to Scalar\_T.

Definition at line 141 of file scalar.h.

Referenced by glucat::matrix::nork\_range(), glucat::to\_demote(), and glucat::to\_promote().

The documentation for this class was generated from the following file:

- glucat/[scalar.h](#)

## 6.24 pade::pade\_log\_denom< Scalar\_T > Struct Template Reference

Coefficients of denominator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< Scalar\_T, 14 >

### Static Public Attributes

- static const [array](#) [denom](#)

#### 6.24.1 Detailed Description

```
template<typename Scalar_T>
struct pade::pade_log_denom< Scalar_T >
```

Coefficients of denominator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)

Definition at line 1731 of file matrix\_multi\_imp.h.

#### 6.24.2 Member Typedef Documentation

### 6.24.2.1 array

```
template<typename Scalar_T>
using pade::pade_log_denom< Scalar_T >::array = std::array<Scalar_T, 14>
```

Definition at line 1733 of file matrix\_multi\_imp.h.

## 6.24.3 Member Data Documentation

### 6.24.3.1 denom

```
template<typename Scalar_T>
const pade_log_denom< Scalar_T >::array pade::pade_log_denom< Scalar_T >::denom [static]
```

**Initial value:**

```
=
{
    1.0,                13.0/2.0,                468.0/25.0,                1573.0/50.0,
    1573.0/46.0,        11583.0/460.0,            10296.0/805.0,            2574.0/575.0,
    11583.0/10925.0,    143.0/874.0,                572.0/37145.0,            117.0/148580.0,
    13.0/742900.0,     1.0/10400600.0
}
```

Definition at line 1734 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.25 pade::pade\_log\_denom< dd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< dd\_real, 22 >

### Static Public Attributes

- static const [array](#) [denom](#)

### 6.25.1 Detailed Description

```
template<>
struct pade::pade_log_denom< dd_real >
```

Definition at line 1820 of file `matrix_multi_imp.h`.

### 6.25.2 Member Typedef Documentation

#### 6.25.2.1 array

```
using pade::pade_log_denom< dd_real >::array = std::array<dd_real, 22>
```

Definition at line 1822 of file `matrix_multi_imp.h`.

### 6.25.3 Member Data Documentation

#### 6.25.3.1 denom

```
const pade_log_denom< dd_real >::array pade::pade_log_denom< dd_real >::denom [static]
```

**Initial value:**

```
=
{
    dd_real("1"),
    dd_real("2100")/dd_real("41"),
    dd_real("341145")/dd_real("1066"),
    dd_real("11069856")/dd_real("19721"),
    dd_real("6918660")/dd_real("19721"),
    dd_real("1410864")/dd_real("16687"),
    dd_real("734825")/dd_real("94054"),
    dd_real("348840")/dd_real("1363783"),
    dd_real("6783")/dd_real("2727566"),
    dd_real("266")/dd_real("53187537"),
    dd_real("7")/dd_real("8155422340"),
    dd_real("21")/dd_real("2"),
    dd_real("12635")/dd_real("82"),
    dd_real("1037799")/dd_real("2132"),
    dd_real("9883800")/dd_real("19721"),
    dd_real("293930")/dd_real("1517"),
    dd_real("88179")/dd_real("3034"),
    dd_real("305235")/dd_real("188108"),
    dd_real("40698")/dd_real("1363783"),
    dd_real("9975")/dd_real("70916716"),
    dd_real("7")/dd_real("70916716"),
    dd_real("1")/dd_real("538257874440")
}
```

Definition at line 1823 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.26 `pade::pade_log_denom< float >` Struct Template Reference

```
#include <matrix_multi_imp.h>
```

## Public Types

- using [array](#) = std::array< float, 10 >

## Static Public Attributes

- static const [array](#) [denom](#)

### 6.26.1 Detailed Description

```
template<>
struct pade::pade_log_denom< float >
```

Definition at line 1758 of file matrix\_multi\_imp.h.

### 6.26.2 Member Typedef Documentation

#### 6.26.2.1 array

```
using pade::pade_log_denom< float >::array = std::array<float, 10>
```

Definition at line 1760 of file matrix\_multi\_imp.h.

### 6.26.3 Member Data Documentation

#### 6.26.3.1 denom

```
const pade_log_denom< float >::array pade::pade_log_denom< float >::denom [static]
```

#### Initial value:

```
=
{
    1.0,          9.0/2.0,          144.0/17.0,    147.0/17.0,
    441.0/85.0,    63.0/34.0,          84.0/221.0,    9.0/221.0,
    9.0/4862.0,    1.0/48620.0
}
```

Definition at line 1761 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)

## 6.27 `pade::pade_log_denom< long double >` Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using `array` = `std::array< long double, 18 >`

### Static Public Attributes

- static const `array` `denom`

#### 6.27.1 Detailed Description

```
template<>
struct pade::pade_log_denom< long double >
```

Definition at line 1785 of file `matrix_multi_imp.h`.

#### 6.27.2 Member Typedef Documentation

##### 6.27.2.1 `array`

```
using pade::pade_log_denom< long double >::array = std::array<long double, 18>
```

Definition at line 1787 of file `matrix_multi_imp.h`.

#### 6.27.3 Member Data Documentation

##### 6.27.3.1 `denom`

```
const pade_log_denom< long double >::array pade::pade_log_denom< long double >::denom [static]
```

##### Initial value:

```
=
{
    1.0L,                17.0L/2.0L,                1088.0L/33.0L,                850.0L/11.0L
    ,
    41650.0L/341.0L,      140777.0L/1023.0L,          1126216.0L/9889.0L,          63206.0L/899.0
    L,
    790075.0L/24273.0L,    60775.0L/5394.0L,                38896.0L/13485.0L,          21658.0L/40455
    .0L,
    21658.0L/310155.0L,    4165.0L/682341.0L,                680.0L/2047023.0L,          34.0L/
    3411705.0L,
    17.0L/129644790.0L,    1.0L/2333606220
}
```

Definition at line 1788 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- `glucat/matrix_multi_imp.h`

## 6.28 pade::pade\_log\_denom< qd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< qd\_real, 34 >

### Static Public Attributes

- static const [array](#) [denom](#)

#### 6.28.1 Detailed Description

```
template<>
struct pade::pade_log_denom< qd_real >
```

Definition at line 1867 of file matrix\_multi\_imp.h.

#### 6.28.2 Member Typedef Documentation

##### 6.28.2.1 array

```
using pade::pade\_log\_denom< qd_real >::array = std::array<qd_real, 34>
```

Definition at line 1869 of file matrix\_multi\_imp.h.

#### 6.28.3 Member Data Documentation



## 6.28.3.1 denom

```
const pade_log_denom< qd_real >::array pade::pade_log_denom< qd_real >::denom [static]
```

**Initial value:**

```
=
{
    qd_real("1"),
    qd_real("2"),
    qd_real("65"),
    qd_real("91"),
    qd_real("3965"),
    qd_real("3599"),
    qd_real("478667"),
    qd_real("68381"),
    qd_real("2479711"),
    qd_real("4959422"),
    qd_real("121505839"),
    qd_real("62755763"),
    qd_real("62755763"),
    qd_real("18889484663"),
    qd_real("79027435835"),
    qd_real("27327687311743"),
    qd_real("1011124430534491"),
    qd_real("7219428434016265740"),
    qd_real("8448")/qd_real("65"),
    qd_real("211420")/qd_real("91"),
    qd_real("32119472")/qd_real("2379"),
    qd_real("603960786")/qd_real("17995"),
    qd_real("2776831200")/qd_real("68381"),
    qd_real("12241197540")/qd_real("478667"),
    qd_real("31387686000")/qd_real("3624193"),
    qd_real("67091178825")/qd_real("42155087"),
    qd_real("19083713088")/qd_real("121505839"),
    qd_real("941630580")/qd_real("116546417"),
    qd_real("12902448")/qd_real("62755763"),
    qd_real("6427850")/qd_real("2698497809"),
    qd_real("8544096")/qd_real("774468871183"),
    qd_real("118668")/qd_real("7191496660985"),
    qd_real("5456")/qd_real("1011124430534491"),
    qd_real("11")/qd_real("70778710137414370"),
    qd_real("33")/qd_real("42284"),
    qd_real("573562")/qd_real("92917044"),
    qd_real("144626625")/qd_real("16692542100"),
    qd_real("1098569010")/qd_real("9939433900"),
    qd_real("2683647153")/qd_real("4708152900"),
    qd_real("88704330")/qd_real("1542684"),
    qd_real("3471039")/qd_real("39556"),
    qd_real("10230")/qd_real("44"),
    qd_real("1")/qd_real("1")
}
```

Definition at line 1870 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.29 pade::pade\_log\_numer< Scalar\_T > Struct Template Reference

Coefficients of numerator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< Scalar\_T, 14 >

### Static Public Attributes

- static const [array numer](#)

## 6.29.1 Detailed Description

```
template<typename Scalar_T>
struct pade::pade_log_numer< Scalar_T >
```

Coefficients of numerator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)

Definition at line 1714 of file matrix\_multi\_imp.h.

## 6.29.2 Member Typedef Documentation

### 6.29.2.1 array

```
template<typename Scalar_T>
using pade::pade_log_numer< Scalar_T >::array = std::array<Scalar_T, 14>
```

Definition at line 1716 of file matrix\_multi\_imp.h.

## 6.29.3 Member Data Documentation

### 6.29.3.1 numer

```
template<typename Scalar_T>
const pade_log_numer< Scalar_T >::array pade::pade_log_numer< Scalar_T >::numer [static]
```

**Initial value:**

```
=
{
    0.0,          1.0,          6.0,          4741.0/300.0,
    1441.0/60.0,  107091.0/4600.0,  8638.0/575.0,  263111.0/40250.0,
    153081.0/80500.0,  395243.0/1101240.0,  28549.0/688275.0,  605453.0/228813200.0,
    785633.0/10296594000.0,  1145993.0/1873980108000.0
}
```

Definition at line 1717 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.30 pade::pade\_log\_numer< dd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

## Public Types

- using [array](#) = std::array< dd\_real, 22 >

## Static Public Attributes

- static const [array](#) [number](#)

### 6.30.1 Detailed Description

```
template<>
struct pade::pade_log_number< dd_real >
```

Definition at line 1800 of file matrix\_multi\_imp.h.

### 6.30.2 Member Typedef Documentation

#### 6.30.2.1 array

```
using pade::pade_log_number< dd_real >::array = std::array<dd_real, 22>
```

Definition at line 1802 of file matrix\_multi\_imp.h.

### 6.30.3 Member Data Documentation

#### 6.30.3.1 numer

```
const pade_log_number< dd_real >::array pade::pade_log_number< dd_real >::numer [static]
```

**Initial value:**

```
=
{
    dd_real("0"),
    dd_real("10"),
    dd_real("21603")/dd_real("164"),
    dd_real("978724")/dd_real("2665"),
    dd_real("12874933")/dd_real("39442"),
    dd_real("2406734")/dd_real("22755"),
    dd_real("30653165")/dd_real("2402928"),
    dd_real("25346331")/dd_real("47074027"),
    dd_real("105689791")/dd_real("15601677520"),
    dd_real("969715")/dd_real("53502994116"),
    dd_real("118999")/dd_real("26204577562592"),
    dd_real("1"),
    dd_real("22781")/dd_real("492"),
    dd_real("5492649")/dd_real("21320"),
    dd_real("4191605")/dd_real("10619"),
    dd_real("11473457")/dd_real("54612"),
    dd_real("166770367")/dd_real("4004880"),
    dd_real("647746389")/dd_real("215195552"),
    dd_real("278270613")/dd_real("3900419380"),
    dd_real("606046475")/dd_real("1379188292768"),
    dd_real("11098301")/dd_real("26204577562592"),
    dd_real("18858053")/dd_real("1392249205900512960")
}
```

Definition at line 1803 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)

## 6.31 pade::pade\_log\_numer< float > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< float, 10 >

### Static Public Attributes

- static const [array](#) [numer](#)

#### 6.31.1 Detailed Description

```
template<>
struct pade::pade_log_numer< float >
```

Definition at line 1746 of file matrix\_multi\_imp.h.

#### 6.31.2 Member Typedef Documentation

##### 6.31.2.1 array

```
using pade::pade_log_numer< float >::array = std::array<float, 10>
```

Definition at line 1748 of file matrix\_multi\_imp.h.

#### 6.31.3 Member Data Documentation

##### 6.31.3.1 numer

```
const pade_log_numer< float >::array pade::pade_log_numer< float >::numer [static]
```

##### Initial value:

```
=
{
    0.0,          1.0,          4.0,          1337.0/204.0,
    385.0/68.0,   1879.0/680.0,   193.0/255.0,   197.0/1820.0,
    419.0/61880.0, 7129.0/61261200.0
}
```

Definition at line 1749 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)

## 6.32 `pade::pade_log_number< long double >` Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using `array` = `std::array< long double, 18 >`

### Static Public Attributes

- static const `array` `number`

#### 6.32.1 Detailed Description

```
template<>
struct pade::pade_log_number< long double >
```

Definition at line 1771 of file `matrix_multi_imp.h`.

#### 6.32.2 Member Typedef Documentation

##### 6.32.2.1 `array`

```
using pade::pade_log_number< long double >::array = std::array<long double, 18>
```

Definition at line 1773 of file `matrix_multi_imp.h`.

#### 6.32.3 Member Data Documentation

##### 6.32.3.1 `number`

```
const pade_log_number< long double >::array pade::pade_log_number< long double >::number [static]
```

##### Initial value:

```
=
{
    0.0L,          1.0L,          8.0L,          3835.0L/132.0
    L,
    8365.0L/132.0L, 11363807.0L/122760.0L, 162981.0L/1705.0L, 9036157.0L/
    125860.0L,
    18009875.0L/453096.0L, 44211925.0L/2718576.0L, 4149566.0L/849555.0L, 16973929.0L/
    16020180.0L,
    172459.0L/1068012.0L, 116317061.0L/7025382936.0L, 19679783.0L/18441630207.0L, 23763863.0L/
    614721006900.0L,
    50747.0L/79318839600.0L, 42142223.0L/14295951736466400.0L
}
```

Definition at line 1774 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- `glucat/matrix_multi_imp.h`

## 6.33 pade::pade\_log\_numer< qd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using `array` = `std::array< qd_real, 34 >`

### Static Public Attributes

- static const `array numer`

#### 6.33.1 Detailed Description

```
template<>  
struct pade::pade_log_numer< qd_real >
```

Definition at line 1841 of file `matrix_multi_imp.h`.

#### 6.33.2 Member Typedef Documentation

##### 6.33.2.1 array

```
using pade::pade_log_numer< qd_real >::array = std::array<qd_real, 34>
```

Definition at line 1843 of file `matrix_multi_imp.h`.

#### 6.33.3 Member Data Documentation

## 6.33.3.1 numer

```
const pade_log_numer< qd_real >::array pade::pade_log_numer< qd_real >::numer [static]
```

## Initial value:

```
=
{
    qd_real("0"),
    qd_real("16"),
    ("780"),
    qd_real("30721")/qd_real("52"),
    ("3640"),
    qd_real("1039099")/qd_real("195"),
    ("555100"),
    qd_real("1564058073")/qd_real("85400"),
    ("1209264"),
    qd_real("725351278")/qd_real("25193"),
    ("147429436"),
    qd_real("4559713849589")/qd_real("201040140"),
    ("320023080"),
    qd_real("74979677195")/qd_real("8000577"),
    ("3481514244"),
    qd_real("1065906022369")/qd_real("515779888"),
    ("438412904800"),
    qd_real("1462444287585964")/qd_real("6041877844275"),
    ("6122436215532"),
    qd_real("64211291334131")/qd_real("4373168725380"),
    ("51080680851480"),
    qd_real("154355972958659")/qd_real("351179680853925"),
    ("2937139148960100"),
    qd_real("4230788929433")/qd_real("704913395750424"),
    ("392923948371995600"),
    qd_real("10537522306718")/qd_real("319250708052246425"),
    ("144249197475035425500"),
    qd_real("260715545088119")/qd_real("4375558990076074573500"),
    ("192874640282553367199880"),
    qd_real("8802625510547")/qd_real("361639950529787563499775"),
    ("1659204093030665341336967700"),
    qd_real("446033437968239")/qd_real("464577146048586295574350956000"),
    ("47386868896955802148583797512000")
}
```

Definition at line 1844 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.34 pade::pade\_sqrt\_denom&lt; Scalar\_T &gt; Struct Template Reference

Coefficients of denominator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)

```
#include <matrix_multi_imp.h>
```

## Public Types

- using [array](#) = std::array< Scalar\_T, 14 >

## Static Public Attributes

- static const [array](#) denom

### 6.34.1 Detailed Description

```
template<typename Scalar_T>
struct pade::pade_sqrt_denom< Scalar_T >
```

Coefficients of denominator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)

Definition at line 1401 of file matrix\_multi\_imp.h.

### 6.34.2 Member Typedef Documentation

#### 6.34.2.1 array

```
template<typename Scalar_T>
using pade::pade_sqrt_denom< Scalar_T >::array = std::array<Scalar_T, 14>
```

Definition at line 1403 of file matrix\_multi\_imp.h.

### 6.34.3 Member Data Documentation

#### 6.34.3.1 denom

```
template<typename Scalar_T>
const pade_sqrt_denom< Scalar_T >::array pade::pade_sqrt_denom< Scalar_T >::denom [static]
```

**Initial value:**

```
=
{
    1.0,                25.0/4.0,        69.0/4.0,        1771.0/64.0,
    7315.0/256.0,       20349.0/1024.0,   4845.0/512.0,   12597.0/4096.0,
    21879.0/32768.0,    12155.0/131072.0,  1001.0/131072.0,  1365.0/4194304.0,
    91.0/16777216.0,   1.0/67108864.0
}
```

Definition at line 1404 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.35 pade::pade\_sqrt\_denom< dd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```



## Public Types

- using [array](#) = std::array< dd\_real, 22 >

## Static Public Attributes

- static const [array](#) [denom](#)

### 6.35.1 Detailed Description

```
template<>
struct pade::pade_sqrt_denom< dd_real >
```

Definition at line 1491 of file matrix\_multi\_imp.h.

### 6.35.2 Member Typedef Documentation

#### 6.35.2.1 array

```
using pade::pade_sqrt_denom< dd_real >::array = std::array<dd_real, 22>
```

Definition at line 1493 of file matrix\_multi\_imp.h.

### 6.35.3 Member Data Documentation

#### 6.35.3.1 denom

```
const pade_sqrt_denom< dd_real >::array pade::pade_sqrt_denom< dd_real >::denom [static]
```

**Initial value:**

```
=
{
    dd_real("1"),
    dd_real("195")/dd_real("4"),
    dd_real("73815")/dd_real("256"),
    dd_real("121737")/dd_real("256"),
    dd_real("4539051")/dd_real("16384"),
    dd_real("4032015")/dd_real("65536"),
    dd_real("86493225")/dd_real("16777216"),
    dd_real("5014575")/dd_real("33554432"),
    dd_real("5311735")/dd_real("4294967296"),
    dd_real("33649")/dd_real("17179869184"),
    dd_real("231")/dd_real("1099511627776"),
    dd_real("41")/dd_real("4"),
    dd_real("9139")/dd_real("64"),
    dd_real("435897")/dd_real("1024"),
    dd_real("840565")/dd_real("2048"),
    dd_real("9641775")/dd_real("65536"),
    dd_real("84672315")/dd_real("4194304"),
    dd_real("67863915")/dd_real("67108864"),
    dd_real("4345965")/dd_real("268435456"),
    dd_real("1081575")/dd_real("17179869184"),
    dd_real("8855")/dd_real("274877906944"),
    dd_real("1")/dd_real("4398046511104")
}
```

Definition at line 1494 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)

## 6.36 `pade::pade_sqrt_denom< float >` Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = `std::array< float, 10 >`

### Static Public Attributes

- static const [array](#) `denom`

#### 6.36.1 Detailed Description

```
template<>
struct pade::pade_sqrt_denom< float >
```

Definition at line 1428 of file `matrix_multi_imp.h`.

#### 6.36.2 Member Typedef Documentation

##### 6.36.2.1 `array`

```
using pade::pade_sqrt_denom< float >::array = std::array<float, 10>
```

Definition at line 1430 of file `matrix_multi_imp.h`.

#### 6.36.3 Member Data Documentation

##### 6.36.3.1 `denom`

```
const pade_sqrt_denom< float >::array pade::pade_sqrt_denom< float >::denom [static]
```

##### Initial value:

```
=
{
    1.0,          17.0/4.0,      15.0/2.0,      455.0/64.0,
    1001.0/256.0,  1287.0/1024.0,  231.0/1024.0,  165.0/8192.0,
    45.0/65536,   1.0/262144.0
}
```

Definition at line 1431 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- `glucat/matrix_multi_imp.h`

## 6.37 `pade::pade_sqrt_denom< long double >` Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using `array` = `std::array< long double, 18 >`

### Static Public Attributes

- static const `array` `denom`

#### 6.37.1 Detailed Description

```
template<>
struct pade::pade_sqrt_denom< long double >
```

Definition at line 1455 of file `matrix_multi_imp.h`.

#### 6.37.2 Member Typedef Documentation

##### 6.37.2.1 `array`

```
using pade::pade_sqrt_denom< long double >::array = std::array<long double, 18>
```

Definition at line 1457 of file `matrix_multi_imp.h`.

#### 6.37.3 Member Data Documentation

##### 6.37.3.1 `denom`

```
const pade_sqrt_denom< long double >::array pade::pade_sqrt_denom< long double >::denom [static]
```

##### Initial value:

```
=
{
    1.0L,          33.0L/4.0L,          31.0L,          4495.0L/64.0L,
    27405.0L/256.0L, 118755.0L/1024.0L,  94185.0L/1024.0L,  444015.0L/8192.0L,
    1562275.0L/65536.0L, 2042975.0L/262144.0L, 245157.0L/131072.0L, 676039.0L/2097152.0L,
    323323.0L/8388608.0L, 101745.0L/33554432.0L, 4845.0L/33554432.0L, 969.0L/268435456.0L,
    153.0L/4294967296.0L, 1.0L/17179869184.0L
}
```

Definition at line 1458 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- `glucat/matrix_multi_imp.h`

## 6.38 pade::pade\_sqrt\_denom< qd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< qd\_real, 34 >

### Static Public Attributes

- static const [array](#) [denom](#)

### 6.38.1 Detailed Description

```
template<>
struct pade::pade_sqrt_denom< qd_real >
```

Definition at line 1538 of file matrix\_multi\_imp.h.

### 6.38.2 Member Typedef Documentation

#### 6.38.2.1 array

```
using pade::pade\_sqrt\_denom< qd_real >::array = std::array<qd_real, 34>
```

Definition at line 1540 of file matrix\_multi\_imp.h.

### 6.38.3 Member Data Documentation

## 6.38.3.1 denom

```
const pade_sqrt_denom< qd_real >::array pade::pade_sqrt_denom< qd_real >::denom [static]
```

## Initial value:

```
=
{
    qd_real("1"),
    qd_real("126"),
    qd_real("557845")/qd_real("256"),
    qd_real("12515965")/qd_real("1024"),
    qd_real("1916797311")/qd_real("65536"),
    qd_real("4450881435")/qd_real("131072"),
    qd_real("171503444385")/qd_real("8388608"),
    qd_real("221120793075")/qd_real("33554432"),
    qd_real("4923689695575")/qd_real("4294967296"),
    qd_real("456864812569")/qd_real("4294967296"),
    qd_real("2804116503573")/qd_real("549755813888"),
    qd_real("263012370465")/qd_real("2199023255552"),
    qd_real("17592186044416"),
    qd_real("176848560525")/qd_real("140737488355328"),
    qd_real("562949953421312"),
    qd_real("1450433115")/qd_real("281474976710656"),
    qd_real("4503599627370496"),
    qd_real("118183439")/qd_real("18014398509481984"),
    qd_real("72057594037927936"),
    qd_real("121737")/qd_real("72057594037927936"),
    qd_real("576460752303423488"),
    qd_real("561")/qd_real("18446744073709551616"),
    qd_real("73786976294838206464"),
    qd_real("65")/qd_real("4"),
    qd_real("39711")/qd_real("64"),
    qd_real("5949147")/qd_real("1024"),
    qd_real("170574723")/qd_real("8192"),
    qd_real("8996462475")/qd_real("262144"),
    qd_real("59826782925")/qd_real("2097152"),
    qd_real("420696483235")/qd_real("33554432"),
    qd_real("797168807855")/qd_real("268435456"),
    qd_real("6499270398159")/qd_real("17179869184"),
    qd_real("3486599885395")/qd_real("137438953472"),
    qd_real("1886827875075")/qd_real("2199023255552"),
    qd_real("240141729555")/qd_real("51538723353"),
    qd_real("977699359")/qd_real("9652005"),
    qd_real("6545")/qd_real("1"),
    qd_real("1")/qd_real("1")
}
```

Definition at line 1541 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.39 pade::pade\_sqrt\_numer&lt; Scalar\_T &gt; Struct Template Reference

Coefficients of numerator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)

```
#include <matrix_multi_imp.h>
```

## Public Types

- using [array](#) = std::array< Scalar\_T, 14 >

## Static Public Attributes

- static const [array](#) [numer](#)

### 6.39.1 Detailed Description

```
template<typename Scalar_T>
struct pade::pade_sqrt_numer< Scalar_T >
```

Coefficients of numerator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)

Definition at line 1384 of file matrix\_multi\_imp.h.

### 6.39.2 Member Typedef Documentation

#### 6.39.2.1 array

```
template<typename Scalar_T>
using pade::pade_sqrt_numer< Scalar_T >::array = std::array<Scalar_T, 14>
```

Definition at line 1386 of file matrix\_multi\_imp.h.

### 6.39.3 Member Data Documentation

#### 6.39.3.1 numer

```
template<typename Scalar_T>
const pade_sqrt_numer< Scalar_T >::array pade::pade_sqrt_numer< Scalar_T >::numer [static]
```

**Initial value:**

```
=
{
    1.0,                27.0/4.0,            81.0/4.0,            2277.0/64.0,
    10395.0/256.0,       32319.0/1024.0,       8721.0/512.0,       26163.0/4096.0,
    53703.0/32768.0,     36465.0/131072.0,     3861.0/131072.0,     7371.0/4194304.0,
    819.0/16777216.0,    27.0/67108864.0
}
```

Definition at line 1387 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- [glucat/matrix\\_multi\\_imp.h](#)

## 6.40 pade::pade\_sqrt\_numer< dd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

## Public Types

- using [array](#) = std::array< dd\_real, 22 >

## Static Public Attributes

- static const [array](#) [number](#)

### 6.40.1 Detailed Description

```
template<>
struct pade::pade_sqrt_number< dd_real >
```

Definition at line 1471 of file matrix\_multi\_imp.h.

### 6.40.2 Member Typedef Documentation

#### 6.40.2.1 array

```
using pade::pade_sqrt_number< dd_real >::array = std::array<dd_real, 22>
```

Definition at line 1473 of file matrix\_multi\_imp.h.

### 6.40.3 Member Data Documentation

#### 6.40.3.1 number

```
const pade_sqrt_number< dd_real >::array pade::pade_sqrt_number< dd_real >::number [static]
```

**Initial value:**

```
=
{
    dd_real("1"),
    dd_real("215")/dd_real("4"),
    dd_real("90687")/dd_real("256"),
    dd_real("168861")/dd_real("256"),
    dd_real("7228859")/dd_real("16384"),
    dd_real("7538115")/dd_real("65536"),
    dd_real("195747825")/dd_real("16777216"),
    dd_real("14375115")/dd_real("33554432"),
    dd_real("20764055")/dd_real("4294967296"),
    dd_real("206701")/dd_real("17179869184"),
    dd_real("3311")/dd_real("1099511627776"),
    dd_real("43")/dd_real("4"),
    dd_real("10621")/dd_real("64"),
    dd_real("567987")/dd_real("1024"),
    dd_real("1246355")/dd_real("2048"),
    dd_real("16583853")/dd_real("65536"),
    dd_real("173376645")/dd_real("4194304"),
    dd_real("171655785")/dd_real("67108864"),
    dd_real("14375115")/dd_real("268435456"),
    dd_real("5167525")/dd_real("17179869184"),
    dd_real("76153")/dd_real("274877906944"),
    dd_real("43")/dd_real("4398046511104")
}
```

Definition at line 1474 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)

## 6.41 pade::pade\_sqrt\_numer< float > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using [array](#) = std::array< float, 10 >

### Static Public Attributes

- static const [array](#) [numer](#)

#### 6.41.1 Detailed Description

```
template<>
struct pade::pade_sqrt_numer< float >
```

Definition at line 1416 of file matrix\_multi\_imp.h.

#### 6.41.2 Member Typedef Documentation

##### 6.41.2.1 array

```
using pade::pade_sqrt_numer< float >::array = std::array<float, 10>
```

Definition at line 1418 of file matrix\_multi\_imp.h.

#### 6.41.3 Member Data Documentation

##### 6.41.3.1 numer

```
const pade_sqrt_numer< float >::array pade::pade_sqrt_numer< float >::numer [static]
```

##### Initial value:

```
=
{
    1.0,          19.0/4.0,      19.0/2.0,      665.0/64.0,
    1729.0/256.0,  2717.0/1024.0,  627.0/1024.0,  627.0/8192.0,
    285.0/65536.0, 19.0/262144.0
}
```

Definition at line 1419 of file matrix\_multi\_imp.h.

The documentation for this struct was generated from the following file:

- glucat/[matrix\\_multi\\_imp.h](#)



## 6.42 `pade::pade_sqrt_numer< long double >` Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using `array` = `std::array< long double, 18 >`

### Static Public Attributes

- static const `array number`

### 6.42.1 Detailed Description

```
template<>
struct pade::pade_sqrt_numer< long double >
```

Definition at line 1441 of file `matrix_multi_imp.h`.

### 6.42.2 Member Typedef Documentation

#### 6.42.2.1 `array`

```
using pade::pade_sqrt_numer< long double >::array = std::array<long double, 18>
```

Definition at line 1443 of file `matrix_multi_imp.h`.

### 6.42.3 Member Data Documentation

#### 6.42.3.1 `number`

```
const pade_sqrt_numer< long double >::array pade::pade_sqrt_numer< long double >::number [static]
```

#### Initial value:

```
=
{
    1.0L,                35.0L/4.0L,                35.0L,                5425.0L/64.0L,
    35525.0L/256.0L,     166257.0L/1024.0L,          143325.0L/1024.0L,          740025.0L/8192.0L,
    2877875.0L/65536.0L,  4206125.0L/262144.0L,          572033.0L/131072.0L,          1820105.0L/2097152.0L,
    1028755.0L/8388608.0L, 395675.0L/33554432.0L,          24225.0L/33554432.0L,          6783.0L/268435456.0L,
    1785.0L/4294967296.0L, 35.0L/17179869184.0L
}
```

Definition at line 1444 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- `glucat/matrix_multi_imp.h`

## 6.43 pade::pade\_sqrt\_numer< qd\_real > Struct Template Reference

```
#include <matrix_multi_imp.h>
```

### Public Types

- using `array` = `std::array< qd_real, 34 >`

### Static Public Attributes

- static const `array numer`

#### 6.43.1 Detailed Description

```
template<>  
struct pade::pade_sqrt_numer< qd_real >
```

Definition at line 1512 of file `matrix_multi_imp.h`.

#### 6.43.2 Member Typedef Documentation

##### 6.43.2.1 array

```
using pade::pade_sqrt_numer< qd_real >::array = std::array<qd_real, 34>
```

Definition at line 1514 of file `matrix_multi_imp.h`.

#### 6.43.3 Member Data Documentation

6.43.3.1 `numer`

```
const pade_sqrt_numer< qd_real >::array pade::pade_sqrt_numer< qd_real >::numer [static]
```

## Initial value:

```
=
{
    qd_real("1"),
    qd_real("134"),
    qd_real("633485")/qd_real("256"),
    qd_real("15246721")/qd_real("1024"),
    qd_real("2518145487")/qd_real("65536"),
    qd_real("6344873535")/qd_real("131072"),
    qd_real("267226297065")/qd_real("8388608"),
    qd_real("379874182975")/qd_real("33554432"),
    qd_real("9425348845815")/qd_real("4294967296"),
    qd_real("987417498133")/qd_real("4294967296"),
    qd_real("6958363175533")/qd_real("549755813888"),
    qd_real("766166470485")/qd_real("219902325552"),
    qd_real("623623871325")/qd_real("140737488355328"),
    qd_real("6478601247")/qd_real("281474976710656"),
    qd_real("719844583")/qd_real("18014398509481984"),
    qd_real("1165197")/qd_real("72057594037927936"),
    qd_real("72057594037927936"),
    qd_real("12529")/qd_real("18446744073709551616"),
    qd_real("67")/qd_real("4"),
    qd_real("43617")/qd_real("64"),
    qd_real("6992857")/qd_real("1024"),
    qd_real("215632197")/qd_real("8192"),
    qd_real("12301285425")/qd_real("262144"),
    qd_real("89075432355")/qd_real("2097152"),
    qd_real("687479618945")/qd_real("33554432"),
    qd_real("1443521895305")/qd_real("268435456"),
    qd_real("13195488384141")/qd_real("17179869184"),
    qd_real("8055248011085")/qd_real("137438953472"),
    qd_real("5056698705201")/qd_real("2199023255552"),
    qd_real("766166470485")/qd_real("219902325552"),
    qd_real("203123203803")/qd_real("5038912081"),
    qd_real("71853815")/qd_real("87703"),
    qd_real("67")/qd_real("4")
}
```

Definition at line 1515 of file `matrix_multi_imp.h`.

The documentation for this struct was generated from the following file:

- `glucat/matrix_multi_imp.h`

6.44 `glucat::numeric_traits< Scalar_T >::promoted<>` Struct Template Reference

Extra traits which extend numeric limits.

```
#include <promotion.h>
```

## Public Types

- using `type` = double
- using `type` = long double
- using `type` = double

### 6.44.1 Detailed Description

```
template<typename Scalar_T>
template<>
struct glucat::numeric_traits< Scalar_T >::promoted<>
```

Extra traits which extend numeric limits.

Promoted type.

Promoted type for long double.

Promoted type for double

Definition at line 70 of file promotion.h.

### 6.44.2 Member Typedef Documentation

#### 6.44.2.1 type [1/3]

```
template<typename Scalar_T >
using glucat::numeric_traits< Scalar_T >::promoted<>::type = double
```

Definition at line 72 of file promotion.h.

#### 6.44.2.2 type [2/3]

```
template<typename Scalar_T >
using glucat::numeric_traits< Scalar_T >::promoted<>::type = long double
```

Definition at line 86 of file promotion.h.

#### 6.44.2.3 type [3/3]

```
template<typename Scalar_T >
using glucat::numeric_traits< Scalar_T >::promoted<>::type = double
```

Definition at line 145 of file scalar.h.

The documentation for this struct was generated from the following files:

- [glucat/promotion.h](#)
- [glucat/scalar.h](#)

## 6.45 glucat::random\_generator< Scalar\_T > Class Template Reference

Random number generator with single instance per Scalar\_T.

```
#include <random.h>
```

### Public Member Functions

- [random\\_generator](#) (const [random\\_generator](#) &)=delete
- auto [operator=](#) (const [random\\_generator](#) &) -> [random\\_generator](#) &=delete
- auto [uniform](#) () -> Scalar\_T
- auto [normal](#) () -> Scalar\_T

### Static Public Member Functions

- static auto [generator](#) () -> [random\\_generator](#) &  
*Single instance of Random number generator.*

### Private Member Functions

- [random\\_generator](#) ()
- [~random\\_generator](#) ()=default

### Private Attributes

- std::mt19937 [uint\\_gen](#)
- std::uniform\_real\_distribution< double > [uniform\\_dist](#)
- std::normal\_distribution< double > [normal\\_dist](#)

### Static Private Attributes

- static const unsigned long [seed](#) = 19590921UL

### Friends

- class [friend\\_for\\_private\\_destructor](#)

#### 6.45.1 Detailed Description

```
template<typename Scalar_T>  
class glucat::random_generator< Scalar_T >
```

Random number generator with single instance per Scalar\_T.

Definition at line 42 of file random.h.

## 6.45.2 Constructor & Destructor Documentation

### 6.45.2.1 random\_generator() [1/2]

```
template<typename Scalar_T >
glucat::random_generator< Scalar_T >::random_generator (
    const random_generator< Scalar_T > & ) [delete]
```

### 6.45.2.2 random\_generator() [2/2]

```
template<typename Scalar_T >
glucat::random_generator< Scalar_T >::random_generator ( ) [inline], [private]
```

Definition at line 61 of file random.h.

References glucat::random\_generator< Scalar\_T >::seed.

### 6.45.2.3 ~random\_generator()

```
template<typename Scalar_T >
glucat::random_generator< Scalar_T >::~~random_generator ( ) [private], [default]
```

## 6.45.3 Member Function Documentation

### 6.45.3.1 generator()

```
template<typename Scalar_T >
static auto glucat::random_generator< Scalar_T >::generator ( ) -> random_generator& [inline],
[static]
```

Single instance of Random number generator.

Definition at line 51 of file random.h.

#### 6.45.3.2 normal()

```
template<typename Scalar_T >
auto glucat::random_generator< Scalar_T >::normal ( ) -> Scalar_T    [inline]
```

Definition at line 70 of file random.h.

References glucat::random\_generator< Scalar\_T >::normal\_dist.

#### 6.45.3.3 operator=()

```
template<typename Scalar_T >
auto glucat::random_generator< Scalar_T >::operator= (
    const random_generator< Scalar_T > & ) -> random_generator &=delete    [delete]
```

#### 6.45.3.4 uniform()

```
template<typename Scalar_T >
auto glucat::random_generator< Scalar_T >::uniform ( ) -> Scalar_T    [inline]
```

Definition at line 68 of file random.h.

References glucat::random\_generator< Scalar\_T >::uniform\_dist.

### 6.45.4 Friends And Related Function Documentation

#### 6.45.4.1 friend\_for\_private\_destructor

```
template<typename Scalar_T >
friend class friend_for_private_destructor    [friend]
```

Friend declaration to avoid compiler warning: "... only defines a private destructor and has no friends" Ref: Carlos O'Ryan, ACE <http://doc.ece.uci.edu>

Definition at line 48 of file random.h.

### 6.45.5 Member Data Documentation

#### 6.45.5.1 normal\_dist

```
template<typename Scalar_T >
std::normal_distribution<double> glucat::random\_generator< Scalar_T >::normal_dist [private]
```

Definition at line 59 of file random.h.

Referenced by [glucat::random\\_generator](#)< Scalar\_T >::normal().

#### 6.45.5.2 seed

```
template<typename Scalar_T >
const unsigned long glucat::random\_generator< Scalar_T >::seed = 19590921UL [static], [private]
```

Definition at line 55 of file random.h.

Referenced by [glucat::random\\_generator](#)< Scalar\_T >::random\_generator().

#### 6.45.5.3 uint\_gen

```
template<typename Scalar_T >
std::mt19937 glucat::random\_generator< Scalar_T >::uint_gen [private]
```

Definition at line 57 of file random.h.

#### 6.45.5.4 uniform\_dist

```
template<typename Scalar_T >
std::uniform_real_distribution<double> glucat::random\_generator< Scalar_T >::uniform_dist
[private]
```

Definition at line 58 of file random.h.

Referenced by [glucat::random\\_generator](#)< Scalar\_T >::uniform().

The documentation for this class was generated from the following file:

- [glucat/random.h](#)

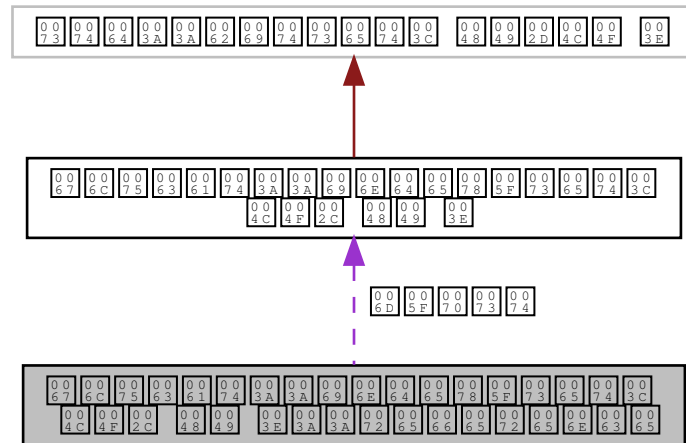


## 6.46 glucat::index\_set< LO, HI >::reference Class Reference

Index set member reference.

```
#include <index_set.h>
```

Collaboration diagram for glucat::index\_set< LO, HI >::reference:



### Public Member Functions

- [reference](#) ()=delete  
*Default constructor is deleted.*
- [reference](#) (index\_set\_t &ist, index\_t idx)  
*index\_set reference*
- [~reference](#) ()=default
- auto [operator==](#) (const [reference](#) &c\_j) const -> bool  
*for b[i] == c[j];*
- auto [operator=](#) (const bool x) -> [reference](#) &  
*for b[i] = x;*
- auto [operator=](#) (const [reference](#) &c\_j) -> [reference](#) &  
*for b[i] = c[j];*
- auto [operator~](#) () const -> bool  
*Flips a bit.*
- [operator bool](#) () const  
*for x = b[i];*
- auto [flip](#) () -> [reference](#) &  
*for b[i].flip();*

### Private Attributes

- [index\\_set\\_t](#) \* m\_pst
- [index\\_t](#) m\_idx

## Friends

- class [index\\_set](#)

### 6.46.1 Detailed Description

```
template<const index_t LO, const index_t HI>  
class glucat::index_set< LO, HI >::reference
```

Index set member reference.

Definition at line 177 of file `index_set.h`.

### 6.46.2 Constructor & Destructor Documentation

#### 6.46.2.1 `reference()` [1/2]

```
template<const index_t LO, const index_t HI>  
glucat::index_set< LO, HI >::reference::reference ( ) [delete]
```

Default constructor is deleted.

#### 6.46.2.2 `reference()` [2/2]

```
template<const index_t LO, const index_t HI>  
glucat::index_set< LO, HI >::reference::reference (   
    index_set_t & ist,  
    index_t idx ) [inline]
```

[index\\_set](#) reference

Definition at line 986 of file `index_set_imp.h`.

#### 6.46.2.3 `~reference()`

```
template<const index_t LO, const index_t HI>  
glucat::index_set< LO, HI >::reference::~reference ( ) [default]
```

### 6.46.3 Member Function Documentation

#### 6.46.3.1 flip()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reference::flip ( ) -> reference& [inline]
```

for b[i].flip();

Definition at line 1050 of file index\_set\_imp.h.

References glucat::index\_set< LO, HI >::reference::flip().

Referenced by glucat::index\_set< LO, HI >::reference::flip().

#### 6.46.3.2 operator bool()

```
template<const index_t LO, const index_t HI>
glucat::index_set< LO, HI >::reference::operator bool ( ) const [inline]
```

for x = b[i];

Definition at line 1042 of file index\_set\_imp.h.

#### 6.46.3.3 operator=() [1/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reference::operator= (
    const bool x ) -> reference& [inline]
```

for b[i] = x;

Definition at line 1004 of file index\_set\_imp.h.

#### 6.46.3.4 operator=() [2/2]

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reference::operator= (
    const reference & c_j ) -> reference& [inline]
```

for b[i] = c[j];

Definition at line 1018 of file index\_set\_imp.h.

#### 6.46.3.5 operator==()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reference::operator== (
    const reference & c_j ) const -> bool [inline]
```

for b[i] == c[j];

Definition at line 996 of file index\_set\_imp.h.

#### 6.46.3.6 operator~()

```
template<const index_t LO, const index_t HI>
auto glucat::index_set< LO, HI >::reference::operator~ ( ) const -> bool [inline]
```

Flips a bit.

flips the bit

Definition at line 1035 of file index\_set\_imp.h.

### 6.46.4 Friends And Related Function Documentation

#### 6.46.4.1 index\_set

```
template<const index_t LO, const index_t HI>
friend class index_set [friend]
```

Definition at line 178 of file index\_set.h.

### 6.46.5 Member Data Documentation

#### 6.46.5.1 m\_idx

```
template<const index_t LO, const index_t HI>
index_t glucat::index_set< LO, HI >::reference::m_idx [private]
```

Definition at line 200 of file index\_set.h.

## 6.46.5.2 m\_pst

```
template<const index_t LO, const index_t HI>
index_set_t* glucat::index_set< LO, HI >::reference::m_pst [private]
```

Definition at line 199 of file index\_set.h.

The documentation for this class was generated from the following files:

- [glucat/index\\_set.h](#)
- [glucat/index\\_set\\_imp.h](#)

## 6.47 glucat::sorted\_range&lt; Map\_T, Sorted\_Map\_T &gt; Class Template Reference

Sorted range for use with output.

```
#include <framed_multi_imp.h>
```

## Public Types

- using [map\\_t](#) = Map\_T
- using [sorted\\_map\\_t](#) = Sorted\_Map\_T
- using [sorted\\_iterator](#) = typename Sorted\_Map\_T::const\_iterator

## Public Member Functions

- [sorted\\_range](#) (Sorted\_Map\_T &sorted\_val, const Map\_T &val)

## Public Attributes

- [sorted\\_iterator](#) sorted\_begin
- [sorted\\_iterator](#) sorted\_end

## 6.47.1 Detailed Description

```
template<typename Map_T, typename Sorted_Map_T>
class glucat::sorted_range< Map_T, Sorted_Map_T >
```

Sorted range for use with output.

Definition at line 1112 of file framed\_multi\_imp.h.

## 6.47.2 Member Typedef Documentation

#### 6.47.2.1 map\_t

```
template<typename Map_T, typename Sorted_Map_T>
using glucat::sorted_range< Map_T, Sorted_Map_T >::map_t = Map_T
```

Definition at line 1115 of file framed\_multi\_imp.h.

#### 6.47.2.2 sorted\_iterator

```
template<typename Map_T, typename Sorted_Map_T>
using glucat::sorted_range< Map_T, Sorted_Map_T >::sorted_iterator = typename Sorted_Map_T↔
::const_iterator
```

Definition at line 1117 of file framed\_multi\_imp.h.

#### 6.47.2.3 sorted\_map\_t

```
template<typename Map_T, typename Sorted_Map_T>
using glucat::sorted_range< Map_T, Sorted_Map_T >::sorted_map_t = Sorted_Map_T
```

Definition at line 1116 of file framed\_multi\_imp.h.

### 6.47.3 Constructor & Destructor Documentation

#### 6.47.3.1 sorted\_range()

```
template<typename Map_T, typename Sorted_Map_T>
glucat::sorted_range< Map_T, Sorted_Map_T >::sorted_range (
    Sorted_Map_T & sorted_val,
    const Map_T & val ) [inline]
```

Definition at line 1119 of file framed\_multi\_imp.h.

References glucat::sorted\_range< Map\_T, Sorted\_Map\_T >::sorted\_begin, and glucat::sorted\_range< Map\_T, Sorted\_Map\_T >::sorted\_end.

### 6.47.4 Member Data Documentation

## 6.47.4.1 sorted\_begin

```
template<typename Map_T, typename Sorted_Map_T>
sorted_iterator glucat::sorted_range< Map_T, Sorted_Map_T >::sorted_begin
```

Definition at line 1126 of file framed\_multi\_imp.h.

Referenced by glucat::operator<<(), and glucat::sorted\_range< Map\_T, Sorted\_Map\_T >::sorted\_range().

## 6.47.4.2 sorted\_end

```
template<typename Map_T, typename Sorted_Map_T>
sorted_iterator glucat::sorted_range< Map_T, Sorted_Map_T >::sorted_end
```

Definition at line 1127 of file framed\_multi\_imp.h.

Referenced by glucat::sorted\_range< Map\_T, Sorted\_Map\_T >::sorted\_range().

The documentation for this class was generated from the following file:

- glucat/framed\_multi\_imp.h

## 6.48 glucat::sorted\_range&lt; Sorted\_Map\_T, Sorted\_Map\_T &gt; Class Template Reference

```
#include <framed_multi_imp.h>
```

## Public Types

- using [map\\_t](#) = Sorted\_Map\_T
- using [sorted\\_map\\_t](#) = Sorted\_Map\_T
- using [sorted\\_iterator](#) = typename Sorted\_Map\_T::const\_iterator

## Public Member Functions

- [sorted\\_range](#) (Sorted\_Map\_T &sorted\_val, const Sorted\_Map\_T &val)

## Public Attributes

- [sorted\\_iterator](#) [sorted\\_begin](#)
- [sorted\\_iterator](#) [sorted\\_end](#)

## 6.48.1 Detailed Description

```
template<typename Sorted_Map_T>
class glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >
```

Definition at line 1131 of file framed\_multi\_imp.h.

## 6.48.2 Member Typedef Documentation

### 6.48.2.1 map\_t

```
template<typename Sorted_Map_T >
using glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >::map_t = Sorted_Map_T
```

Definition at line 1134 of file framed\_multi\_imp.h.

### 6.48.2.2 sorted\_iterator

```
template<typename Sorted_Map_T >
using glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >::sorted_iterator = typename Sorted_Map_T::const_iterator
```

Definition at line 1136 of file framed\_multi\_imp.h.

### 6.48.2.3 sorted\_map\_t

```
template<typename Sorted_Map_T >
using glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >::sorted_map_t = Sorted_Map_T
```

Definition at line 1135 of file framed\_multi\_imp.h.

## 6.48.3 Constructor & Destructor Documentation

### 6.48.3.1 sorted\_range()

```
template<typename Sorted_Map_T >
glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >::sorted_range (
    Sorted_Map_T & sorted_val,
    const Sorted_Map_T & val ) [inline]
```

Definition at line 1138 of file framed\_multi\_imp.h.

## 6.48.4 Member Data Documentation



## 6.48.4.1 sorted\_begin

```
template<typename Sorted_Map_T >
sorted_iterator glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >::sorted_begin
```

Definition at line 1142 of file framed\_multi\_imp.h.

## 6.48.4.2 sorted\_end

```
template<typename Sorted_Map_T >
sorted_iterator glucat::sorted_range< Sorted_Map_T, Sorted_Map_T >::sorted_end
```

Definition at line 1143 of file framed\_multi\_imp.h.

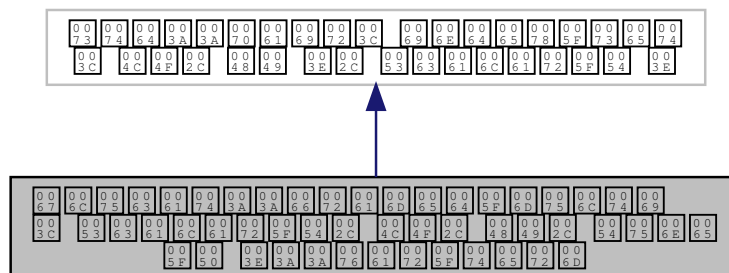
The documentation for this class was generated from the following file:

- [glucat/framed\\_multi\\_imp.h](#)

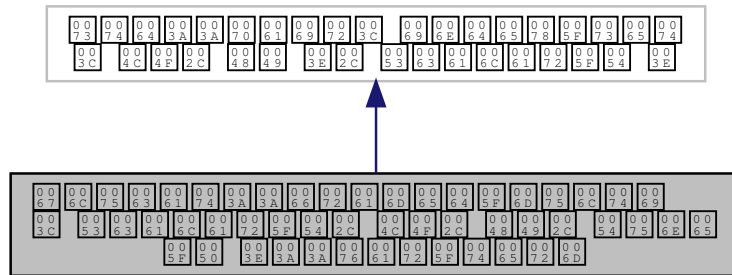
## 6.49 glucat::framed\_multi&lt; Scalar\_T, LO, HI, Tune\_P &gt;::var\_term Class Reference

Variable term.

Inheritance diagram for glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >::var\_term:



Collaboration diagram for `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term`:



## Public Types

- using `var_pair_t` = `std::pair< index_set< LO, HI >, Scalar_T >`

## Public Member Functions

- `~var_term()` = default  
*Destructor.*
- `var_term()`  
*Default constructor.*
- `var_term(const index_set_t ist, const Scalar_T &crd=Scalar_T(1))`  
*Construct a variable term from an index set and a scalar coordinate.*
- `auto operator*= (const term_t &rhs) -> var_term_t &`  
*Product of variable term and term.*

## Static Public Member Functions

- `static auto classname() -> const std::string`  
*Class name used in messages.*

### 6.49.1 Detailed Description

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P =
tuning<>>
class glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term
```

Variable term.

Definition at line 279 of file `framed_multi.h`.

## 6.49.2 Member Typedef Documentation

### 6.49.2.1 `var_pair_t`

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
using glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term::var_pair_t = std::pair<index_set<LO, HI>, Scalar_T>
```

Definition at line 283 of file `framed_multi.h`.

## 6.49.3 Constructor & Destructor Documentation

### 6.49.3.1 `~var_term()`

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term::~~var_term ( ) [default]
```

Destructor.

### 6.49.3.2 `var_term()` [1/2]

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term::var_term ( ) [inline]
```

Default constructor.

Definition at line 291 of file `framed_multi.h`.

### 6.49.3.3 `var_term()` [2/2]

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::var_term::var_term (
    const index_set_t ist,
    const Scalar_T & crd = Scalar_T(1) ) [inline]
```

Construct a variable term from an index set and a scalar coordinate.

Definition at line 295 of file `framed_multi.h`.

## 6.49.4 Member Function Documentation

### 6.49.4.1 `classname()`

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
static auto glucat::framed\_multi< Scalar_T, LO, HI, Tune_P >::var_term::classname ( ) ->
const std::string [inline], [static]
```

Class name used in messages.

Definition at line 286 of file `framed_multi.h`.

### 6.49.4.2 `operator*=( )`

```
template<typename Scalar_T = double, const index_t LO = DEFAULT_LO, const index_t HI = DEFAULT_HI, typename Tune_P = tuning<>>
auto glucat::framed\_multi< Scalar_T, LO, HI, Tune_P >::var_term::operator*= (
    const term\_t & rhs ) -> var\_term\_t [inline]
```

Product of variable term and term.

Definition at line 299 of file `framed_multi.h`.

References `PyClical::rhs`.

The documentation for this class was generated from the following file:

- [glucat/framed\\_multi.h](#)

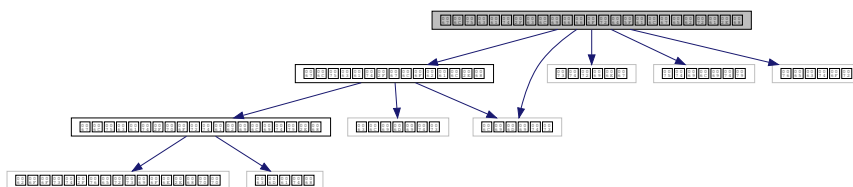
## Chapter 7

# File Documentation

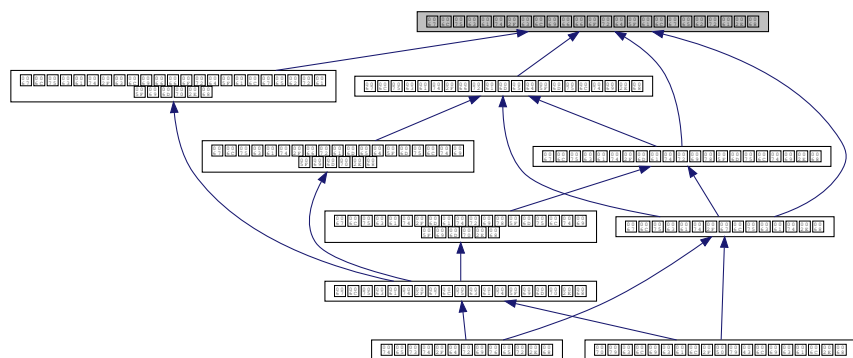
### 7.1 glucat/clifford\_algebra.h File Reference

```
#include "glucat/global.h"  
#include <limits>  
#include <string>  
#include <utility>  
#include <vector>
```

Include dependency graph for clifford\_algebra.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::clifford\\_algebra< Scalar\\_T, Index\\_Set\\_T, Multivector\\_T >](#)  
*clifford\_algebra<> declares the operations of a Clifford algebra*

## Namespaces

- [glucat](#)

## Macros

- [#define \\_GLUCAT\\_CLIFFORD\\_ALGEBRA\\_OPERATIONS](#)

## Functions

- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, template< typename, const index\\_t, const index\\_t, typename > class RHS, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::operator!=](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &lhs, const RHS< Scalar\_T, LO, HI, Tune\_P > &rhs) -> bool  
*Test for inequality of multivectors.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::operator!=](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &lhs, const Scalar\_T &scr) -> bool  
*Test for inequality of multivector and scalar.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::operator!=](#) (const Scalar\_T &scr, const Multivector< Scalar\_T, LO, HI, Tune\_P > &rhs) -> bool  
*Test for inequality of scalar and multivector.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::error\\_squared\\_tol](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &val) -> Scalar\_T  
*Quadratic norm error tolerance relative to a specific multivector.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, template< typename, const index\\_t, const index\\_t, typename > class RHS, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::error\\_squared](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &lhs, const RHS< Scalar\_T, LO, HI, Tune\_P > &rhs, const Scalar\_T threshold) -> Scalar\_T  
*Relative or absolute error using the quadratic norm.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, template< typename, const index\\_t, const index\\_t, typename > class RHS, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::approx\\_equal](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &lhs, const RHS< Scalar\_T, LO, HI, Tune\_P > &rhs, const Scalar\_T threshold, const Scalar\_T tolerance) -> bool  
*Test for approximate equality of multivectors.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, template< typename, const index\\_t, const index\\_t, typename > class RHS, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::approx\\_equal](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &lhs, const RHS< Scalar\_T, LO, HI, Tune\_P > &rhs) -> bool  
*Test for approximate equality of multivectors.*
- [template<template< typename, const index\\_t, const index\\_t, typename > class Multivector, typename Scalar\\_T , const index\\_t LO, const index\\_t HI, typename Tune\\_P >](#)  
[auto glucat::operator+](#) (const Multivector< Scalar\_T, LO, HI, Tune\_P > &lhs, const Scalar\_T &scr) -> const Multivector< Scalar\_T, LO, HI, Tune\_P >  
*Geometric sum of multivector and scalar.*

- Generated by Doxygen

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::star (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> Scalar_T`  
*Hestenes scalar product.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator/ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const Scalar_T &scr) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Quotient of multivector and scalar.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator/ (const Scalar_T &scr, const Multivector< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Quotient of scalar and multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator/ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Geometric quotient.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator| (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Transformation via twisted adjoint action.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::inv (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Geometric multiplicative inverse.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, int rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Integer power of multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Multivector power of multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::outer_pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, int rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Outer product power of multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::scalar (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Scalar part.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::real (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Real part: synonym for scalar part.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::imag (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`



*Imaginary part: deprecated (always 0)*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::pure (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Pure part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::even (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Even part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::odd (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Odd part.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::vector_part (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const std::vector< Scalar_T >`

*Vector part of multivector, as a vector\_t with respect to frame()*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::involute (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Main involution, each {i} is replaced by -{i} in each term, eg. {1}\*{2} -> (-{2})\*(-{1})*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::reverse (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Reversion, eg. {1}\*{2} -> {2}\*{1}.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::conj (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Conjugation, rev o invo == invo o rev.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::quad (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Scalar\_T quadratic form == (rev(x)\*x)(0)*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::norm (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Scalar\_T norm == sum of norm of coordinates.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::abs (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Absolute value == sqrt(norm)*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::max_abs (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Maximum of absolute values of components of multivector: multivector infinity norm.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::complexifier (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of -1 which commutes with all members of the frame of the given multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::elliptic (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::sqrt (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::sqrt (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::clifford\_exp (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Exponential of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::cos (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Cosine of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::cos (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::acos (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse cosine of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::acos (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::cosh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Hyperbolic cosine of multivector.*

- Generated by Doxygen

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atan (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inverse tangent of multivector with specified complexifier.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atan (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inverse tangent of multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::tanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Hyperbolic tangent of multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inverse hyperbolic tangent of multivector with specified complexifier.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Inverse hyperbolic tangent of multivector.*

## 7.1.1 Macro Definition Documentation

### 7.1.1.1 \_GLUCAT\_CLIFFORD\_ALGEBRA\_OPERATIONS

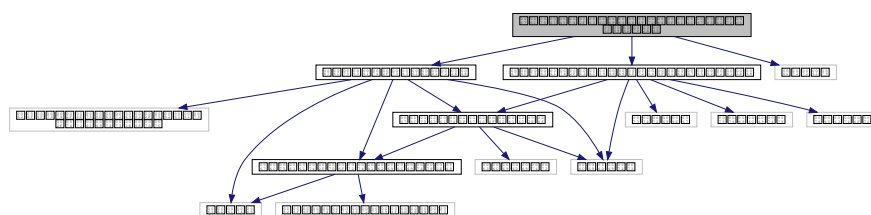
```
#define _GLUCAT_CLIFFORD_ALGEBRA_OPERATIONS
```

Definition at line 145 of file `clifford_algebra.h`.

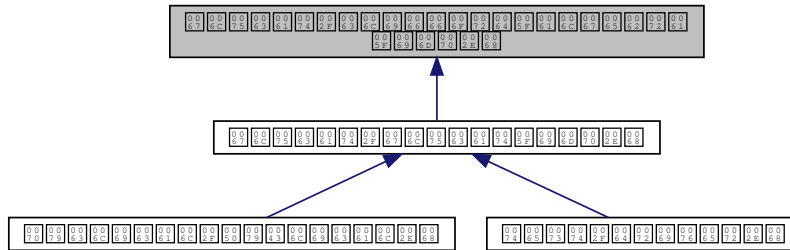
## 7.2 `glucat/clifford_algebra_imp.h` File Reference

```
#include "glucat/clifford_algebra.h"
#include "glucat/scalar.h"
#include <array>
```

Include dependency graph for `clifford_algebra_imp.h`:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [glucat](#)

## Functions

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator!= (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> bool`  
*Test for inequality of multivectors.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator!= (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const Scalar_T &scr) -> bool`  
*Test for inequality of multivector and scalar.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator!= (const Scalar_T &scr, const Multivector< Scalar_T, LO, HI, Tune_P > &rhs) -> bool`  
*Test for inequality of scalar and multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::error\_squared\_tol (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Quadratic norm error tolerance relative to a specific multivector.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::error\_squared (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs, const Scalar_T threshold) -> Scalar_T`  
*Relative or absolute error using the quadratic norm.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::approx\_equal (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs, const Scalar_T threshold, const Scalar_T tolerance) -> bool`  
*Test for approximate equality of multivectors.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::approx\_equal (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> bool`



*Inner product.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::operator% (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Left contraction.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::star (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> Scalar_T`

*Hestenes scalar product.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::operator/ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const Scalar_T &scr) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Quotient of multivector and scalar.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::operator/ (const Scalar_T &scr, const Multivector< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Quotient of scalar and multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::operator/ (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Geometric quotient.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::operator| (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Transformation via twisted adjoint action.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::inv (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Geometric multiplicative inverse.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, int rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Integer power of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, template< typename, const index_t, const index_t, typename > class RHS, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, const RHS< Scalar_T, LO, HI, Tune_P > &rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Multivector power of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::outer_pow (const Multivector< Scalar_T, LO, HI, Tune_P > &lhs, int rhs) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Outer product power of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P > auto glucat::scalar (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Scalar part.*



- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::real (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Real part: synonym for scalar part.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::imag (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Imaginary part: deprecated (always 0)*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::pure (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Pure part.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::even (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Even part.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::odd (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Odd part.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::vector\_part (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const std::vector< Scalar_T, LO, HI, Tune_P >`  
*Vector part of multivector, as a vector\_t with respect to frame()*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::involute (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Main involution, each {i} is replaced by -{i} in each term, eg. {1}\*{2} -> (-{2})\*(-{1})*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::reverse (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Reversion, eg. {1}\*{2} -> {2}\*{1}.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::conj (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`  
*Conjugation, rev o invo == invo o rev.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::quad (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Scalar\_T quadratic form == (rev(x)\*x)(0)*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::norm (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Scalar\_T norm == sum of norm of coordinates.*
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::abs (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`  
*Absolute value == sqrt(norm)*



- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::max_abs (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> Scalar_T`

*Maximum of absolute values of components of multivector: multivector infinity norm.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::complexifier (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of -1 which commutes with all members of the frame of the given multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::elliptic (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static void glucat::check_complex (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false)`

*Check that i is a valid complexifier for val.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::sqrt (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::sqrt (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::clifford_exp (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Exponential of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::cosh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Hyperbolic cosine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::acosh (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse hyperbolic cosine of multivector with specified complexifier.*



- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::asin (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse sine of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::tanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Hyperbolic tangent of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse hyperbolic tangent of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atanh (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse hyperbolic tangent of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::tan (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Tangent of multivector with specified complexifier.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::tan (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Tangent of multivector.*

- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atan (const Multivector< Scalar_T, LO, HI, Tune_P > &val, const Multivector< Scalar_T, LO, HI, Tune_P > &i, const bool prechecked=false) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse tangent of multivector with specified complexifier.*

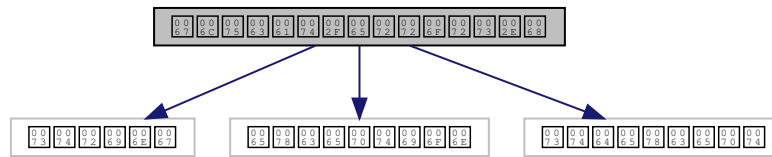
- `template<template< typename, const index_t, const index_t, typename > class Multivector, typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::atan (const Multivector< Scalar_T, LO, HI, Tune_P > &val) -> const Multivector< Scalar_T, LO, HI, Tune_P >`

*Inverse tangent of multivector.*

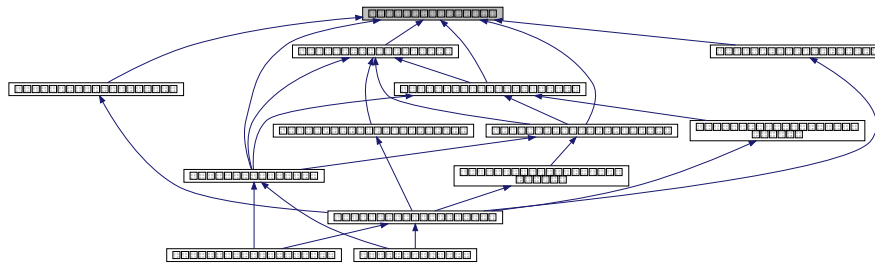
## 7.3 glucat/errors.h File Reference

```
#include <string>
#include <exception>
#include <stdexcept>
```

Include dependency graph for errors.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::glucat\\_error](#)  
*Abstract exception class.*
- class [glucat::error< Class\\_T >](#)  
*Specific exception class.*

## Namespaces

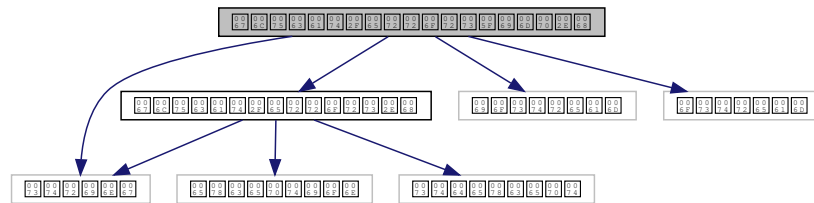
- [glucat](#)

## 7.4 glucat/errors\_imp.h File Reference

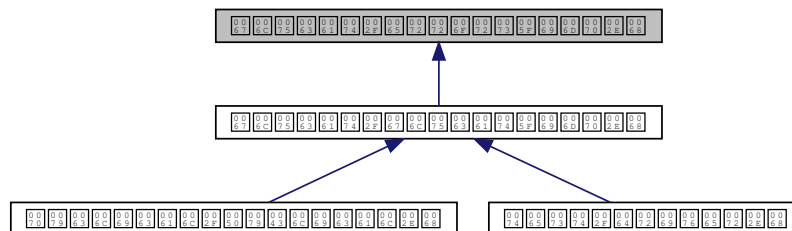
```
#include "glucat/errors.h"
#include <string>
#include <iostream>
```

```
#include <ostream>
```

Include dependency graph for errors\_imp.h:



This graph shows which files directly or indirectly include this file:



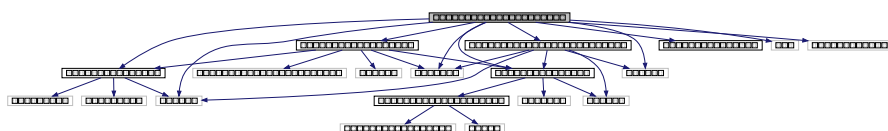
## Namespaces

- [glucat](#)

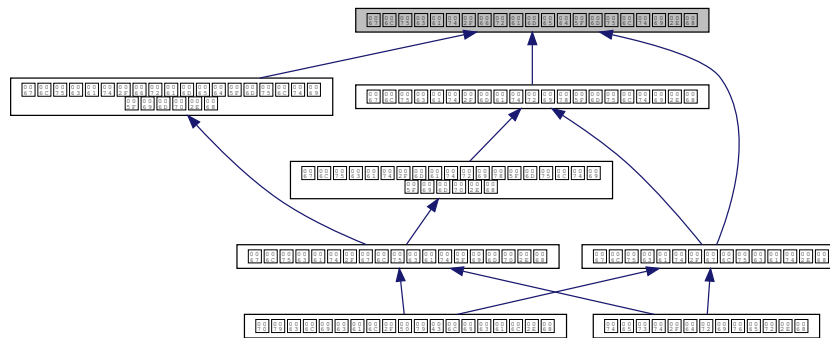
## 7.5 glucat/framed\_multi.h File Reference

```
#include "glucat/global.h"
#include "glucat/errors.h"
#include "glucat/index_set.h"
#include "glucat/clifford_algebra.h"
#include "glucat/tuning.h"
#include <string>
#include <utility>
#include <map>
#include <unordered_map>
#include <vector>
```

Include dependency graph for framed\_multi.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::framed\\_multi< Scalar\\_T, LO, HI, Tune\\_P >](#)  
A *framed\_multi<Scalar\_T,LO,HI,Tune\_P>* is a framed approximation to a multivector.
- class [glucat::matrix\\_multi< Scalar\\_T, LO, HI, Tune\\_P >](#)  
A *matrix\_multi<Scalar\_T,LO,HI,Tune\_P>* is a matrix approximation to a multivector.
- class [glucat::index\\_set\\_hash< LO, HI >](#)
- class [glucat::framed\\_multi< Scalar\\_T, LO, HI, Tune\\_P >](#)  
A *framed\_multi<Scalar\_T,LO,HI,Tune\_P>* is a framed approximation to a multivector.
- class [glucat::framed\\_multi< Scalar\\_T, LO, HI, Tune\\_P >::hash\\_size\\_t](#)
- class [glucat::framed\\_multi< Scalar\\_T, LO, HI, Tune\\_P >::var\\_term](#)  
Variable term.
- struct [std::numeric\\_limits< glucat::framed\\_multi< Scalar\\_T, LO, HI, Tune\\_P > >](#)  
Numeric limits for *framed\_multi* inherit limits for the corresponding scalar type.

## Namespaces

- [glucat](#)
- [std](#)

## Functions

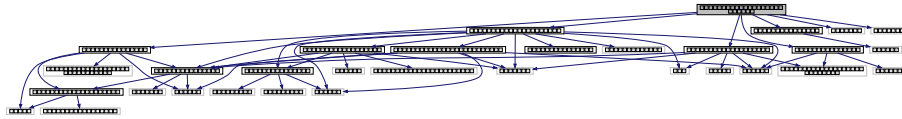
- template<typename [Scalar\\_T](#) , const index\_t [LO](#), const index\_t [HI](#), typename [Tune\\_P](#) >  
auto [glucat::operator\\*](#) (const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) > &lhs, const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) > &rhs) -> const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) >  
*Geometric product.*
- template<typename [Scalar\\_T](#) , const index\_t [LO](#), const index\_t [HI](#), typename [Tune\\_P](#) >  
auto [glucat::operator^](#) (const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) > &lhs, const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) > &rhs) -> const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) >  
*Outer product.*
- template<typename [Scalar\\_T](#) , const index\_t [LO](#), const index\_t [HI](#), typename [Tune\\_P](#) >  
auto [glucat::operator &](#) (const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) > &lhs, const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) > &rhs) -> const framed\_multi< [Scalar\\_T](#), [LO](#), [HI](#), [Tune\\_P](#) >  
*Inner product.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator% (const framed_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed_multi<`  
`Scalar_T, LO, HI, Tune_P > &rhs) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Left contraction.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::star (const framed_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed_multi< Scalar_T, LO,`  
`HI, Tune_P > &rhs) -> Scalar_T`  
*Hestenes scalar product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator/ (const framed_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed_multi< Scalar_↵`  
`_T, LO, HI, Tune_P > &rhs) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Geometric quotient.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator| (const framed_multi< Scalar_T, LO, HI, Tune_P > &lhs, const framed_multi< Scalar_↵`  
`_T, LO, HI, Tune_P > &rhs) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Transformation via twisted adjoint action.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator>> (std::istream &s, framed_multi< Scalar_T, LO, HI, Tune_P > &val) -> std::istream`  
`&`  
*Read multivector from input.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator<< (std::ostream &os, const framed_multi< Scalar_T, LO, HI, Tune_P > &val) -> std::↵`  
`::ostream &`  
*Write multivector to output.*
- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`auto glucat::operator<< (std::ostream &os, const std::pair< const index_set< LO, HI >, Scalar_T > &term)`  
`-> std::ostream &`  
*Write term to output.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::exp (const framed_multi< Scalar_T, LO, HI, Tune_P > &val) -> const framed_multi< Scalar_T,`  
`LO, HI, Tune_P >`  
*Exponential of multivector.*
- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`static auto glucat::crd_of_mult (const std::pair< const index_set< LO, HI >, Scalar_T > &lhs, const std::↵`  
`::pair< const index_set< LO, HI >, Scalar_T > &rhs) -> Scalar_T`  
*Coordinate of product of terms.*
- `template<typename Scalar_T , const index_t LO, const index_t HI>`  
`auto glucat::operator* (const std::pair< const index_set< LO, HI >, Scalar_T > &lhs, const std::pair< const`  
`index_set< LO, HI >, Scalar_T > &rhs) -> const std::pair< const index_set< LO, HI >, Scalar_T >`  
*Product of terms.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::sqrt (const framed_multi< Scalar_T, LO, HI, Tune_P > &val, const framed_multi< Scalar_T, LO,`  
`HI, Tune_P > &i, bool prechecked) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Square root of multivector with specified complexifier.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const framed_multi< Scalar_T, LO, HI, Tune_P > &val, const framed_multi< Scalar_T, LO,`  
`HI, Tune_P > &i, bool prechecked) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`  
*Natural logarithm of multivector with specified complexifier.*

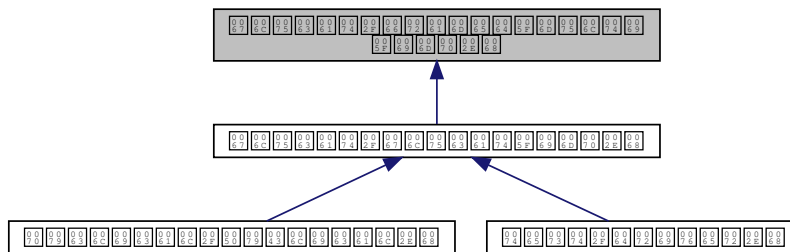
## 7.6 glucat/framed\_multi\_imp.h File Reference

```
#include "glucat/framed_multi.h"
#include "glucat/scalar.h"
```

```
#include "glucat/random.h"
#include "glucat/generation.h"
#include "glucat/matrix.h"
#include <sstream>
#include <fstream>
Include dependency graph for framed_multi_imp.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::sorted\\_range< Map\\_T, Sorted\\_Map\\_T >](#)  
*Sorted range for use with output.*
- class [glucat::sorted\\_range< Sorted\\_Map\\_T, Sorted\\_Map\\_T >](#)

## Namespaces

- [glucat](#)

## Macros

- [#define \\_GLUCAT\\_HASH\\_N\(x\) \(x\)](#)
- [#define \\_GLUCAT\\_HASH\\_SIZE\\_T\(x\) \(typename multivector\\_t::hash\\_size\\_t\)\(x\)](#)



## Functions

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator\\*](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_↵  
 \_T, LO, HI, Tune\_P > &rhs) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Geometric product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator^](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_↵  
 \_T, LO, HI, Tune\_P > &rhs) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Outer product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator&](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_↵  
 \_T, LO, HI, Tune\_P > &rhs) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Inner product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator%](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_↵  
 \_T, LO, HI, Tune\_P > &rhs) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Left contraction.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::star](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_T, LO,  
 HI, Tune\_P > &rhs) -> Scalar\_T  
*Hestenes scalar product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator/](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_↵  
 \_T, LO, HI, Tune\_P > &rhs) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Geometric quotient.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator|](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const framed\_multi< Scalar\_↵  
 \_T, LO, HI, Tune\_P > &rhs) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Transformation via twisted adjoint action.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator<<](#) (std::ostream &os, const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &val) -> std::↵  
 ostream &  
*Write multivector to output.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI>  
 auto [glucat::operator<<](#) (std::ostream &os, const std::pair< const index\_set< LO, HI >, Scalar\_T > &term)  
 -> std::ostream &  
*Write term to output.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator>>](#) (std::istream &s, framed\_multi< Scalar\_T, LO, HI, Tune\_P > &val) -> std::istream  
 &  
*Read multivector from input.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI>  
 static auto [glucat::crd\\_of\\_mult](#) (const std::pair< const index\_set< LO, HI >, Scalar\_T > &lhs, const std::↵  
 ::pair< const index\_set< LO, HI >, Scalar\_T > &rhs) -> Scalar\_T  
*Coordinate of product of terms.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI>  
 auto [glucat::operator\\*](#) (const std::pair< const index\_set< LO, HI >, Scalar\_T > &lhs, const std::pair< const  
 index\_set< LO, HI >, Scalar\_T > &rhs) -> const std::pair< const index\_set< LO, HI >, Scalar\_T >  
*Product of terms.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::sqrt](#) (const framed\_multi< Scalar\_T, LO, HI, Tune\_P > &val, const framed\_multi< Scalar\_T, LO,  
 HI, Tune\_P > &i, bool prechecked) -> const framed\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Square root of multivector with specified complexifier.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::exp (const framed_multi< Scalar_T, LO, HI, Tune_P > &val) -> const framed_multi< Scalar_T,`  
`LO, HI, Tune_P >`

*Exponential of multivector.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const framed_multi< Scalar_T, LO, HI, Tune_P > &val, const framed_multi< Scalar_T, LO,`  
`HI, Tune_P > &i, bool prechecked) -> const framed_multi< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

## 7.6.1 Macro Definition Documentation

### 7.6.1.1 \_GLUCAT\_HASH\_N

```
#define _GLUCAT_HASH_N(  
    x ) (x)
```

Definition at line 54 of file framed\_multi\_imp.h.

### 7.6.1.2 \_GLUCAT\_HASH\_SIZE\_T

```
#define _GLUCAT_HASH_SIZE_T(  
    x ) (typename multivector_t::hash_size_t)(x)
```

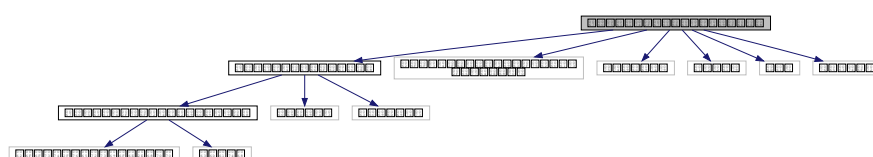
Definition at line 55 of file framed\_multi\_imp.h.

Referenced by `glucat::framed_multi< Scalar_T, LO, HI, Tune_P >::framed_multi()`, `glucat::operator &()`, `glucat::operator%()`, `glucat::operator*()`, and `glucat::operator^()`.

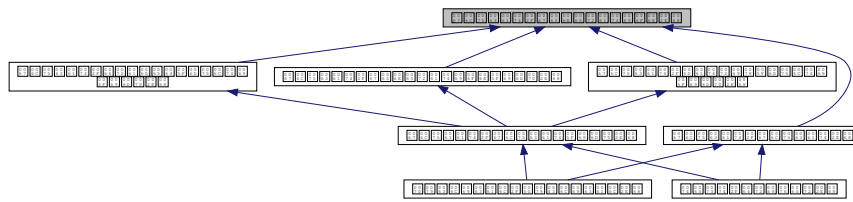
## 7.7 glucat/generation.h File Reference

```
#include "glucat/global.h"  
#include <boost/numeric/ublas/fwd.hpp>  
#include <utility>  
#include <array>  
#include <map>  
#include <vector>
```

Include dependency graph for generation.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::gen::generator\\_table< Matrix\\_T >](#)  
*Table of generators for specific signatures.*

## Namespaces

- [glucat](#)
- [glucat::gen](#)

## Typedefs

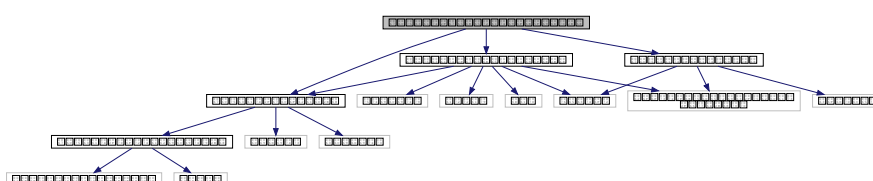
- using [glucat::gen::signature\\_t](#) = `std::pair< index_t, index_t >`  
*A signature is a pair of indices,  $p$ ,  $q$ , with  $p == \text{frame.max}()$ ,  $q == -\text{frame.min}()$*

## Variables

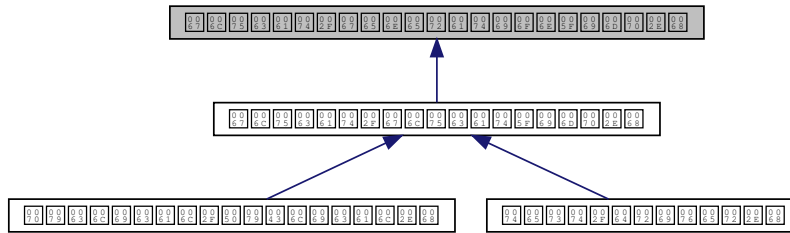
- static const `std::array< index_t, 8 >` [glucat::gen::offset\\_to\\_super](#) = {0,-1, 0,-1,-2, 3, 2, 1}  
*Offsets between the current signature and that of the real superalgebra.*

## 7.8 glucat/generation\_imp.h File Reference

```
#include "glucat/global.h"
#include "glucat/generation.h"
#include "glucat/matrix.h"
Include dependency graph for generation_imp.h:
```



This graph shows which files directly or indirectly include this file:

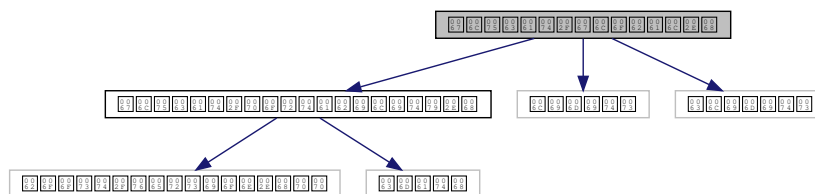


## Namespaces

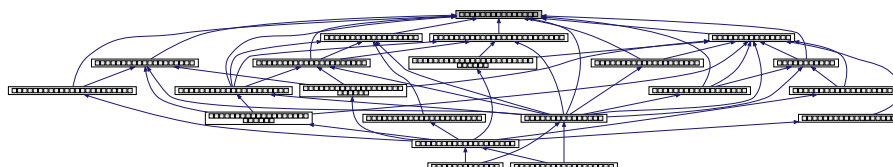
- [glucat](#)
- [glucat::gen](#)

## 7.9 glucat/global.h File Reference

```
#include "glucat/portability.h"
#include <limits>
#include <climits>
Include dependency graph for global.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- struct [glucat::CTAssertion< bool >](#)  
*Compile time assertion.*
- struct [glucat::CTAssertion< true >](#)
- class [glucat::compare\\_types< LHS\\_T, RHS\\_T >](#)  
*Type comparison.*
- class [glucat::compare\\_types< T, T >](#)
- class [glucat::bool\\_to\\_type< truth\\_value >](#)  
*Bool to type.*

## Namespaces

- [glucat](#)

## Macros

- `#define _GLUCAT_CTAssert(expr, msg) namespace { struct msg { glucat::CTAssertion<(expr)> ERROR↵  
_##msg; }; }`

## Typedefs

- using [glucat::index\\_t](#) = int  
*Size of index\_t should be enough to represent LO, HI.*
- using [glucat::set\\_value\\_t](#) = unsigned long  
*Size of set\_value\_t should be enough to contain index\_set<LO,HI>*

## Functions

- [glucat::\\_GLUCAT\\_CTAssert](#) (std::numeric\_limits< unsigned char >::radix==2, CannotDetermineBitsPer↵  
Char) const index\_t BITS\_PER\_CHAR  
*If radix of unsigned char is not 2, we can't easily determine number of bits from sizeof.*
- [glucat::\\_GLUCAT\\_CTAssert](#) (\_GLUCAT\_BITS\_PER\_ULONG==BITS\_PER\_SET\_VALUE, BitsPerULong↵  
DoesNotMatchSetValueT) const index\_t DEFAULT\_LO  
*Default lowest index in an index set.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::pos\\_mod](#) (LHS\_T lhs, RHS\_T rhs) -> LHS\_T  
*Modulo function which works reliably for lhs < 0.*

## Variables

- const double [glucat::MS\\_PER\\_S](#) = 1000.0  
*Timing constant: deprecated here - moved to [test/timing.h](#).*
- const index\_t [glucat::BITS\\_PER\\_SET\\_VALUE](#) = std::numeric\_limits<set\_value\_t>::digits  
*Number of bits in set\_value\_t.*
- const index\_t [glucat::DEFAULT\\_HI](#) = index\_t(BITS\_PER\_SET\_VALUE / 2)  
*Default highest index in an index set.*

## 7.9.1 Macro Definition Documentation

### 7.9.1.1 \_GLUCAT\_CTAssert

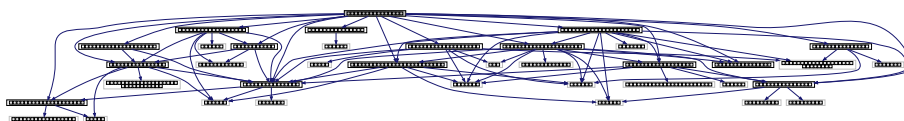
```
#define _GLUCAT_CTAssert(  
    expr,  
    msg ) namespace { struct msg { glucat::CTAssertion<(expr)> ERROR_##msg; }; }
```

Definition at line 48 of file global.h.

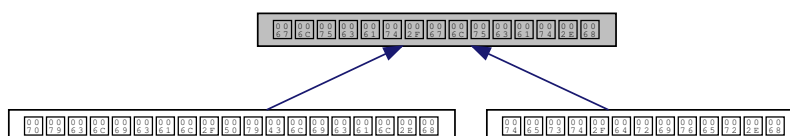
## 7.10 glucat/glucat.h File Reference

```
#include "glucat/portability.h"  
#include "glucat/global.h"  
#include "glucat/errors.h"  
#include "glucat/index_set.h"  
#include "glucat/scalar.h"  
#include "glucat/long_double.h"  
#include "glucat/qd.h"  
#include "glucat/promotion.h"  
#include "glucat/random.h"  
#include "glucat/clifford_algebra.h"  
#include "glucat/tuning.h"  
#include "glucat/framed_multi.h"  
#include "glucat/generation.h"  
#include "glucat/matrix.h"  
#include "glucat/matrix_multi.h"
```

Include dependency graph for glucat.h:

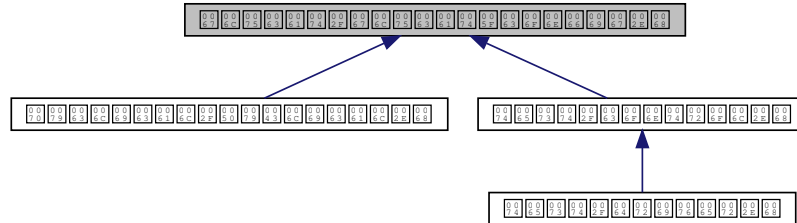


This graph shows which files directly or indirectly include this file:



## 7.11 glucat/glucat\_config.h File Reference

This graph shows which files directly or indirectly include this file:



### Macros

- `#define GLUCAT_HAVE_CXX11 1`
- `#define GLUCAT_HAVE_INTTYPES_H 1`
- `#define GLUCAT_HAVE_STDINT_H 1`
- `#define GLUCAT_HAVE_STDIO_H 1`
- `#define GLUCAT_HAVE_STDLIB_H 1`
- `#define GLUCAT_HAVE_STRINGS_H 1`
- `#define GLUCAT_HAVE_STRING_H 1`
- `#define GLUCAT_HAVE_SYS_STAT_H 1`
- `#define GLUCAT_HAVE_SYS_TYPES_H 1`
- `#define GLUCAT_HAVE_UNISTD_H 1`
- `#define GLUCAT_PACKAGE "glucat"`
- `#define GLUCAT_PACKAGE_BUGREPORT ""`
- `#define GLUCAT_PACKAGE_NAME "glucat"`
- `#define GLUCAT_PACKAGE_STRING "glucat 0.12.0"`
- `#define GLUCAT_PACKAGE_TARNAME "glucat"`
- `#define GLUCAT_PACKAGE_URL ""`
- `#define GLUCAT_PACKAGE_VERSION "0.12.0"`
- `#define GLUCAT_STDC_HEADERS 1`
- `#define GLUCAT_VERSION "0.12.0"`

### 7.11.1 Macro Definition Documentation

#### 7.11.1.1 GLUCAT\_HAVE\_CXX11

```
#define GLUCAT_HAVE_CXX11 1
```

Definition at line 20 of file glucat\_config.h.

#### 7.11.1.2 GLUCAT\_HAVE\_INTTYPES\_H

```
#define GLUCAT_HAVE_INTTYPES_H 1
```

Definition at line 28 of file `glucat_config.h`.

#### 7.11.1.3 GLUCAT\_HAVE\_STDINT\_H

```
#define GLUCAT_HAVE_STDINT_H 1
```

Definition at line 39 of file `glucat_config.h`.

#### 7.11.1.4 GLUCAT\_HAVE\_STDIO\_H

```
#define GLUCAT_HAVE_STDIO_H 1
```

Definition at line 44 of file `glucat_config.h`.

#### 7.11.1.5 GLUCAT\_HAVE\_STDLIB\_H

```
#define GLUCAT_HAVE_STDLIB_H 1
```

Definition at line 49 of file `glucat_config.h`.

#### 7.11.1.6 GLUCAT\_HAVE\_STRING\_H

```
#define GLUCAT_HAVE_STRING_H 1
```

Definition at line 59 of file `glucat_config.h`.

#### 7.11.1.7 GLUCAT\_HAVE\_STRINGS\_H

```
#define GLUCAT_HAVE_STRINGS_H 1
```

Definition at line 54 of file `glucat_config.h`.



#### 7.11.1.8 GLUCAT\_HAVE\_SYS\_STAT\_H

```
#define GLUCAT_HAVE_SYS_STAT_H 1
```

Definition at line 64 of file glucat\_config.h.

#### 7.11.1.9 GLUCAT\_HAVE\_SYS\_TYPES\_H

```
#define GLUCAT_HAVE_SYS_TYPES_H 1
```

Definition at line 69 of file glucat\_config.h.

#### 7.11.1.10 GLUCAT\_HAVE\_UNISTD\_H

```
#define GLUCAT_HAVE_UNISTD_H 1
```

Definition at line 74 of file glucat\_config.h.

#### 7.11.1.11 GLUCAT\_PACKAGE

```
#define GLUCAT_PACKAGE "glucat"
```

Definition at line 79 of file glucat\_config.h.

#### 7.11.1.12 GLUCAT\_PACKAGE\_BUGREPORT

```
#define GLUCAT_PACKAGE_BUGREPORT ""
```

Definition at line 84 of file glucat\_config.h.

#### 7.11.1.13 GLUCAT\_PACKAGE\_NAME

```
#define GLUCAT_PACKAGE_NAME "glucat"
```

Definition at line 89 of file glucat\_config.h.

Referenced by glucat::control\_t::control\_t().

#### 7.11.1.14 GLUCAT\_PACKAGE\_STRING

```
#define GLUCAT_PACKAGE_STRING "glucat 0.12.0"
```

Definition at line 94 of file glucat\_config.h.

#### 7.11.1.15 GLUCAT\_PACKAGE\_TARNAME

```
#define GLUCAT_PACKAGE_TARNAME "glucat"
```

Definition at line 99 of file glucat\_config.h.

#### 7.11.1.16 GLUCAT\_PACKAGE\_URL

```
#define GLUCAT_PACKAGE_URL ""
```

Definition at line 104 of file glucat\_config.h.

#### 7.11.1.17 GLUCAT\_PACKAGE\_VERSION

```
#define GLUCAT_PACKAGE_VERSION "0.12.0"
```

Definition at line 109 of file glucat\_config.h.

#### 7.11.1.18 GLUCAT\_STDC\_HEADERS

```
#define GLUCAT_STDC_HEADERS 1
```

Definition at line 116 of file glucat\_config.h.

#### 7.11.1.19 GLUCAT\_VERSION

```
#define GLUCAT_VERSION "0.12.0"
```

Definition at line 121 of file glucat\_config.h.

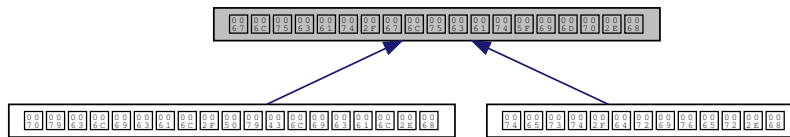
Referenced by glucat::control\_t::control\_t().

## 7.12 glucat/glucat\_imp.h File Reference

```
#include "glucat/errors_imp.h"
#include "glucat/index_set_imp.h"
#include "glucat/scalar_imp.h"
#include "glucat/clifford_algebra_imp.h"
#include "glucat/random.h"
#include "glucat/framed_multi_imp.h"
#include "glucat/matrix_imp.h"
#include "glucat/generation_imp.h"
#include "glucat/matrix_multi_imp.h"
Include dependency graph for glucat_imp.h:
```

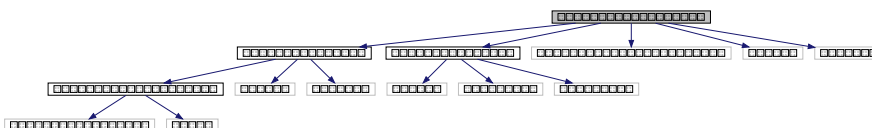


This graph shows which files directly or indirectly include this file:

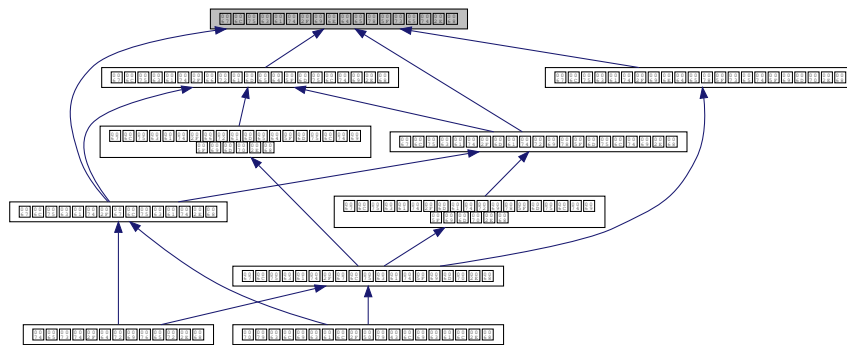


## 7.13 glucat/index\_set.h File Reference

```
#include "glucat/global.h"
#include "glucat/errors.h"
#include <boost/static_assert.hpp>
#include <bitset>
#include <utility>
Include dependency graph for index_set.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::index\\_set< LO, HI >](#)  
*Index set class based on `std::bitset<>` in Gnu standard C++ library.*
- class [glucat::index\\_set< LO, HI >](#)  
*Index set class based on `std::bitset<>` in Gnu standard C++ library.*
- class [glucat::index\\_set< LO, HI >::reference](#)  
*Index set member reference.*

## Namespaces

- [glucat](#)

## Functions

- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator^ (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO, HI >`  
*Symmetric set difference: exclusive or.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator & (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO, HI >`  
*Set intersection: and.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator| (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO, HI >`  
*Set union: or.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::compare (const index_set< LO, HI > &a, const index_set< LO, HI > &b) -> int`  
*"lexicographic compare" eg. {3,4,5} is less than {3,7,8}*
- `glucat::GLUCAT\_CTAssert (sizeof(set_value_t) >=sizeof(std::bitset< DEFAULT_HI-DEFAULT_LO >), Default_index_set_too_big_for_value) template< const index_t LO`  
*Size of set\_value\_t should be enough to contain `bitset<DEFAULT_HI-DEFAULT_LO>`*
- `const index_t HI auto glucat::operator<< (std::ostream &os, const index_set< LO, HI > &ist) -> std::ostream &`

*Write out index set.*

- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator>> (std::istream &s, index_set< LO, HI > &ist) -> std::istream &`

*Read in index set.*

- `auto glucat::sign\_of\_square (index_t j) -> int`

*Square of generator {j}.*

- `template<const index_t LO, const index_t HI>`  
`auto glucat::min\_neg (const index_set< LO, HI > &ist) -> index_t`

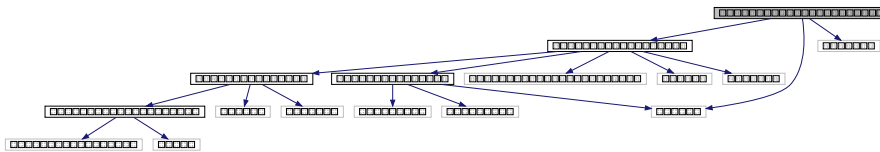
*Minimum negative index, or 0 if none.*

- `template<const index_t LO, const index_t HI>`  
`auto glucat::max\_pos (const index_set< LO, HI > &ist) -> index_t`

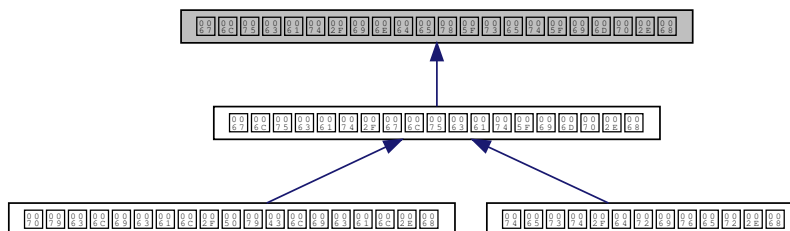
*Maximum positive index, or 0 if none.*

## 7.14 glucat/index\_set\_imp.h File Reference

```
#include "glucat/index_set.h"
#include <string>
#include <sstream>
Include dependency graph for index_set_imp.h:
```



This graph shows which files directly or indirectly include this file:



## Namespaces

- [glucat](#)

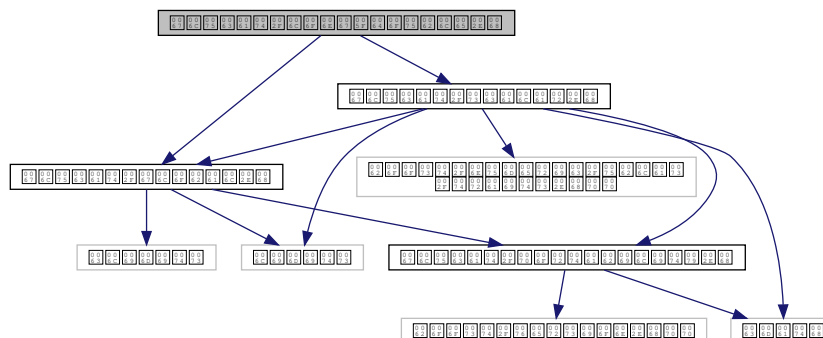
## Functions

- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator^ (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO, HI >`  
*Symmetric set difference: exclusive or.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator & (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO, HI >`  
*Set intersection: and.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator| (const index_set< LO, HI > &lhs, const index_set< LO, HI > &rhs) -> const index_set< LO, HI >`  
*Set union: or.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::compare (const index_set< LO, HI > &a, const index_set< LO, HI > &b) -> int`  
*"lexicographic compare" eg. {3,4,5} is less than {3,7,8}*
- `const index_t HI auto glucat::operator<< (std::ostream &os, const index_set< LO, HI > &ist) -> std::ostream &`  
*Write out index set.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::operator>> (std::istream &s, index_set< LO, HI > &ist) -> std::istream &`  
*Read in index set.*
- `static auto glucat::inverse_reversed_gray (unsigned long x) -> unsigned long`  
*Inverse reversed Gray code.*
- `static auto glucat::inverse_gray (unsigned long x) -> unsigned long`  
*Inverse Gray code.*
- `auto glucat::sign_of_square (index_t j) -> int`  
*Square of generator {j}.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::min_neg (const index_set< LO, HI > &ist) -> index_t`  
*Minimum negative index, or 0 if none.*
- `template<const index_t LO, const index_t HI>`  
`auto glucat::max_pos (const index_set< LO, HI > &ist) -> index_t`  
*Maximum positive index, or 0 if none.*

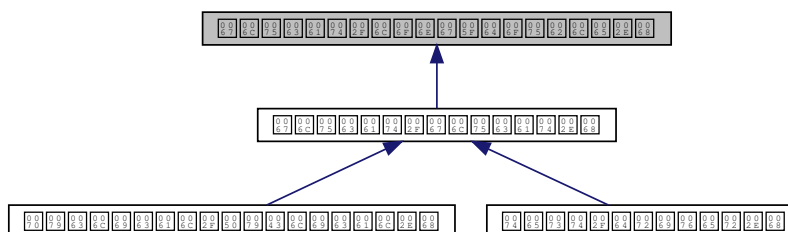
## 7.15 glucat/long\_double.h File Reference

```
#include "glucat/global.h"
#include "glucat/scalar.h"
```

Include dependency graph for long\_double.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [glucat](#)

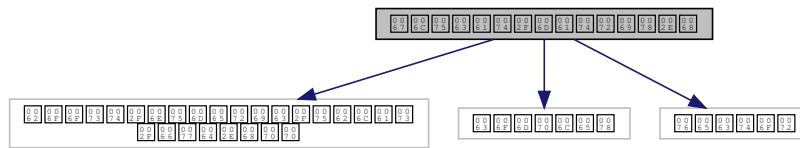
## Variables

- static const long double [glucat::l\\_pi](#) = 3.1415926535897932384626433832795029L
- static const long double [glucat::l\\_ln2](#) = 0.6931471805599453094172321214581766L

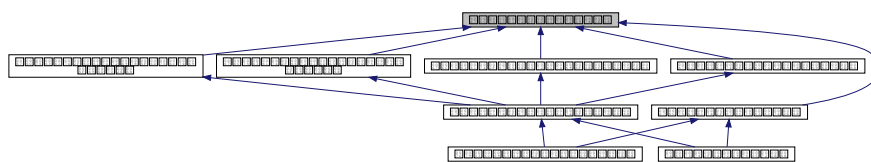
## 7.16 glucat/matrix.h File Reference

```
#include <boost/numeric/ublas/fwd.hpp>
#include <complex>
```

```
#include <vector>
Include dependency graph for matrix.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- struct [glucat::matrix::eig\\_genus< Matrix\\_T >](#)  
*Structure containing classification of eigenvalues.*

## Namespaces

- [glucat](#)
- [glucat::matrix](#)

## Typedefs

- using [glucat::matrix::eig\\_case\\_t](#) = enum { safe\_eigs, neg\_real\_eigs, both\_eigs}  
*Classification of eigenvalues of a matrix.*

## Functions

- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::kron](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> const RHS\_T  
*Kronecker tensor product of matrices - as per Matlab kron.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::mono\\_kron](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> const RHS\_T  
*Sparse Kronecker tensor product of monomial matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::nork](#) (const LHS\_T &lhs, const RHS\_T &rhs, const bool mono=true) -> const RHS\_T  
*Left inverse of Kronecker product.*



- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::signed\\_perm\\_nork](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> const RHS\_T  
*Left inverse of Kronecker product where lhs is a signed permutation matrix.*
- template<typename Matrix\_T >  
auto [glucat::matrix::nnz](#) (const Matrix\_T &m) -> typename Matrix\_T::size\_type  
*Number of non-zeros.*
- template<typename Matrix\_T >  
auto [glucat::matrix::isinf](#) (const Matrix\_T &m) -> bool  
*Infinite.*
- template<typename Matrix\_T >  
auto [glucat::matrix::isnan](#) (const Matrix\_T &m) -> bool  
*Not a Number.*
- template<typename Matrix\_T >  
auto [glucat::matrix::unit](#) (const typename Matrix\_T::size\_type n) -> const Matrix\_T  
*Unit matrix - as per Matlab eye.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::mono\\_prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of monomial matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::sparse\\_prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of sparse matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of matrices.*
- template<typename Scalar\_T , typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::inner](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> Scalar\_T  
*Inner product:  $\sum(x(i,j)*y(i,j))/x.nrows()$*
- template<typename Matrix\_T >  
auto [glucat::matrix::norm\\_frob2](#) (const Matrix\_T &val) -> typename Matrix\_T::value\_type  
*Square of Frobenius norm.*
- template<typename Matrix\_T >  
auto [glucat::matrix::trace](#) (const Matrix\_T &val) -> typename Matrix\_T::value\_type  
*Matrix trace.*
- template<typename Matrix\_T >  
auto [glucat::matrix::eigenvalues](#) (const Matrix\_T &val) -> std::vector< std::complex< double > >  
*Eigenvalues of a matrix.*
- template<typename Matrix\_T >  
auto [glucat::matrix::classify\\_eigenvalues](#) (const Matrix\_T &val) -> eig\_genus< Matrix\_T >  
*Classify the eigenvalues of a matrix.*

## 7.17 glucat/matrix\_imp.h File Reference

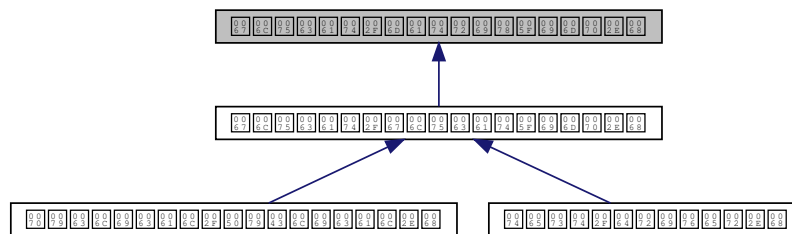
```
#include "glucat/errors.h"
#include "glucat/scalar.h"
#include "glucat/matrix.h"
#include <boost/numeric/ublas/vector.hpp>
#include <boost/numeric/ublas/vector_proxy.hpp>
#include <boost/numeric/ublas/matrix.hpp>
#include <boost/numeric/ublas/matrix_expression.hpp>
```

```
#include <boost/numeric/ublas/matrix_proxy.hpp>
#include <boost/numeric/ublas/matrix_sparse.hpp>
#include <boost/numeric/ublas/operation.hpp>
#include <boost/numeric/ublas/operation_sparse.hpp>
#include <boost/numeric/bindings/lapack/driver/gees.hpp>
#include <boost/numeric/bindings/ublas.hpp>
#include <set>
#include <vector>
```

Include dependency graph for matrix\_imp.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [glucat](#)
- [glucat::matrix](#)

## Functions

- `template<typename LHS_T , typename RHS_T >`  
`auto glucat::matrix::kron (const LHS_T &lhs, const RHS_T &rhs) -> const RHS_T`  
*Kronecker tensor product of matrices - as per Matlab kron.*
- `template<typename LHS_T , typename RHS_T >`  
`auto glucat::matrix::mono\_kron (const LHS_T &lhs, const RHS_T &rhs) -> const RHS_T`  
*Sparse Kronecker tensor product of monomial matrices.*
- `template<typename LHS_T , typename RHS_T >`  
`void glucat::matrix::nork\_range (RHS_T &result, const typename LHS_T::const_iterator2 lhs_it2, const RHS_T &rhs, const typename RHS_T::size_type res_s1, const typename RHS_T::size_type res_s2)`  
*Utility routine for nork: calculate result for a range of indices.*
- `template<typename LHS_T , typename RHS_T >`  
`auto glucat::matrix::nork (const LHS_T &lhs, const RHS_T &rhs, const bool mono=true) -> const RHS_T`  
*Left inverse of Kronecker product.*
- `template<typename LHS_T , typename RHS_T >`  
`auto glucat::matrix::signed\_perm\_nork (const LHS_T &lhs, const RHS_T &rhs) -> const RHS_T`  
*Left inverse of Kronecker product where lhs is a signed permutation matrix.*

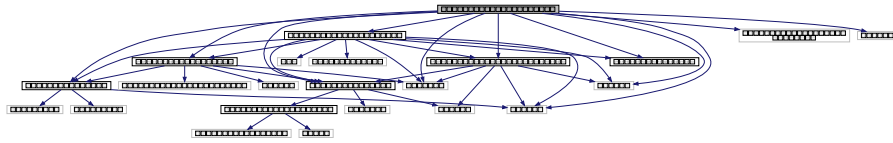
- template<typename Matrix\_T >  
auto [glucat::matrix::nnz](#) (const Matrix\_T &m) -> typename Matrix\_T::size\_type  
*Number of non-zeros.*
- template<typename Matrix\_T >  
auto [glucat::matrix::isinf](#) (const Matrix\_T &m) -> bool  
*Infinite.*
- template<typename Matrix\_T >  
auto [glucat::matrix::isnan](#) (const Matrix\_T &m) -> bool  
*Not a Number.*
- template<typename Matrix\_T >  
auto [glucat::matrix::unit](#) (const typename Matrix\_T::size\_type n) -> const Matrix\_T  
*Unit matrix - as per Matlab eye.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::mono\\_prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of monomial matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::sparse\\_prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of sparse matrices.*
- template<typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::prod](#) (const ublas::matrix\_expression< LHS\_T > &lhs, const ublas::matrix\_expression< RHS\_T > &rhs) -> const typename RHS\_T::expression\_type  
*Product of matrices.*
- template<typename Scalar\_T , typename LHS\_T , typename RHS\_T >  
auto [glucat::matrix::inner](#) (const LHS\_T &lhs, const RHS\_T &rhs) -> Scalar\_T  
*Inner product:  $\sum(x(i,j)*y(i,j))/x.nrows()$*
- template<typename Matrix\_T >  
auto [glucat::matrix::norm\\_frob2](#) (const Matrix\_T &val) -> typename Matrix\_T::value\_type  
*Square of Frobenius norm.*
- template<typename Matrix\_T >  
auto [glucat::matrix::trace](#) (const Matrix\_T &val) -> typename Matrix\_T::value\_type  
*Matrix trace.*
- template<typename Matrix\_T >  
static auto [glucat::matrix::to\\_lapack](#) (const Matrix\_T &val) -> ublas::matrix< double, ublas::column\_major >  
*Convert matrix to LAPACK format.*
- template<typename Matrix\_T >  
auto [glucat::matrix::eigenvalues](#) (const Matrix\_T &val) -> std::vector< std::complex< double > >  
*Eigenvalues of a matrix.*
- template<typename Matrix\_T >  
auto [glucat::matrix::classify\\_eigenvalues](#) (const Matrix\_T &val) -> eig\_genus< Matrix\_T >  
*Classify the eigenvalues of a matrix.*

## 7.18 glucat/matrix\_multi.h File Reference

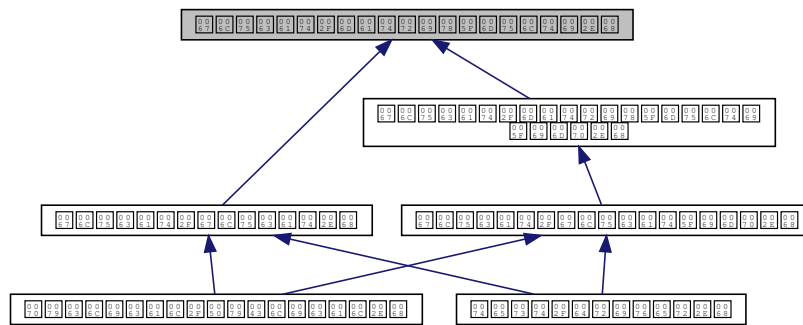
```
#include "glucat/global.h"
#include "glucat/errors.h"
#include "glucat/index_set.h"
#include "glucat/clifford_algebra.h"
#include "glucat/tuning.h"
#include "glucat/framed_multi.h"
#include <boost/numeric/ublas/fwd.hpp>
```

```
#include <fstream>
#include <string>
#include <utility>
#include <vector>
```

Include dependency graph for `matrix_multi.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::framed\\_multi< Scalar\\_T, LO, HI, Tune\\_P >](#)  
A `framed_multi<Scalar_T,LO,HI,Tune_P>` is a framed approximation to a multivector.
- class [glucat::matrix\\_multi< Scalar\\_T, LO, HI, Tune\\_P >](#)  
A `matrix_multi<Scalar_T,LO,HI,Tune_P>` is a matrix approximation to a multivector.
- class [glucat::matrix\\_multi< Scalar\\_T, LO, HI, Tune\\_P >](#)  
A `matrix_multi<Scalar_T,LO,HI,Tune_P>` is a matrix approximation to a multivector.
- struct [std::numeric\\_limits< glucat::matrix\\_multi< Scalar\\_T, LO, HI, Tune\\_P > >](#)  
Numeric limits for `matrix_multi` inherit limits for the corresponding scalar type.

## Namespaces

- [glucat](#)
- [std](#)

## Functions

- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator\\*](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Geometric product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator^](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Outer product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator &](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Inner product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator%](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Left contraction.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::star](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> Scalar\_T  
*Hestenes scalar product.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator/](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Geometric quotient.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator|](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Transformation via twisted adjoint action.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator>>](#) (std::istream &s, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &val) -> std::istream &  
*Read multivector from input.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::operator<<](#) (std::ostream &os, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &val) -> std::ostream &  
*Write multivector to output.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::reframe](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs\_reframed, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs\_reframed) -> const index\_set< LO, HI >  
*Find a common frame for operands of a binary operator.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::sqrt](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &val, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &i, bool prechecked) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Square root of multivector with specified complexifier.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::matrix\\_sqrt](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &val, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &i, const index\_t level) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Square root of multivector with specified complexifier.*
- template<typename Scalar\_T , const index\_t LO, const index\_t HI, typename Tune\_P >  
 auto [glucat::log](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &val, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &i, bool prechecked) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Natural logarithm of multivector with specified complexifier.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::matrix\_log (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, const matrix_multi< Scalar_T, LO, HI, Tune_P > &i, const index_t level) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

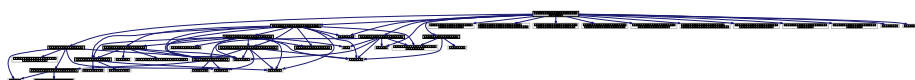
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::exp (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`

*Exponential of multivector.*

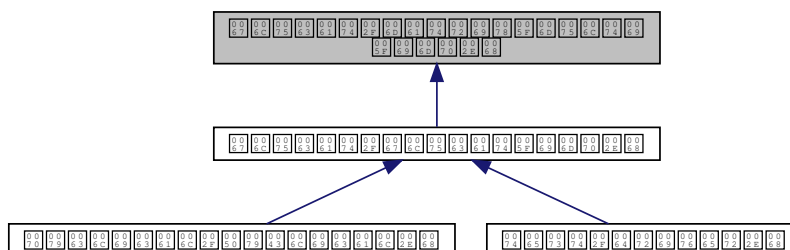
## 7.19 glucat/matrix\_multi\_imp.h File Reference

```
#include "glucat/matrix_multi.h"
#include "glucat/scalar.h"
#include "glucat/generation.h"
#include "glucat/matrix.h"
#include <boost/numeric/ublas/matrix.hpp>
#include <boost/numeric/ublas/matrix_expression.hpp>
#include <boost/numeric/ublas/matrix_proxy.hpp>
#include <boost/numeric/ublas/matrix_sparse.hpp>
#include <boost/numeric/ublas/operation.hpp>
#include <boost/numeric/ublas/operation_sparse.hpp>
#include <boost/numeric/ublas/triangular.hpp>
#include <boost/numeric/ublas/lu.hpp>
#include <boost/numeric/ublas/io.hpp>
#include <fstream>
#include <iomanip>
#include <array>
#include <iostream>
```

Include dependency graph for matrix\_multi\_imp.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::basis\\_table](#)< Scalar\_T, LO, HI, Matrix\_T >  
*Table of basis elements used as a cache by basis\_element()*
- struct [pade::pade\\_sqrt\\_numer](#)< Scalar\_T >  
*Coefficients of numerator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)*
- struct [pade::pade\\_sqrt\\_denom](#)< Scalar\_T >  
*Coefficients of denominator polynomials of Pade approximations produced by Pade1(sqrt(1+x),x,n,n)*
- struct [pade::pade\\_sqrt\\_numer](#)< float >
- struct [pade::pade\\_sqrt\\_denom](#)< float >
- struct [pade::pade\\_sqrt\\_numer](#)< long double >
- struct [pade::pade\\_sqrt\\_denom](#)< long double >
- struct [pade::pade\\_sqrt\\_numer](#)< dd\_real >
- struct [pade::pade\\_sqrt\\_denom](#)< dd\_real >
- struct [pade::pade\\_sqrt\\_numer](#)< qd\_real >
- struct [pade::pade\\_sqrt\\_denom](#)< qd\_real >
- struct [pade::pade\\_log\\_numer](#)< Scalar\_T >  
*Coefficients of numerator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)*
- struct [pade::pade\\_log\\_denom](#)< Scalar\_T >  
*Coefficients of denominator polynomials of Pade approximations produced by Pade1(log(1+x),x,n,n)*
- struct [pade::pade\\_log\\_numer](#)< float >
- struct [pade::pade\\_log\\_denom](#)< float >
- struct [pade::pade\\_log\\_numer](#)< long double >
- struct [pade::pade\\_log\\_denom](#)< long double >
- struct [pade::pade\\_log\\_numer](#)< dd\_real >
- struct [pade::pade\\_log\\_denom](#)< dd\_real >
- struct [pade::pade\\_log\\_numer](#)< qd\_real >
- struct [pade::pade\\_log\\_denom](#)< qd\_real >

## Namespaces

- [glucat](#)
- [pade](#)

## Functions

- auto [glucat::offset\\_level](#) (const index\_t p, const index\_t q) -> index\_t  
*Determine the log2 dim corresponding to signature p, q.*
- template<typename Matrix\_Index\_T, const index\_t LO, const index\_t HI>  
static auto [glucat::folded\\_dim](#) (const index\_set< LO, HI > &sub) -> Matrix\_Index\_T  
*Determine the matrix dimension of the fold of a subalgebra.*
- template<typename Scalar\_T, const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [glucat::reframe](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs\_reframed, matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs\_reframed) -> const index\_set< LO, HI >  
*Find a common frame for operands of a binary operator.*
- template<typename Scalar\_T, const index\_t LO, const index\_t HI, typename Tune\_P >  
auto [glucat::operator\\*](#) (const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &lhs, const matrix\_multi< Scalar\_T, LO, HI, Tune\_P > &rhs) -> const matrix\_multi< Scalar\_T, LO, HI, Tune\_P >  
*Geometric product.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator^ (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T,`  
`LO, HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Outer product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator & (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_↵`  
`_T, LO, HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Inner product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator% (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_↵`  
`_T, LO, HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Left contraction.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::star (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P > &rhs) -> Scalar_T`  
*Hestenes scalar product.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator/ (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T,`  
`LO, HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Geometric quotient.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator| (const matrix_multi< Scalar_T, LO, HI, Tune_P > &lhs, const matrix_multi< Scalar_T,`  
`LO, HI, Tune_P > &rhs) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Transformation via twisted adjoint action.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator<< (std::ostream &os, const matrix_multi< Scalar_T, LO, HI, Tune_P > &val) -> std_↵`  
`::ostream &`  
*Write multivector to output.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::operator>> (std::istream &s, matrix_multi< Scalar_T, LO, HI, Tune_P > &val) -> std::istream &`  
*Read multivector from input.*
- `template<typename Multivector_T , typename Matrix_T , typename Basis_Matrix_T >`  
`static auto glucat::fast (const Matrix_T &X, index_t level) -> Multivector_T`  
*Inverse generalized Fast Fourier Transform.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P , const size_t Size>`  
`static auto glucat::pade\_approx (const std::array< Scalar_T, Size > &numer, const std::array< Scalar_T,`  
`Size > &denom, const matrix_multi< Scalar_T, LO, HI, Tune_P > &X) -> const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P >`  
*Pade' approximation.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static void glucat::db\_step (matrix_multi< Scalar_T, LO, HI, Tune_P > &M, matrix_multi< Scalar_T, LO, HI,`  
`Tune_P > &Y)`  
*Single step of product form of Denman-Beavers square root iteration.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto glucat::db\_sqrt (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, Scalar_T norm_tol=std_↵`  
`::pow(std::numeric_limits< Scalar_T >::epsilon(), 4)) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Product form of Denman-Beavers square root iteration.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto glucat::cr\_sqrt (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, Scalar_T norm_Y_tol=std_↵`  
`::pow(std::numeric_limits< Scalar_T >::epsilon(), 1)) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`  
*Cyclic reduction square root iteration.*
- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::matrix\_sqrt (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, const matrix_multi< Scalar_↵`  
`_T, LO, HI, Tune_P > &i, const index_t level) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`



*Square root of multivector with specified complexifier.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::sqrt (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P > &i, bool prechecked) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`

*Square root of multivector with specified complexifier.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto glucat::pade_log (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val) -> const matrix_multi<`  
`Scalar_T, LO, HI, Tune_P >`

*Pade' approximation of log.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`static auto glucat::cascade_log (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val) -> const matrix_↵`  
`multi< Scalar_T, LO, HI, Tune_P >`

*Incomplete square root cascade and Pade' approximation of log.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::matrix_log (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, const matrix_multi< Scalar_↵`  
`_T, LO, HI, Tune_P > &i, const index_t level) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::log (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val, const matrix_multi< Scalar_T, LO,`  
`HI, Tune_P > &i, bool prechecked) -> const matrix_multi< Scalar_T, LO, HI, Tune_P >`

*Natural logarithm of multivector with specified complexifier.*

- `template<typename Scalar_T , const index_t LO, const index_t HI, typename Tune_P >`  
`auto glucat::exp (const matrix_multi< Scalar_T, LO, HI, Tune_P > &val) -> const matrix_multi< Scalar_T,`  
`LO, HI, Tune_P >`

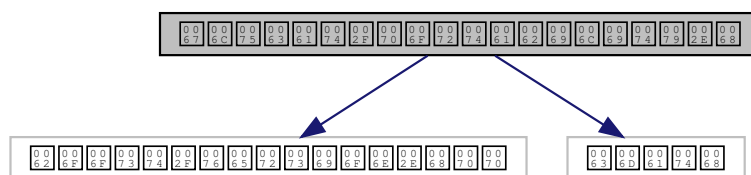
*Exponential of multivector.*

## 7.20 glucat/portability.h File Reference

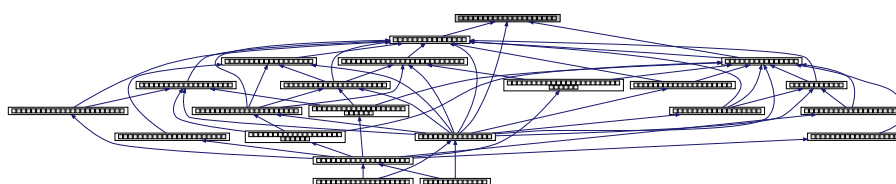
```
#include <boost/version.hpp>
```

```
#include <cmath>
```

Include dependency graph for portability.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define _GLUCAT_ISNAN(x) (x != x)`
- `#define _GLUCAT_ISINF(x) (!_GLUCAT_ISNAN(x) && _GLUCAT_ISNAN(x-x))`
- `#define UBLAS_ABS abs`
- `#define UBLAS_SQRT sqrt`

### 7.20.1 Macro Definition Documentation

#### 7.20.1.1 \_GLUCAT\_ISINF

```
#define _GLUCAT_ISINF(  
    x ) ( !_GLUCAT_ISNAN(x) && _GLUCAT_ISNAN(x-x) )
```

Definition at line 43 of file portability.h.

Referenced by `glucat::numeric_traits< Scalar_T >::isInf()`.

#### 7.20.1.2 \_GLUCAT\_ISNAN

```
#define _GLUCAT_ISNAN(  
    x ) ( x != x )
```

Definition at line 42 of file portability.h.

Referenced by `glucat::numeric_traits< Scalar_T >::isNaN()`.

#### 7.20.1.3 UBLAS\_ABS

```
#define UBLAS_ABS abs
```

Definition at line 51 of file portability.h.

Referenced by `glucat::numeric_traits< Scalar_T >::abs()`.

#### 7.20.1.4 UBLAS\_SQRT

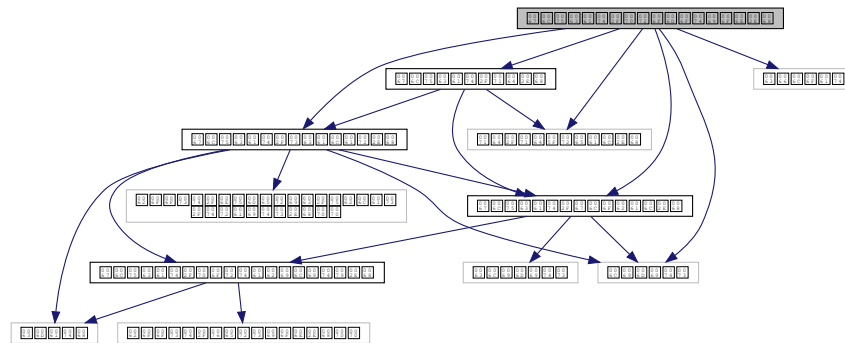
```
#define UBLAS_SQRT sqrt
```

Definition at line 52 of file portability.h.

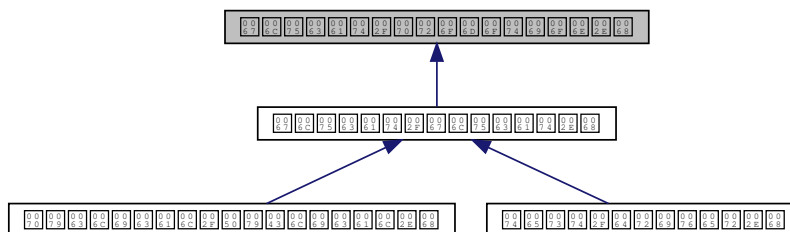
Referenced by `glucat::numeric_traits< Scalar_T >::sqrt()`.

## 7.21 glucat/promotion.h File Reference

```
#include "glucat/global.h"
#include "glucat/scalar.h"
#include "glucat/qd.h"
#include <cfloat>
#include <limits>
#include <qd/qd_real.h>
Include dependency graph for promotion.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

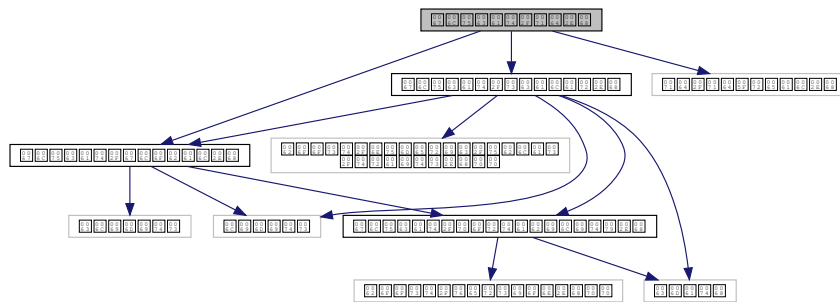
- struct `glucat::numeric_traits< Scalar_T >::promoted<>`  
Extra traits which extend numeric limits.
- struct `glucat::numeric_traits< Scalar_T >::demoted<>`  
Demoted type for long double.
- struct `glucat::numeric_traits< Scalar_T >::promoted<>`  
Extra traits which extend numeric limits.

### Namespaces

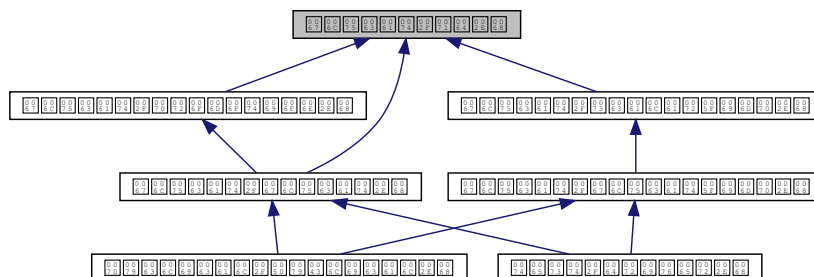
- `glucat`

## 7.22 glucat/qd.h File Reference

```
#include "glucat/global.h"
#include "glucat/scalar.h"
#include <qd/qd_real.h>
Include dependency graph for qd.h:
```



This graph shows which files directly or indirectly include this file:



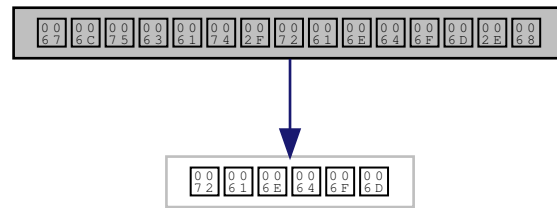
### Namespaces

- [glucat](#)

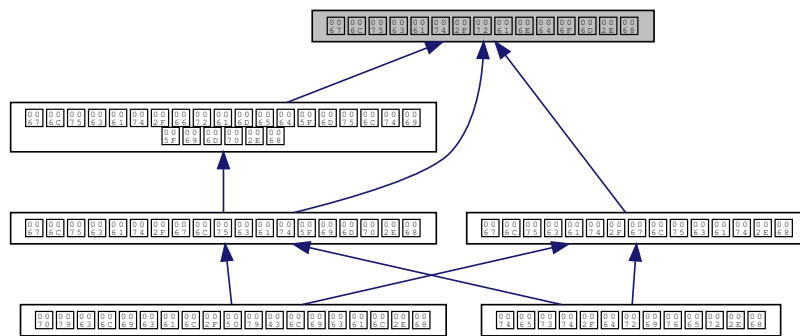
## 7.23 glucat/random.h File Reference

```
#include <random>
```

Include dependency graph for random.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glucat::random\\_generator< Scalar\\_T >](#)  
*Random number generator with single instance per Scalar\_T.*

## Namespaces

- [glucat](#)

## 7.24 glucat/scalar.h File Reference

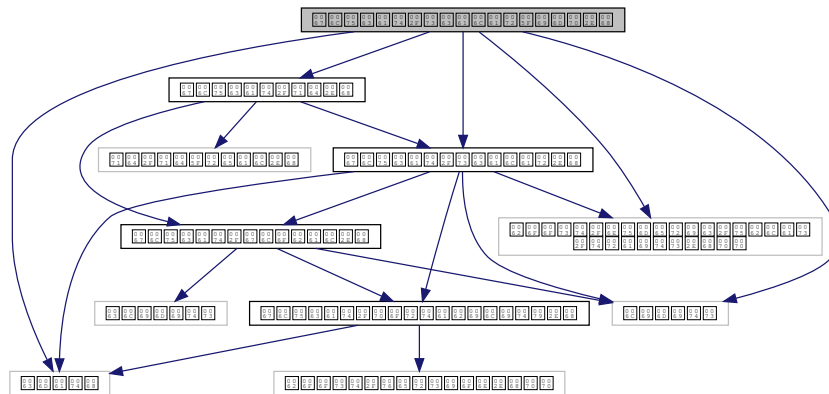
```
#include "glucat/portability.h"
#include "glucat/global.h"
#include <boost/numeric/ublas/traits.hpp>
#include <cmath>
```



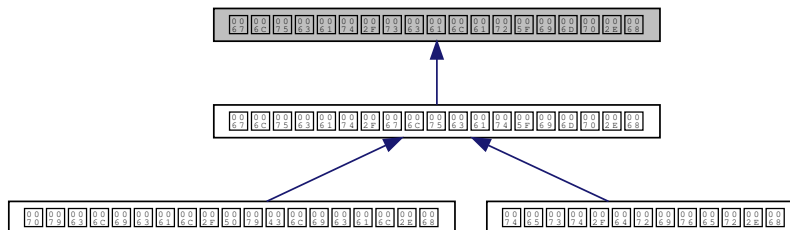
## 7.25 glucat/scalar\_imp.h File Reference

```
#include "glucat/scalar.h"
#include "glucat/qd.h"
#include <boost/numeric/ublas/traits.hpp>
#include <cmath>
#include <limits>
```

Include dependency graph for scalar\_imp.h:



This graph shows which files directly or indirectly include this file:



### Namespaces

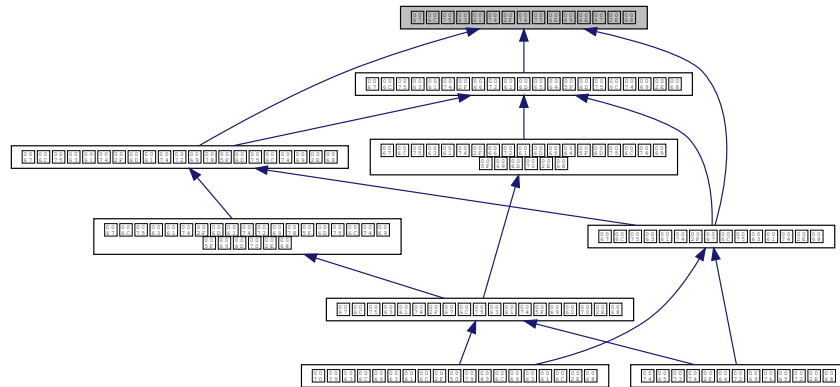
- [glucat](#)

### Functions

- `template<typename Scalar_T >`  
`auto glucat::to\_promote (const Scalar_T &val) -> typename numeric_traits< Scalar_T >::promoted::type`  
*Cast to promote.*
- `template<typename Scalar_T >`  
`auto glucat::to\_demote (const Scalar_T &val) -> typename numeric_traits< Scalar_T >::demoted::type`  
*Cast to demote.*

## 7.26 glucat/tuning.h File Reference

This graph shows which files directly or indirectly include this file:



### Functions

- [\\_GLUCAT\\_CTAssert](#) (std::numeric\_limits< unsigned int >::radix==2, CannotSetThresholds) namespace glucat

### 7.26.1 Function Documentation

#### 7.26.1.1 \_GLUCAT\_CTAssert()

```
_GLUCAT_CTAssert (
    std::numeric_limits< unsigned int >::radix == 2,
    CannotSetThresholds )
```

Base class for policies

Precision policy

Tuning policy

Minimum index count needed to invoke matrix multiplication algorithm

Maximum steps of iterative refinement in division algorithm

Maximum number of steps in cyclic reduction square root iteration

Maximum number of steps in Denman-Beavers square root iteration

Maximum number of incomplete square roots in cascade log algorithm

Maximum number of steps in incomplete square root within cascade log algorithm



Maximum index count of folded frames in basis cache

Minimum map size needed to invoke generalized FFT

Minimum matrix dimension needed to invoke inverse generalized FFT

Minimum size needed for to invoke faster products algorithms

Denominator of proportion of different bits allowed in approximate equality

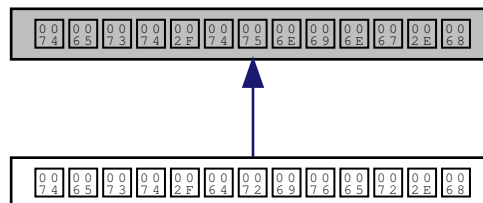
Extra number of different bits allowed in approximate equality

Precision used for exp, log and sqrt functions

Definition at line 35 of file tuning.h.

## 7.27 test/tuning.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [glucat](#)

### Typedefs

- using [glucat::tuning\\_slow](#) = tuning< Tuning\_Slow\_Mult\_Matrix\_Threshold, Tuning\_Default\_Div\_Max\_Steps, Tuning\_Default\_CR\_Sqrt\_Max\_Steps, Tuning\_Default\_DB\_Sqrt\_Max\_Steps, Tuning\_Default\_Log\_Max\_↵\_Outer\_Steps, Tuning\_Default\_Log\_Max\_Inner\_Steps, Tuning\_Slow\_Basis\_Max\_Count, Tuning\_Slow\_↵\_Fast\_Size\_Threshold, Tuning\_Slow\_Inv\_Fast\_Dim\_Threshold, Tuning\_Slow\_Products\_Size\_Threshold, Tuning\_Default\_Denom\_Different\_Bits, Tuning\_Default\_Extra\_Different\_Bits, Tuning\_Default\_Function\_↵\_Precision >
- using [glucat::tuning\\_naive](#) = tuning< Tuning\_Naive\_Mult\_Matrix\_Threshold, Tuning\_Default\_Div\_Max\_↵\_Steps, Tuning\_Default\_CR\_Sqrt\_Max\_Steps, Tuning\_Default\_DB\_Sqrt\_Max\_Steps, Tuning\_Default\_Log\_↵\_Max\_Outer\_Steps, Tuning\_Default\_Log\_Max\_Inner\_Steps, Tuning\_Naive\_Basis\_Max\_Count, Tuning\_↵\_Naive\_Fast\_Size\_Threshold, Tuning\_Naive\_Inv\_Fast\_Dim\_Threshold, Tuning\_Default\_Products\_Size\_↵\_Threshold, Tuning\_Default\_Denom\_Different\_Bits, Tuning\_Default\_Extra\_Different\_Bits, Tuning\_Default\_↵\_Function\_Precision >
- using [glucat::tuning\\_fast](#) = tuning< Tuning\_Fast\_Mult\_Matrix\_Threshold, Tuning\_Fast\_Div\_Max\_Steps, Tuning\_Fast\_CR\_Sqrt\_Max\_Steps, Tuning\_Fast\_DB\_Sqrt\_Max\_Steps, Tuning\_Fast\_Log\_Max\_Outer\_↵\_Steps, Tuning\_Fast\_Log\_Max\_Inner\_Steps, Tuning\_Fast\_Basis\_Max\_Count, Tuning\_Fast\_Fast\_Size\_↵\_Threshold, Tuning\_Fast\_Inv\_Fast\_Dim\_Threshold, Tuning\_Fast\_Products\_Size\_Threshold, Tuning\_↵\_Default\_Denom\_Different\_Bits, Tuning\_Default\_Extra\_Different\_Bits, Tuning\_Default\_Function\_Precision >

## Variables

- const unsigned int [glucat::Tuning\\_Int\\_Digits](#) = std::numeric\_limits<int>::digits
- const unsigned int [glucat::Tuning\\_Max\\_Threshold](#) = 1 << Tuning\_Int\_Digits
- const unsigned int [glucat::Tuning\\_Slow\\_Mult\\_Matrix\\_Threshold](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Slow\\_Basis\\_Max\\_Count](#) = 0
- const unsigned int [glucat::Tuning\\_Slow\\_Fast\\_Size\\_Threshold](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Slow\\_Inv\\_Fast\\_Dim\\_Threshold](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Slow\\_Products\\_Size\\_Threshold](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Naive\\_Mult\\_Matrix\\_Threshold](#) = 0
- const unsigned int [glucat::Tuning\\_Naive\\_Basis\\_Max\\_Count](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Naive\\_Fast\\_Size\\_Threshold](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Naive\\_Inv\\_Fast\\_Dim\\_Threshold](#) = Tuning\_Max\_Threshold
- const unsigned int [glucat::Tuning\\_Fast\\_Mult\\_Matrix\\_Threshold](#) = 0
- const unsigned int [glucat::Tuning\\_Fast\\_Div\\_Max\\_Steps](#) = 0
- const unsigned int [glucat::Tuning\\_Fast\\_CR\\_Sqrt\\_Max\\_Steps](#) = 256
- const unsigned int [glucat::Tuning\\_Fast\\_DB\\_Sqrt\\_Max\\_Steps](#) = 256
- const unsigned int [glucat::Tuning\\_Fast\\_Log\\_Max\\_Outer\\_Steps](#) = 16
- const unsigned int [glucat::Tuning\\_Fast\\_Log\\_Max\\_Inner\\_Steps](#) = 8
- const unsigned int [glucat::Tuning\\_Fast\\_Basis\\_Max\\_Count](#) = 1
- const unsigned int [glucat::Tuning\\_Fast\\_Fast\\_Size\\_Threshold](#) = 0
- const unsigned int [glucat::Tuning\\_Fast\\_Inv\\_Fast\\_Dim\\_Threshold](#) = 0
- const unsigned int [glucat::Tuning\\_Fast\\_Products\\_Size\\_Threshold](#) = 0

## 7.28 pyclical/glucat.pxd File Reference

### Namespaces

- [glucat](#)

## 7.29 pyclical/PyClical.h File Reference

```
#include "glucat/glucat_config.h"
#include "glucat/glucat.h"
#include "glucat/glucat_imp.h"
#include <iostream>
#include <sstream>
#include <iomanip>
#include <limits>
```

Include dependency graph for PyClical.h:



### Namespaces

- [cga3](#)

*Definitions for 3D Conformal Geometric Algebra [DL].*

## Typedefs

- using `String` = `std::string`
- using `IndexSet` = `index_set< lo_ndx, hi_ndx >`
- using `scalar_t` = `double`
- using `Clifford` = `matrix_multi< scalar_t, lo_ndx, hi_ndx, tuning_promoted >`

## Functions

- `template<typename Scalar_T >`  
`PyObject * PyFloat_FromDouble (Scalar_T v)`
- `template<typename Index_Set_T >`  
`String index_set_to_repr (const Index_Set_T &ist)`  
*The "official" string representation of Index\_Set\_T ist.*
- `template<typename Index_Set_T >`  
`String index_set_to_str (const Index_Set_T &ist)`  
*The "informal" string representation of Index\_Set\_T ist.*
- `template<typename Multivector_T >`  
`String clifford_to_repr (const Multivector_T &mv)`  
*The "official" string representation of Multivector\_T mv.*
- `template<typename Multivector_T >`  
`String clifford_to_str (const Multivector_T &mv)`  
*The "informal" string representation of Multivector\_T mv.*
- `template<typename Multivector_T >`  
`Multivector_T cga3::cga3 (const Multivector_T &x)`  
*Convert Euclidean 3D vector to Conformal Geometric Algebra null vector [DL (10.50)].*
- `template<typename Multivector_T >`  
`Multivector_T cga3::cga3std (const Multivector_T &X)`  
*Convert CGA3 null vector to standard Conformal Geometric Algebra null vector [DL (10.52)].*
- `template<typename Multivector_T >`  
`Multivector_T cga3::agc3 (const Multivector_T &X)`  
*Convert CGA3 null vector to Euclidean 3D vector [DL (10.50)].*

## Variables

- `String glucat_package_version` = `GLUCAT_PACKAGE_VERSION`
- `const index_t lo_ndx` = `DEFAULT_LO`
- `const index_t hi_ndx` = `DEFAULT_HI`
- `const scalar_t epsilon` = `std::numeric_limits<scalar_t>::epsilon()`

### 7.29.1 Typedef Documentation

#### 7.29.1.1 Clifford

```
using Clifford = matrix_multi<scalar_t, lo_ndx, hi_ndx, tuning_promoted>
```

Definition at line 148 of file PyClical.h.

### 7.29.1.2 IndexSet

```
using IndexSet = index_set<lo_ndx, hi_ndx>
```

Definition at line 145 of file PyClical.h.

### 7.29.1.3 scalar\_t

```
using scalar_t = double
```

Definition at line 147 of file PyClical.h.

### 7.29.1.4 String

```
using String = std::string
```

Definition at line 51 of file PyClical.h.

## 7.29.2 Function Documentation

### 7.29.2.1 clifford\_to\_repr()

```
template<typename Multivector_T >
String clifford_to_repr (
    const Multivector_T & mv ) [inline]
```

The "official" string representation of Multivector\_T mv.

Definition at line 75 of file PyClical.h.

Referenced by PyClical.clifford::\_\_repr\_\_().

### 7.29.2.2 clifford\_to\_str()

```
template<typename Multivector_T >
String clifford_to_str (
    const Multivector_T & mv ) [inline]
```

The "informal" string representation of Multivector\_T mv.

Definition at line 86 of file PyClical.h.

References glucat::abs(), PyClical::e(), and epsilon.

Referenced by PyClical.clifford::\_\_str\_\_().

### 7.29.2.3 index\_set\_to\_repr()

```
template<typename Index_Set_T >
String index_set_to_repr (
    const Index_Set_T & ist ) [inline]
```

The "official" string representation of Index\_Set\_T ist.

Definition at line 57 of file PyClical.h.

References PyClical::ist.

Referenced by PyClical.index\_set::\_\_repr\_\_().

### 7.29.2.4 index\_set\_to\_str()

```
template<typename Index_Set_T >
String index_set_to_str (
    const Index_Set_T & ist ) [inline]
```

The "informal" string representation of Index\_Set\_T ist.

Definition at line 66 of file PyClical.h.

References PyClical::ist.

Referenced by PyClical.index\_set::\_\_str\_\_().

### 7.29.2.5 PyFloat\_FromDouble()

```
template<typename Scalar_T >
PyObject* PyFloat_FromDouble (
    Scalar_T v ) [inline]
```

Create a PyFloatObject object from Scalar\_T v. Needed because Scalar\_T might not be the same as double.

Definition at line 45 of file PyClical.h.

## 7.29.3 Variable Documentation

### 7.29.3.1 epsilon

```
const scalar_t epsilon = std::numeric_limits<scalar_t>::epsilon()
```

Definition at line 150 of file PyClical.h.

Referenced by `glucat::cascade_log()`, `glucat::matrix::classify_eigenvalues()`, `clifford_to_str()`, and `glucat::error_squared_tol()`.

### 7.29.3.2 glucat\_package\_version

```
String glucat_package_version = GLUCAT_PACKAGE_VERSION
```

Definition at line 53 of file PyClical.h.

### 7.29.3.3 hi\_ndx

```
const index_t hi_ndx = DEFAULT_HI
```

Definition at line 144 of file PyClical.h.

### 7.29.3.4 lo\_ndx

```
const index_t lo_ndx = DEFAULT_LO
```

Definition at line 143 of file PyClical.h.

## 7.30 pyclical/PyClical.pxd File Reference

### Namespaces

- [PyClical](#)

## 7.31 pyclical/PyClical.pyx File Reference

### Classes

- class [PyClical.index\\_set](#)
- class [PyClical.index\\_set](#)
- class [PyClical.clifford](#)
- class [PyClical.clifford](#)

## Namespaces

- [PyClical](#)

## Functions

- [def PyClical.index\\_set\\_hidden\\_doctests \(\)](#)
- [def PyClical.clifford\\_hidden\\_doctests \(\)](#)
- [def PyClical.e \(obj\)](#)
- [def PyClical.istpq \(p, q\)](#)
- [def PyClical.\\_test \(\)](#)

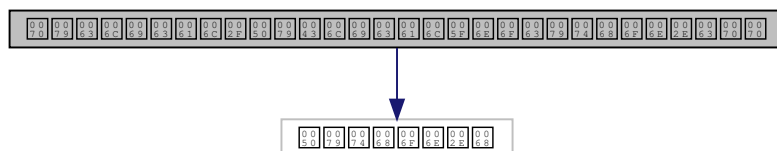
## Variables

- [PyClical.\\_\\_version\\_\\_](#) = `str(glucat_package_version,'utf-8')`
- [PyClical.lhs](#)
- [PyClical.rhs](#)
- [PyClical.threshold](#) = `error_squared_tol(rhs)` if threshold is None else threshold
- [PyClical.None](#)
- [PyClical.tol](#) = `error_squared_tol(rhs)` if tol is None else tol
- [PyClical.obj](#)
- [PyClical.i](#)
- [PyClical.ixt](#)
- [PyClical.fill](#)
- [PyClical.scalar\\_epsilon](#) = `epsilon`
- `float` [PyClical.pi](#) = `atan(clifford(1.0)) * 4.0`
- `float` [PyClical.tau](#) = `atan(clifford(1.0)) * 8.0`
- [PyClical.cl](#) = `clifford`
- [PyClical.ist](#) = `index_set`
- `def` [PyClical.ninf3](#) = `e(4) + e(-1)`
- `def` [PyClical.nbar3](#) = `e(4) - e(-1)`

## 7.32 pyclical/PyClical\_nocython.cpp File Reference

```
#include "Python.h"
```

Include dependency graph for PyClical\_nocython.cpp:



## Macros

- `#define` [PY\\_SSIZE\\_T\\_CLEAN](#)

### 7.32.1 Macro Definition Documentation

#### 7.32.1.1 PY\_SSIZE\_T\_CLEAN

```
#define PY_SSIZE_T_CLEAN
```

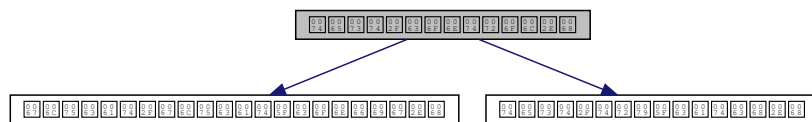
Definition at line 23 of file PyClical\_nocython.cpp.

### 7.33 test/control.h File Reference

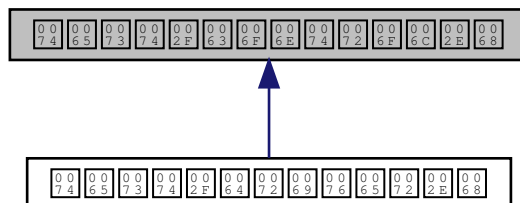
```
#include "glucat/glucat_config.h"
```

```
#include "test/try_catch.h"
```

Include dependency graph for control.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [glucat::control\\_t](#)  
*Parameters to control tests.*

### Namespaces

- [glucat](#)



## 7.34 test/driver.h File Reference

```
#include "glucat/glucat.h"
#include "glucat/glucat_imp.h"
#include "test/tuning.h"
#include "test/try_catch.h"
#include "test/control.h"
#include <stdio>
Include dependency graph for driver.h:
```



## 7.35 test/timing.h File Reference

### Namespaces

- [glucat](#)
- [glucat::timing](#)

### Functions

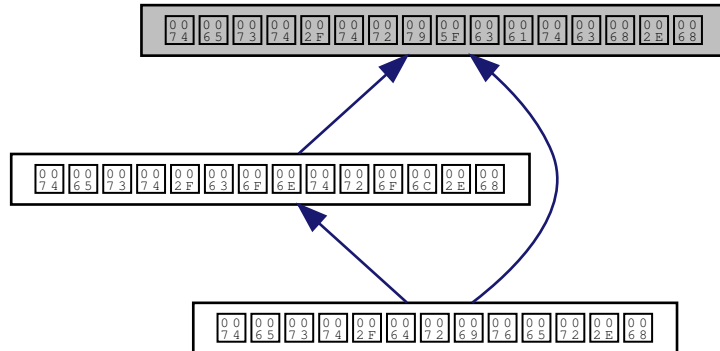
- static double [glucat::timing::elapsed](#) (clock\_t cpu\_time)  
*Elapsed time in milliseconds.*

### Variables

- const double [glucat::timing::MS\\_PER\\_SEC](#) = 1000.0  
*Timing constant: milliseconds per second.*
- const double [glucat::timing::MS\\_PER\\_CLOCK](#) = MS\_PER\_SEC / double(CLOCKS\_PER\_SEC)  
*Timing constant: milliseconds per clock.*
- const int [glucat::timing::EXTRA\\_TRIALS](#) = 2  
*Timing constant: trial expansion factor.*

## 7.36 test/try\_catch.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [glucat](#)

### Typedefs

- typedef int(\* [glucat::intfn](#)) ()  
*For exception catching: pointer to function returning int.*
- typedef int(\* [glucat::intintfn](#)) (int)  
*For exception catching: pointer to function of int returning int.*

### Functions

- int [glucat::try\\_catch](#) (intfn f)  
*Exception catching for functions returning int.*
- int [glucat::try\\_catch](#) (intintfn f, int arg)  
*Exception catching for functions of int returning int.*

# Index

- `_GLUCAT_CLIFFORD_ALGEBRA_OPERATIONS`
  - `clifford_algebra.h`, [272](#)
- `_GLUCAT_CTAssert`
  - `global.h`, [290](#)
  - `glucat`, [24](#), [25](#)
  - `glucat/tuning.h`, [316](#)
- `_GLUCAT_HASH_N`
  - `framed_multi_imp.h`, [286](#)
- `_GLUCAT_HASH_SIZE_T`
  - `framed_multi_imp.h`, [286](#)
- `_GLUCAT_ISINF`
  - `portability.h`, [310](#)
- `_GLUCAT_ISNAN`
  - `portability.h`, [310](#)
- `__add__`
  - `PyClical::clifford`, [96](#)
- `__and__`
  - `PyClical::clifford`, [96](#)
  - `PyClical::index_set`, [180](#)
- `__call__`
  - `PyClical::clifford`, [96](#)
- `__cinit__`
  - `PyClical::clifford`, [97](#)
  - `PyClical::index_set`, [181](#)
- `__contains__`
  - `PyClical::clifford`, [97](#)
  - `PyClical::index_set`, [181](#)
- `__dealloc__`
  - `PyClical::clifford`, [98](#)
  - `PyClical::index_set`, [182](#)
- `__getitem__`
  - `PyClical::clifford`, [98](#)
  - `PyClical::index_set`, [182](#)
- `__iadd__`
  - `PyClical::clifford`, [98](#)
- `__iand__`
  - `PyClical::clifford`, [99](#)
  - `PyClical::index_set`, [182](#)
- `__idiv__`
  - `PyClical::clifford`, [99](#)
- `__imod__`
  - `PyClical::clifford`, [99](#)
- `__imul__`
  - `PyClical::clifford`, [100](#)
- `__invert__`
  - `PyClical::index_set`, [183](#)
- `__ior__`
  - `PyClical::clifford`, [100](#)
  - `PyClical::index_set`, [183](#)
- `__isub__`
  - `PyClical::clifford`, [100](#)
- `__iter__`
  - `PyClical::clifford`, [101](#)
  - `PyClical::index_set`, [183](#)
- `__ixor__`
  - `PyClical::clifford`, [101](#)
  - `PyClical::index_set`, [184](#)
- `__mod__`
  - `PyClical::clifford`, [101](#)
- `__mul__`
  - `PyClical::clifford`, [102](#)
- `__neg__`
  - `PyClical::clifford`, [102](#)
- `__or__`
  - `PyClical::clifford`, [102](#)
  - `PyClical::index_set`, [184](#)
- `__pos__`
  - `PyClical::clifford`, [103](#)
- `__pow__`
  - `PyClical::clifford`, [103](#)
- `__repr__`
  - `PyClical::clifford`, [103](#)
  - `PyClical::index_set`, [184](#)
- `__richcmp__`
  - `PyClical::clifford`, [104](#)
  - `PyClical::index_set`, [185](#)
- `__setitem__`
  - `PyClical::index_set`, [185](#)
- `__str__`
  - `PyClical::clifford`, [104](#)
  - `PyClical::index_set`, [185](#)
- `__sub__`
  - `PyClical::clifford`, [104](#)
- `__truediv__`
  - `PyClical::clifford`, [105](#)
- `__version__`
  - `PyClical`, [85](#)
- `__xor__`
  - `PyClical::clifford`, [105](#)
  - `PyClical::index_set`, [186](#)
- `__test__`
  - `PyClical`, [82](#)
- `~basis_table`
  - `glucat::basis_table`, [92](#)
- `~clifford_algebra`
  - `glucat::clifford_algebra`, [116](#)
- `~control_t`
  - `glucat::control_t`, [129](#)

- ~framed\_multi
  - glucat::framed\_multi, 144
- ~generator\_table
  - glucat::gen::generator\_table, 157
- ~glucat\_error
  - glucat::glucat\_error, 161
- ~matrix\_multi
  - glucat::matrix\_multi, 196
- ~random\_generator
  - glucat::random\_generator, 250
- ~reference
  - glucat::index\_set::reference, 254
- ~var\_term
  - glucat::framed\_multi::var\_term, 263
- abs
  - glucat, 25
  - glucat::numeric\_traits, 212
  - PyClical::clifford, 105
- acos
  - glucat, 25
  - glucat::numeric\_traits, 212
- acosh
  - glucat, 26
- agc3
  - cga3, 9
- approx\_equal
  - glucat, 26, 27
- array
  - pade::pade\_log\_denom, 223
  - pade::pade\_log\_denom< dd\_real >, 225
  - pade::pade\_log\_denom< float >, 226
  - pade::pade\_log\_denom< long double >, 227
  - pade::pade\_log\_denom< qd\_real >, 228
  - pade::pade\_log\_number, 230
  - pade::pade\_log\_number< dd\_real >, 231
  - pade::pade\_log\_number< float >, 232
  - pade::pade\_log\_number< long double >, 233
  - pade::pade\_log\_number< qd\_real >, 234
  - pade::pade\_sqrt\_denom, 236
  - pade::pade\_sqrt\_denom< dd\_real >, 237
  - pade::pade\_sqrt\_denom< float >, 238
  - pade::pade\_sqrt\_denom< long double >, 239
  - pade::pade\_sqrt\_denom< qd\_real >, 240
  - pade::pade\_sqrt\_number, 242
  - pade::pade\_sqrt\_number< dd\_real >, 243
  - pade::pade\_sqrt\_number< float >, 244
  - pade::pade\_sqrt\_number< long double >, 245
  - pade::pade\_sqrt\_number< qd\_real >, 246
- asin
  - glucat, 27
  - glucat::numeric\_traits, 213
- asinh
  - glucat, 28
- atan
  - glucat, 28, 29
  - glucat::numeric\_traits, 213
- atanh
  - glucat, 29
- BITS\_PER\_SET\_VALUE
  - glucat, 67
- BOOST\_STATIC\_ASSERT
  - glucat::index\_set, 169
- basis
  - glucat::basis\_table, 92
- basis\_element
  - glucat::matrix\_multi, 201
- basis\_matrix\_t
  - glucat::matrix\_multi, 194
- basis\_table
  - glucat::basis\_table, 92
- bitset\_t
  - glucat::index\_set, 167
- call
  - glucat::control\_t, 129
- cascade\_log
  - glucat, 30
- catch\_exceptions
  - glucat::control\_t, 129
- centre\_pm4\_qp4
  - glucat::framed\_multi, 149
- centre\_pp4\_qm4
  - glucat::framed\_multi, 149
- centre\_qp1\_pm1
  - glucat::framed\_multi, 149
- cga3, 9
  - agc3, 9
  - cga3, 9
  - cga3std, 10
- cga3std
  - cga3, 10
- check\_complex
  - glucat, 30
- cl
  - PyClical, 86
- classify\_eigenvalues
  - glucat::matrix, 74
- classname
  - glucat::clifford\_algebra, 117
  - glucat::error, 137
  - glucat::framed\_multi, 149
  - glucat::framed\_multi::var\_term, 264
  - glucat::glucat\_error, 161
  - glucat::index\_set, 169
  - glucat::matrix\_multi, 202
- Clifford
  - PyClical.h, 319
- clifford\_algebra.h
  - \_GLUCAT\_CLIFFORD\_ALGEBRA\_OPERATIO↵NS, 272
- clifford\_exp
  - glucat, 30
- clifford\_hidden\_doctests
  - PyClical, 82
- clifford\_to\_repr
  - PyClical.h, 320
- clifford\_to\_str

- PyClical.h, 320
- compare
  - glucat, 31
  - glucat::index\_set, 178
- complexifier
  - glucat, 31
- conj
  - glucat, 31
  - glucat::clifford\_algebra, 117
  - glucat::numeric\_traits, 213
  - PyClical::clifford, 106
- const\_iterator
  - glucat::framed\_multi, 141
- control
  - glucat::control\_t, 130
- control\_t
  - glucat::control\_t, 128, 129
- cos
  - glucat, 32
  - glucat::numeric\_traits, 213
- cosh
  - glucat, 32
  - glucat::numeric\_traits, 214
- count
  - glucat::index\_set, 170
  - PyClical::index\_set, 186
- count\_neg
  - glucat::index\_set, 170
  - PyClical::index\_set, 186
- count\_pos
  - glucat::index\_set, 170
  - PyClical::index\_set, 187
- cr\_sqrt
  - glucat, 33
- crd\_of\_mult
  - glucat, 33
- DEFAULT\_HI
  - glucat, 67
- db\_sqrt
  - glucat, 34
- db\_step
  - glucat, 34
- default\_truncation
  - glucat::clifford\_algebra, 125
- denom
  - pade::pade\_log\_denom, 224
  - pade::pade\_log\_denom< dd\_real >, 225
  - pade::pade\_log\_denom< float >, 226
  - pade::pade\_log\_denom< long double >, 227
  - pade::pade\_log\_denom< qd\_real >, 228
  - pade::pade\_sqrt\_denom, 236
  - pade::pade\_sqrt\_denom< dd\_real >, 237
  - pade::pade\_sqrt\_denom< float >, 238
  - pade::pade\_sqrt\_denom< long double >, 239
  - pade::pade\_sqrt\_denom< qd\_real >, 240
- divide
  - glucat::framed\_multi, 150
- e
  - PyClical, 83
- EXTRA\_TRIALS
  - glucat::timing, 80
- eig\_case\_t
  - glucat::matrix, 74
- eigenvalues
  - glucat::matrix, 74
- elapsed
  - glucat::timing, 80
- elliptic
  - glucat, 34
- epsilon
  - PyClical.h, 321
- error
  - glucat::error, 136
- error\_squared
  - glucat, 35
- error\_squared\_tol
  - glucat, 35
- error\_t
  - glucat::framed\_multi, 141
  - glucat::index\_set, 167
  - glucat::matrix\_multi, 194
- even
  - glucat, 35
  - glucat::clifford\_algebra, 117
  - PyClical::clifford, 106
- exp
  - glucat, 36
  - glucat::framed\_multi, 152
  - glucat::numeric\_traits, 214
- fast
  - glucat, 36
  - glucat::framed\_multi, 150
- fast\_framed\_multi
  - glucat::framed\_multi, 150
  - glucat::matrix\_multi, 202
- fast\_matrix\_multi
  - glucat::framed\_multi, 150
  - glucat::matrix\_multi, 202
- fill
  - PyClical, 86
- flip
  - glucat::index\_set, 170, 171
  - glucat::index\_set::reference, 254
- fmod
  - glucat::numeric\_traits, 214
- fold
  - glucat::framed\_multi, 151
  - glucat::index\_set, 171
- folded\_dim
  - glucat, 37
- frame
  - glucat::clifford\_algebra, 117
  - PyClical::clifford, 106
- framed\_multi
  - glucat::framed\_multi, 144–148, 152

- glucat::matrix\_multi, 203
- framed\_multi\_imp.h
  - \_GLUCAT\_HASH\_N, 286
  - \_GLUCAT\_HASH\_SIZE\_T, 286
- framed\_multi\_t
  - glucat::framed\_multi, 141
  - glucat::matrix\_multi, 194
- framed\_pair\_t
  - glucat::framed\_multi, 141
- friend\_for\_private\_destructor
  - glucat::basis\_table, 93
  - glucat::control\_t, 131
  - glucat::gen::generator\_table, 159
  - glucat::random\_generator, 251
- GLUCAT\_HAVE\_CXX11
  - glucat\_config.h, 291
- GLUCAT\_HAVE\_INTTYPES\_H
  - glucat\_config.h, 291
- GLUCAT\_HAVE\_STDINT\_H
  - glucat\_config.h, 292
- GLUCAT\_HAVE\_STDIO\_H
  - glucat\_config.h, 292
- GLUCAT\_HAVE\_STDLIB\_H
  - glucat\_config.h, 292
- GLUCAT\_HAVE\_STRING\_H
  - glucat\_config.h, 292
- GLUCAT\_HAVE\_STRINGS\_H
  - glucat\_config.h, 292
- GLUCAT\_HAVE\_SYS\_STAT\_H
  - glucat\_config.h, 292
- GLUCAT\_HAVE\_SYS\_TYPES\_H
  - glucat\_config.h, 293
- GLUCAT\_HAVE\_UNISTD\_H
  - glucat\_config.h, 293
- GLUCAT\_PACKAGE\_BUGREPORT
  - glucat\_config.h, 293
- GLUCAT\_PACKAGE\_NAME
  - glucat\_config.h, 293
- GLUCAT\_PACKAGE\_STRING
  - glucat\_config.h, 293
- GLUCAT\_PACKAGE\_TARNAME
  - glucat\_config.h, 294
- GLUCAT\_PACKAGE\_URL
  - glucat\_config.h, 294
- GLUCAT\_PACKAGE\_VERSION
  - glucat\_config.h, 294
- GLUCAT\_PACKAGE
  - glucat\_config.h, 293
- GLUCAT\_STDC\_HEADERS
  - glucat\_config.h, 294
- GLUCAT\_VERSION
  - glucat\_config.h, 294
- gen\_from\_pm1\_qm1
  - glucat::gen::generator\_table, 157
- gen\_from\_pm4\_qp4
  - glucat::gen::generator\_table, 157
- gen\_from\_pp4\_qm4
  - glucat::gen::generator\_table, 157
- gen\_from\_qp1\_pm1
  - glucat::gen::generator\_table, 158
- gen\_vector
  - glucat::gen::generator\_table, 158
- generator
  - glucat::gen::generator\_table, 158
  - glucat::random\_generator, 250
- generator\_table
  - glucat::gen::generator\_table, 156, 157
- global.h
  - \_GLUCAT\_CTAssert, 290
- glucat, 10
  - \_GLUCAT\_CTAssert, 24, 25
  - abs, 25
  - acos, 25
  - acosh, 26
  - approx\_equal, 26, 27
  - asin, 27
  - asinh, 28
  - atan, 28, 29
  - atanh, 29
  - BITS\_PER\_SET\_VALUE, 67
  - cascade\_log, 30
  - check\_complex, 30
  - clifford\_exp, 30
  - compare, 31
  - complexifier, 31
  - conj, 31
  - cos, 32
  - cosh, 32
  - cr\_sqrt, 33
  - crd\_of\_mult, 33
  - DEFAULT\_HI, 67
  - db\_sqrt, 34
  - db\_step, 34
  - elliptic, 34
  - error\_squared, 35
  - error\_squared\_tol, 35
  - even, 35
  - exp, 36
  - fast, 36
  - folded\_dim, 37
  - imag, 37
  - index\_t, 23
  - intfn, 23
  - intintfn, 23
  - inv, 37
  - inverse\_gray, 37
  - inverse\_reversed\_gray, 38
  - involute, 38
  - l\_ln2, 67
  - l\_pi, 67
  - log, 38, 39
  - log2, 39
  - MS\_PER\_S, 67
  - matrix\_log, 40
  - matrix\_sqrt, 40
  - max\_abs, 40

- max\_pos, [41](#)
- min\_neg, [41](#)
- norm, [41](#)
- odd, [41](#)
- offset\_level, [42](#)
- operator &, [42–44](#)
- operator!=, [45](#)
- operator<<, [52](#), [53](#)
- operator>>, [53](#), [54](#)
- operator\*, [47](#), [48](#)
- operator^, [54](#), [55](#)
- operator+, [49](#)
- operator-, [50](#)
- operator/, [51](#), [52](#)
- operator%, [46](#)
- operator|, [55](#), [56](#)
- outer\_pow, [57](#)
- pade\_approx, [57](#)
- pade\_log, [57](#)
- pos\_mod, [58](#)
- pow, [58](#)
- pure, [59](#)
- quad, [59](#)
- real, [59](#)
- reframe, [59](#)
- reverse, [60](#)
- scalar, [60](#)
- set\_value\_t, [23](#)
- sign\_of\_square, [60](#)
- sin, [61](#)
- sinh, [61](#)
- sqrt, [62](#), [63](#)
- star, [63](#), [64](#)
- tan, [64](#)
- tanh, [65](#)
- to\_demote, [65](#)
- to\_promote, [65](#)
- try\_catch, [66](#)
- Tuning\_Fast\_Basis\_Max\_Count, [67](#)
- Tuning\_Fast\_CR\_Sqrt\_Max\_Steps, [68](#)
- Tuning\_Fast\_DB\_Sqrt\_Max\_Steps, [68](#)
- Tuning\_Fast\_Div\_Max\_Steps, [68](#)
- Tuning\_Fast\_Fast\_Size\_Threshold, [68](#)
- Tuning\_Fast\_Inv\_Fast\_Dim\_Threshold, [68](#)
- Tuning\_Fast\_Log\_Max\_Inner\_Steps, [68](#)
- Tuning\_Fast\_Log\_Max\_Outer\_Steps, [69](#)
- Tuning\_Fast\_Mult\_Matrix\_Threshold, [69](#)
- Tuning\_Fast\_Products\_Size\_Threshold, [69](#)
- Tuning\_Int\_Digits, [69](#)
- Tuning\_Max\_Threshold, [69](#)
- Tuning\_Naive\_Basis\_Max\_Count, [69](#)
- Tuning\_Naive\_Fast\_Size\_Threshold, [70](#)
- Tuning\_Naive\_Inv\_Fast\_Dim\_Threshold, [70](#)
- Tuning\_Naive\_Mult\_Matrix\_Threshold, [70](#)
- Tuning\_Slow\_Basis\_Max\_Count, [70](#)
- Tuning\_Slow\_Fast\_Size\_Threshold, [70](#)
- Tuning\_Slow\_Inv\_Fast\_Dim\_Threshold, [70](#)
- Tuning\_Slow\_Mult\_Matrix\_Threshold, [71](#)
- Tuning\_Slow\_Products\_Size\_Threshold, [71](#)
- tuning\_fast, [23](#)
- tuning\_naive, [24](#)
- tuning\_slow, [24](#)
- vector\_part, [66](#)
- glucat/clifford\_algebra.h, [265](#)
- glucat/clifford\_algebra\_imp.h, [272](#)
- glucat/errors.h, [279](#)
- glucat/errors\_imp.h, [280](#)
- glucat/framed\_multi.h, [281](#)
- glucat/framed\_multi\_imp.h, [283](#)
- glucat/generation.h, [286](#)
- glucat/generation\_imp.h, [287](#)
- glucat/global.h, [288](#)
- glucat/glucat.h, [290](#)
- glucat/glucat\_config.h, [291](#)
- glucat/glucat\_imp.h, [295](#)
- glucat/index\_set.h, [295](#)
- glucat/index\_set\_imp.h, [297](#)
- glucat/long\_double.h, [298](#)
- glucat/matrix.h, [299](#)
- glucat/matrix\_imp.h, [301](#)
- glucat/matrix\_multi.h, [303](#)
- glucat/matrix\_multi\_imp.h, [306](#)
- glucat/portability.h, [309](#)
- glucat/promotion.h, [311](#)
- glucat/qd.h, [312](#)
- glucat/random.h, [312](#)
- glucat/scalar.h, [313](#)
- glucat/scalar\_imp.h, [315](#)
- glucat/tuning.h, [316](#)
- \_GLUCAT\_CTAssert, [316](#)
- glucat::CTAssertion< bool >, [132](#)
- glucat::CTAssertion< true >, [132](#)
- glucat::basis\_table
  - ~basis\_table, [92](#)
  - basis, [92](#)
  - basis\_table, [92](#)
  - friend\_for\_private\_destructor, [93](#)
  - operator=, [93](#)
- glucat::basis\_table< Scalar\_T, LO, HI, Matrix\_T >, [91](#)
- glucat::bool\_to\_type< truth\_value >, [93](#)
- glucat::clifford\_algebra
  - ~clifford\_algebra, [116](#)
  - classname, [117](#)
  - conj, [117](#)
  - default\_truncation, [125](#)
  - even, [117](#)
  - frame, [117](#)
  - grade, [117](#)
  - index\_set\_t, [115](#)
  - inv, [118](#)
  - involute, [118](#)
  - isinf, [118](#)
  - isnan, [118](#)
  - max\_abs, [118](#)
  - multivector\_t, [115](#)
  - norm, [119](#)

- odd, [119](#)
- operator &=, [119](#)
- operator\*=, [120](#)
- operator^=, [122](#)
- operator(), [119](#)
- operator+=, [120](#)
- operator-, [120](#)
- operator-=, [121](#)
- operator/=, [121](#)
- operator==, [121](#), [122](#)
- operator%=: [119](#)
- operator[], [122](#)
- operator|=: [122](#)
- outer\_pow, [122](#)
- pair\_t, [116](#)
- pow, [123](#)
- pure, [123](#)
- quad, [123](#)
- reverse, [123](#)
- scalar, [123](#)
- scalar\_t, [116](#)
- truncated, [124](#)
- v\_hi, [125](#)
- v\_lo, [125](#)
- vector\_part, [124](#)
- vector\_t, [116](#)
- write, [124](#)
- glucat::clifford\_algebra< Scalar\_T, Index\_Set\_↔  
T, Multivector\_T >, [113](#)
- glucat::compare\_types< LHS\_T, RHS\_T >, [126](#)
- glucat::compare\_types< T, T >, [126](#)
- glucat::control\_t, [127](#)
  - ~control\_t, [129](#)
  - call, [129](#)
  - catch\_exceptions, [129](#)
  - control, [130](#)
  - control\_t, [128](#), [129](#)
  - friend\_for\_private\_destructor, [131](#)
  - m\_catch\_exceptions, [131](#)
  - m\_valid, [131](#)
  - m\_verbose\_output, [131](#)
  - operator=, [130](#)
  - valid, [130](#)
  - verbose, [130](#)
- glucat::error
  - classname, [137](#)
  - error, [136](#)
  - heading, [137](#)
  - print\_error\_msg, [137](#)
- glucat::error< Class\_T >, [135](#)
- glucat::framed\_multi
  - ~framed\_multi, [144](#)
  - centre\_pm4\_qp4, [149](#)
  - centre\_pp4\_qm4, [149](#)
  - centre\_qp1\_pm1, [149](#)
  - classname, [149](#)
  - const\_iterator, [141](#)
  - divide, [150](#)
  - error\_t, [141](#)
  - exp, [152](#)
  - fast, [150](#)
  - fast\_framed\_multi, [150](#)
  - fast\_matrix\_multi, [150](#)
  - fold, [151](#)
  - framed\_multi, [144–148](#), [152](#)
  - framed\_multi\_t, [141](#)
  - framed\_pair\_t, [141](#)
  - index\_set\_t, [142](#)
  - iterator, [142](#)
  - map\_t, [142](#)
  - matrix\_multi, [152](#)
  - matrix\_multi\_t, [142](#)
  - matrix\_t, [142](#)
  - multivector\_t, [143](#)
  - nbr\_terms, [151](#)
  - operator &, [153](#)
  - operator<<, [153](#), [154](#)
  - operator>>, [154](#)
  - operator\*, [153](#)
  - operator^, [154](#)
  - operator+=, [151](#)
  - operator/, [153](#)
  - operator%, [153](#)
  - operator|, [154](#)
  - random, [151](#)
  - scalar\_t, [143](#)
  - size\_type, [143](#)
  - sorted\_map\_t, [143](#)
  - star, [154](#)
  - term\_t, [143](#)
  - tune\_p, [144](#)
  - unfold, [152](#)
  - var\_term\_t, [144](#)
  - vector\_t, [144](#)
- glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >, [138](#)
- glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >↔
  - ::hash\_size\_t, [162](#)
- glucat::framed\_multi< Scalar\_T, LO, HI, Tune\_P >↔
  - ::var\_term, [261](#)
- glucat::framed\_multi::hash\_size\_t
  - hash\_size\_t, [162](#)
  - n, [163](#)
  - operator(), [163](#)
- glucat::framed\_multi::var\_term
  - ~var\_term, [263](#)
  - classname, [264](#)
  - operator\*=, [264](#)
  - var\_pair\_t, [263](#)
  - var\_term, [263](#)
- glucat::gen, [71](#)
  - offset\_to\_super, [72](#)
  - signature\_t, [71](#)
- glucat::gen::generator\_table
  - ~generator\_table, [157](#)
  - friend\_for\_private\_destructor, [159](#)
  - gen\_from\_pm1\_qm1, [157](#)



- gen\_from\_pm4\_qp4, 157
- gen\_from\_pp4\_qm4, 157
- gen\_from\_qp1\_pm1, 158
- gen\_vector, 158
- generator, 158
- generator\_table, 156, 157
- operator(), 158
- operator=, 159
- glucat::gen::generator\_table< Matrix\_T >, 155
- glucat::glucat\_error, 160
  - ~glucat\_error, 161
  - classname, 161
  - glucat\_error, 161
  - heading, 161
  - name, 162
  - print\_error\_msg, 161
- glucat::index\_set
  - BOOST\_STATIC\_ASSERT, 169
  - bitset\_t, 167
  - classname, 169
  - compare, 178
  - count, 170
  - count\_neg, 170
  - count\_pos, 170
  - error\_t, 167
  - flip, 170, 171
  - fold, 171
  - hash\_fn, 171
  - index\_pair\_t, 167
  - index\_set, 168, 169
  - index\_set\_t, 167
  - is\_contiguous, 172
  - lex\_less\_than, 172
  - max, 172
  - min, 172
  - operator &, 178
  - operator &=, 173
  - operator!=, 173
  - operator<, 173
  - operator~, 175
  - operator^, 178
  - operator^=, 174
  - operator==, 173
  - operator[], 174
  - operator|, 178
  - operator|=, 174
  - reference, 178
  - reset, 175
  - set, 175, 176
  - sign\_of\_mult, 176
  - sign\_of\_square, 176
  - test, 177
  - unfold, 177
  - v\_hi, 179
  - v\_lo, 179
  - value\_of\_fold, 177
- glucat::index\_set< LO, HI >, 164
- glucat::index\_set< LO, HI >::reference, 253
- glucat::index\_set::reference
  - ~reference, 254
  - flip, 254
  - index\_set, 256
  - m\_idx, 256
  - m\_pst, 256
  - operator bool, 255
  - operator~, 256
  - operator=, 255
  - operator==, 255
  - reference, 254
- glucat::index\_set\_hash
  - index\_set\_t, 190
  - operator(), 190
- glucat::index\_set\_hash< LO, HI >, 189
- glucat::matrix, 72
  - classify\_eigenvalues, 74
  - eig\_case\_t, 74
  - eigenvalues, 74
  - inner, 74
  - isinf, 75
  - isnan, 75
  - kron, 75
  - mono\_kron, 75
  - mono\_prod, 76
  - nnz, 76
  - nork, 76
  - nork\_range, 77
  - norm\_frob2, 77
  - prod, 77
  - signed\_perm\_nork, 78
  - sparse\_prod, 78
  - to\_lapack, 78
  - trace, 79
  - unit, 79
- glucat::matrix::eig\_genus
  - m\_eig\_case, 134
  - m\_is\_singular, 134
  - m\_safe\_arg, 135
  - Scalar\_T, 134
- glucat::matrix::eig\_genus< Matrix\_T >, 133
- glucat::matrix\_multi
  - ~matrix\_multi, 196
  - basis\_element, 201
  - basis\_matrix\_t, 194
  - classname, 202
  - error\_t, 194
  - fast\_framed\_multi, 202
  - fast\_matrix\_multi, 202
  - framed\_multi, 203
  - framed\_multi\_t, 194
  - index\_set\_t, 194
  - m\_frame, 207
  - m\_matrix, 207
  - matrix\_index\_t, 194
  - matrix\_log, 204
  - matrix\_multi, 196–201, 204
  - matrix\_multi\_t, 195

- matrix\_sqrt, [204](#)
- matrix\_t, [195](#)
- multivector\_t, [195](#)
- operator &, [204](#)
- operator<<, [205](#)
- operator>>, [205](#)
- operator\*, [205](#)
- operator^, [206](#)
- operator+=, [202](#)
- operator/, [205](#)
- operator=, [203](#)
- operator%, [205](#)
- operator|, [206](#)
- orientation\_t, [195](#)
- random, [203](#)
- reframe, [206](#)
- scalar\_t, [195](#)
- star, [206](#)
- term\_t, [196](#)
- tune\_p, [196](#)
- vector\_t, [196](#)
- glucat::matrix\_multi< Scalar\_T, LO, HI, Tune\_P >, [191](#)
- glucat::numeric\_traits
  - abs, [212](#)
  - acos, [212](#)
  - asin, [213](#)
  - atan, [213](#)
  - conj, [213](#)
  - cos, [213](#)
  - cosh, [214](#)
  - exp, [214](#)
  - fmod, [214](#)
  - imag, [214](#)
  - isInf, [215](#)
  - isNaN\_or\_isInf, [216](#)
  - isNaN, [215](#), [216](#)
  - ln\_2, [216](#), [217](#)
  - log, [217](#)
  - log2, [217](#)
  - NaN, [217](#)
  - pi, [218](#)
  - pow, [218](#)
  - real, [218](#)
  - sin, [219](#)
  - sinh, [219](#)
  - sqrt, [219](#)
  - tan, [219](#)
  - tanh, [220](#)
  - to\_double, [220](#)
  - to\_int, [220](#)
  - to\_scalar\_t, [220–222](#)
- glucat::numeric\_traits< Scalar\_T >, [210](#)
- glucat::numeric\_traits< Scalar\_T >::demoted<>, [132](#)
- glucat::numeric\_traits< Scalar\_T >::promoted<>, [247](#)
- glucat::numeric\_traits::demoted
  - type, [133](#)
- glucat::numeric\_traits::promoted
  - type, [248](#)
- glucat::random\_generator
  - ~random\_generator, [250](#)
  - friend\_for\_private\_destructor, [251](#)
  - generator, [250](#)
  - normal, [250](#)
  - normal\_dist, [251](#)
  - operator=, [251](#)
  - random\_generator, [250](#)
  - seed, [252](#)
  - uint\_gen, [252](#)
  - uniform, [251](#)
  - uniform\_dist, [252](#)
- glucat::random\_generator< Scalar\_T >, [249](#)
- glucat::sorted\_range
  - map\_t, [257](#)
  - sorted\_begin, [258](#)
  - sorted\_end, [259](#)
  - sorted\_iterator, [258](#)
  - sorted\_map\_t, [258](#)
  - sorted\_range, [258](#)
- glucat::sorted\_range< Map\_T, Sorted\_Map\_T >, [257](#)
- glucat::sorted\_range< Sorted\_Map\_T, Sorted\_Map\_T >, [259](#)
  - map\_t, [260](#)
  - sorted\_begin, [260](#)
  - sorted\_end, [261](#)
  - sorted\_iterator, [260](#)
  - sorted\_map\_t, [260](#)
  - sorted\_range, [260](#)
- glucat::timing, [79](#)
  - EXTRA\_TRIALS, [80](#)
  - elapsed, [80](#)
  - MS\_PER\_CLOCK, [80](#)
  - MS\_PER\_SEC, [80](#)
- glucat\_config.h
  - GLUCAT\_HAVE\_CXX11, [291](#)
  - GLUCAT\_HAVE\_INTTYPES\_H, [291](#)
  - GLUCAT\_HAVE\_STDINT\_H, [292](#)
  - GLUCAT\_HAVE\_STDIO\_H, [292](#)
  - GLUCAT\_HAVE\_STDLIB\_H, [292](#)
  - GLUCAT\_HAVE\_STRING\_H, [292](#)
  - GLUCAT\_HAVE\_STRINGS\_H, [292](#)
  - GLUCAT\_HAVE\_SYS\_STAT\_H, [292](#)
  - GLUCAT\_HAVE\_SYS\_TYPES\_H, [293](#)
  - GLUCAT\_HAVE\_UNISTD\_H, [293](#)
  - GLUCAT\_PACKAGE\_BUGREPORT, [293](#)
  - GLUCAT\_PACKAGE\_NAME, [293](#)
  - GLUCAT\_PACKAGE\_STRING, [293](#)
  - GLUCAT\_PACKAGE\_TARNAME, [294](#)
  - GLUCAT\_PACKAGE\_URL, [294](#)
  - GLUCAT\_PACKAGE\_VERSION, [294](#)
  - GLUCAT\_PACKAGE, [293](#)
  - GLUCAT\_STDC\_HEADERS, [294](#)
  - GLUCAT\_VERSION, [294](#)
- glucat\_error
  - glucat::glucat\_error, [161](#)
- glucat\_package\_version
  - PyClical.h, [322](#)

- grade
  - glucat::clifford\_algebra, 117
- hash\_fn
  - glucat::index\_set, 171
  - PyClical::index\_set, 187
- hash\_size\_t
  - glucat::framed\_multi::hash\_size\_t, 162
- heading
  - glucat::error, 137
  - glucat::glucat\_error, 161
- hi\_ndx
  - PyClical.h, 322
- i
  - PyClical, 86
- imag
  - glucat, 37
  - glucat::numeric\_traits, 214
- index\_pair\_t
  - glucat::index\_set, 167
- index\_set
  - glucat::index\_set, 168, 169
  - glucat::index\_set::reference, 256
- index\_set\_hidden\_doctests
  - PyClical, 84
- index\_set\_t
  - glucat::clifford\_algebra, 115
  - glucat::framed\_multi, 142
  - glucat::index\_set, 167
  - glucat::index\_set\_hash, 190
  - glucat::matrix\_multi, 194
- index\_set\_to\_repr
  - PyClical.h, 320
- index\_set\_to\_str
  - PyClical.h, 321
- index\_t
  - glucat, 23
- IndexSet
  - PyClical.h, 319
- inner
  - glucat::matrix, 74
- instance
  - PyClical::clifford, 112
  - PyClical::index\_set, 189
- intfn
  - glucat, 23
- intintfn
  - glucat, 23
- inv
  - glucat, 37
  - glucat::clifford\_algebra, 118
  - PyClical::clifford, 107
- inverse\_gray
  - glucat, 37
- inverse\_reversed\_gray
  - glucat, 38
- involute
  - glucat, 38
- glucat::clifford\_algebra, 118
  - PyClical::clifford, 107
- is\_contiguous
  - glucat::index\_set, 172
- isInf
  - glucat::numeric\_traits, 215
- isNaN\_or\_isInf
  - glucat::numeric\_traits, 216
- isNaN
  - glucat::numeric\_traits, 215, 216
- isinf
  - glucat::clifford\_algebra, 118
  - glucat::matrix, 75
  - PyClical::clifford, 107
- isnan
  - glucat::clifford\_algebra, 118
  - glucat::matrix, 75
  - PyClical::clifford, 108
- ist
  - PyClical, 86
- istpq
  - PyClical, 85
- iterator
  - glucat::framed\_multi, 142
- ixt
  - PyClical, 86
- kron
  - glucat::matrix, 75
- l\_ln2
  - glucat, 67
- l\_pi
  - glucat, 67
- lex\_less\_than
  - glucat::index\_set, 172
- lhs
  - PyClical, 87
- ln\_2
  - glucat::numeric\_traits, 216, 217
- lo\_ndx
  - PyClical.h, 322
- log
  - glucat, 38, 39
  - glucat::numeric\_traits, 217
- log2
  - glucat, 39
  - glucat::numeric\_traits, 217
- m\_catch\_exceptions
  - glucat::control\_t, 131
- m\_eig\_case
  - glucat::matrix::eig\_genus, 134
- m\_frame
  - glucat::matrix\_multi, 207
- m\_idx
  - glucat::index\_set::reference, 256
- m\_is\_singular
  - glucat::matrix::eig\_genus, 134

- m\_matrix
  - glucat::matrix\_multi, 207
- m\_pst
  - glucat::index\_set::reference, 256
- m\_safe\_arg
  - glucat::matrix::eig\_genus, 135
- m\_valid
  - glucat::control\_t, 131
- m\_verbose\_output
  - glucat::control\_t, 131
- MS\_PER\_CLOCK
  - glucat::timing, 80
- MS\_PER\_SEC
  - glucat::timing, 80
- MS\_PER\_S
  - glucat, 67
- map\_t
  - glucat::framed\_multi, 142
  - glucat::sorted\_range, 257
  - glucat::sorted\_range< Sorted\_Map\_T, Sorted\_Map\_T >, 260
- matrix\_index\_t
  - glucat::matrix\_multi, 194
- matrix\_log
  - glucat, 40
  - glucat::matrix\_multi, 204
- matrix\_multi
  - glucat::framed\_multi, 152
  - glucat::matrix\_multi, 196–201, 204
- matrix\_multi\_t
  - glucat::framed\_multi, 142
  - glucat::matrix\_multi, 195
- matrix\_sqrt
  - glucat, 40
  - glucat::matrix\_multi, 204
- matrix\_t
  - glucat::framed\_multi, 142
  - glucat::matrix\_multi, 195
- max
  - glucat::index\_set, 172
  - PyClical::index\_set, 187
- max\_abs
  - glucat, 40
  - glucat::clifford\_algebra, 118
  - PyClical::clifford, 108
- max\_pos
  - glucat, 41
- min
  - glucat::index\_set, 172
  - PyClical::index\_set, 188
- min\_neg
  - glucat, 41
- mono\_kron
  - glucat::matrix, 75
- mono\_prod
  - glucat::matrix, 76
- multivector\_t
  - glucat::clifford\_algebra, 115
- glucat::framed\_multi, 143
- glucat::matrix\_multi, 195
- n
  - glucat::framed\_multi::hash\_size\_t, 163
- name
  - glucat::glucat\_error, 162
- NaN
  - glucat::numeric\_traits, 217
- nbar3
  - PyClical, 87
- nbr\_terms
  - glucat::framed\_multi, 151
- ninf3
  - PyClical, 87
- nnz
  - glucat::matrix, 76
- None
  - PyClical, 87
- nork
  - glucat::matrix, 76
- nork\_range
  - glucat::matrix, 77
- norm
  - glucat, 41
  - glucat::clifford\_algebra, 119
  - PyClical::clifford, 108
- norm\_frob2
  - glucat::matrix, 77
- normal
  - glucat::random\_generator, 250
- normal\_dist
  - glucat::random\_generator, 251
- numer
  - pade::pade\_log\_number, 230
  - pade::pade\_log\_number< dd\_real >, 231
  - pade::pade\_log\_number< float >, 232
  - pade::pade\_log\_number< long double >, 233
  - pade::pade\_log\_number< qd\_real >, 234
  - pade::pade\_sqrt\_number, 242
  - pade::pade\_sqrt\_number< dd\_real >, 243
  - pade::pade\_sqrt\_number< float >, 244
  - pade::pade\_sqrt\_number< long double >, 245
  - pade::pade\_sqrt\_number< qd\_real >, 246
- obj
  - PyClical, 87
- odd
  - glucat, 41
  - glucat::clifford\_algebra, 119
  - PyClical::clifford, 109
- offset\_level
  - glucat, 42
- offset\_to\_super
  - glucat::gen, 72
- operator &
  - glucat, 42–44
  - glucat::framed\_multi, 153
  - glucat::index\_set, 178

- glucat::matrix\_multi, 204
- operator &=
  - glucat::clifford\_algebra, 119
  - glucat::index\_set, 173
- operator bool
  - glucat::index\_set::reference, 255
- operator !=
  - glucat, 45
  - glucat::index\_set, 173
- operator <
  - glucat::index\_set, 173
- operator <<
  - glucat, 52, 53
  - glucat::framed\_multi, 153, 154
  - glucat::matrix\_multi, 205
- operator >>
  - glucat, 53, 54
  - glucat::framed\_multi, 154
  - glucat::matrix\_multi, 205
- operator \*
  - glucat, 47, 48
  - glucat::framed\_multi, 153
  - glucat::matrix\_multi, 205
- operator \*=
  - glucat::clifford\_algebra, 120
  - glucat::framed\_multi::var\_term, 264
- operator ~
  - glucat::index\_set, 175
  - glucat::index\_set::reference, 256
- operator ^
  - glucat, 54, 55
  - glucat::framed\_multi, 154
  - glucat::index\_set, 178
  - glucat::matrix\_multi, 206
- operator ^=
  - glucat::clifford\_algebra, 122
  - glucat::index\_set, 174
- operator()
  - glucat::clifford\_algebra, 119
  - glucat::framed\_multi::hash\_size\_t, 163
  - glucat::gen::generator\_table, 158
  - glucat::index\_set\_hash, 190
- operator +
  - glucat, 49
- operator +=
  - glucat::clifford\_algebra, 120
  - glucat::framed\_multi, 151
  - glucat::matrix\_multi, 202
- operator -
  - glucat, 50
  - glucat::clifford\_algebra, 120
- operator =
  - glucat::clifford\_algebra, 121
- operator /
  - glucat, 51, 52
  - glucat::framed\_multi, 153
  - glucat::matrix\_multi, 205
- operator /=
  - glucat::clifford\_algebra, 121
- operator=
  - glucat::basis\_table, 93
  - glucat::control\_t, 130
  - glucat::gen::generator\_table, 159
  - glucat::index\_set::reference, 255
  - glucat::matrix\_multi, 203
  - glucat::random\_generator, 251
- operator ==
  - glucat::clifford\_algebra, 121, 122
  - glucat::index\_set, 173
  - glucat::index\_set::reference, 255
- operator %
  - glucat, 46
  - glucat::framed\_multi, 153
  - glucat::matrix\_multi, 205
- operator %=
  - glucat::clifford\_algebra, 119
- operator []
  - glucat::clifford\_algebra, 122
  - glucat::index\_set, 174
- operator |
  - glucat, 55, 56
  - glucat::framed\_multi, 154
  - glucat::index\_set, 178
  - glucat::matrix\_multi, 206
- operator |=
  - glucat::clifford\_algebra, 122
  - glucat::index\_set, 174
- orientation\_t
  - glucat::matrix\_multi, 195
- outer\_pow
  - glucat, 57
  - glucat::clifford\_algebra, 122
  - PyClical::clifford, 109
- PY\_SSIZE\_T\_CLEAN
  - PyClical\_nocython.cpp, 324
- pade, 81
- pade::pade\_log\_denom
  - array, 223
  - denom, 224
- pade::pade\_log\_denom < dd\_real >, 224
  - array, 225
  - denom, 225
- pade::pade\_log\_denom < float >, 225
  - array, 226
  - denom, 226
- pade::pade\_log\_denom < long double >, 227
  - array, 227
  - denom, 227
- pade::pade\_log\_denom < qd\_real >, 228
  - array, 228
  - denom, 228
- pade::pade\_log\_denom < Scalar\_T >, 223
- pade::pade\_log\_numer
  - array, 230
  - numer, 230
- pade::pade\_log\_numer < dd\_real >, 230

- array, 231
- numer, 231
- pade::pade\_log\_numer< float >, 232
  - array, 232
  - numer, 232
- pade::pade\_log\_numer< long double >, 233
  - array, 233
  - numer, 233
- pade::pade\_log\_numer< qd\_real >, 234
  - array, 234
  - numer, 234
- pade::pade\_log\_numer< Scalar\_T >, 229
- pade::pade\_sqrt\_denom
  - array, 236
  - denom, 236
- pade::pade\_sqrt\_denom< dd\_real >, 236
  - array, 237
  - denom, 237
- pade::pade\_sqrt\_denom< float >, 238
  - array, 238
  - denom, 238
- pade::pade\_sqrt\_denom< long double >, 239
  - array, 239
  - denom, 239
- pade::pade\_sqrt\_denom< qd\_real >, 240
  - array, 240
  - denom, 240
- pade::pade\_sqrt\_denom< Scalar\_T >, 235
- pade::pade\_sqrt\_numer
  - array, 242
  - numer, 242
- pade::pade\_sqrt\_numer< dd\_real >, 242
  - array, 243
  - numer, 243
- pade::pade\_sqrt\_numer< float >, 244
  - array, 244
  - numer, 244
- pade::pade\_sqrt\_numer< long double >, 245
  - array, 245
  - numer, 245
- pade::pade\_sqrt\_numer< qd\_real >, 246
  - array, 246
  - numer, 246
- pade::pade\_sqrt\_numer< Scalar\_T >, 241
- pade\_approx
  - glucat, 57
- pade\_log
  - glucat, 57
- pair\_t
  - glucat::clifford\_algebra, 116
- pi
  - glucat::numeric\_traits, 218
  - PyClical, 88
- portability.h
  - \_GLUCAT\_ISINF, 310
  - \_GLUCAT\_ISNAN, 310
  - UBLAS\_ABS, 310
  - UBLAS\_SQRT, 310
- pos\_mod
  - glucat, 58
- pow
  - glucat, 58
  - glucat::clifford\_algebra, 123
  - glucat::numeric\_traits, 218
  - PyClical::clifford, 109
- print\_error\_msg
  - glucat::error, 137
  - glucat::glucat\_error, 161
- prod
  - glucat::matrix, 77
- pure
  - glucat, 59
  - glucat::clifford\_algebra, 123
  - PyClical::clifford, 110
- PyClical, 81
  - \_\_version\_\_, 85
  - \_test, 82
  - cl, 86
  - clifford\_hidden\_doctests, 82
  - e, 83
  - fill, 86
  - i, 86
  - index\_set\_hidden\_doctests, 84
  - ist, 86
  - istpq, 85
  - ixt, 86
  - lhs, 87
  - nbar3, 87
  - ninf3, 87
  - None, 87
  - obj, 87
  - pi, 88
  - rhs, 88
  - scalar\_epsilon, 88
  - tau, 88
  - threshold, 88
  - tol, 89
- PyClical.clifford, 94
- PyClical.h
  - Clifford, 319
  - clifford\_to\_repr, 320
  - clifford\_to\_str, 320
  - epsilon, 321
  - glucat\_package\_version, 322
  - hi\_ndx, 322
  - index\_set\_to\_repr, 320
  - index\_set\_to\_str, 321
  - IndexSet, 319
  - lo\_ndx, 322
  - PyFloat\_FromDouble, 321
  - scalar\_t, 320
  - String, 320
- PyClical.index\_set, 179
- PyClical::clifford
  - \_\_add\_\_, 96
  - \_\_and\_\_, 96

- `__call__`, 96
- `__cinit__`, 97
- `__contains__`, 97
- `__dealloc__`, 98
- `__getitem__`, 98
- `__iadd__`, 98
- `__iand__`, 99
- `__idiv__`, 99
- `__imod__`, 99
- `__imul__`, 100
- `__ior__`, 100
- `__isub__`, 100
- `__iter__`, 101
- `__ixor__`, 101
- `__mod__`, 101
- `__mul__`, 102
- `__neg__`, 102
- `__or__`, 102
- `__pos__`, 103
- `__pow__`, 103
- `__repr__`, 103
- `__richcmp__`, 104
- `__str__`, 104
- `__sub__`, 104
- `__truediv__`, 105
- `__xor__`, 105
- `abs`, 105
- `conj`, 106
- `even`, 106
- `frame`, 106
- `instance`, 112
- `inv`, 107
- `involute`, 107
- `isinf`, 107
- `isnan`, 108
- `max_abs`, 108
- `norm`, 108
- `odd`, 109
- `outer_pow`, 109
- `pow`, 109
- `pure`, 110
- `quad`, 110
- `reframe`, 110
- `reverse`, 111
- `scalar`, 111
- `truncated`, 111
- `vector_part`, 112
- `PyClical::index_set`
  - `__and__`, 180
  - `__cinit__`, 181
  - `__contains__`, 181
  - `__dealloc__`, 182
  - `__getitem__`, 182
  - `__iand__`, 182
  - `__invert__`, 183
  - `__ior__`, 183
  - `__iter__`, 183
  - `__ixor__`, 184
  - `__or__`, 184
  - `__repr__`, 184
  - `__richcmp__`, 185
  - `__setitem__`, 185
  - `__str__`, 185
  - `__xor__`, 186
  - `count`, 186
  - `count_neg`, 186
  - `count_pos`, 187
  - `hash_fn`, 187
  - `instance`, 189
  - `max`, 187
  - `min`, 188
  - `sign_of_mult`, 188
  - `sign_of_square`, 188
- `PyClical_nocython.cpp`
  - `PY_SSIZE_T_CLEAN`, 324
- `PyFloat_FromDouble`
  - `PyClical.h`, 321
- `pyclical/PyClical.h`, 318
- `pyclical/PyClical.pxd`, 322
- `pyclical/PyClical.pyx`, 322
- `pyclical/PyClical_nocython.cpp`, 323
- `pyclical/glucat.pxd`, 318
- `quad`
  - `glucat`, 59
  - `glucat::clifford_algebra`, 123
  - `PyClical::clifford`, 110
- `random`
  - `glucat::framed_multi`, 151
  - `glucat::matrix_multi`, 203
- `random_generator`
  - `glucat::random_generator`, 250
- `real`
  - `glucat`, 59
  - `glucat::numeric_traits`, 218
- `reference`
  - `glucat::index_set`, 178
  - `glucat::index_set::reference`, 254
- `reframe`
  - `glucat`, 59
  - `glucat::matrix_multi`, 206
  - `PyClical::clifford`, 110
- `reset`
  - `glucat::index_set`, 175
- `reverse`
  - `glucat`, 60
  - `glucat::clifford_algebra`, 123
  - `PyClical::clifford`, 111
- `rhs`
  - `PyClical`, 88
- `scalar`
  - `glucat`, 60
  - `glucat::clifford_algebra`, 123
  - `PyClical::clifford`, 111
- `scalar_epsilon`

- PyClical, 88
- Scalar\_T
  - glucat::matrix::eig\_genus, 134
- scalar\_t
  - glucat::clifford\_algebra, 116
  - glucat::framed\_multi, 143
  - glucat::matrix\_multi, 195
  - PyClical.h, 320
- seed
  - glucat::random\_generator, 252
- set
  - glucat::index\_set, 175, 176
- set\_value\_t
  - glucat, 23
- sign\_of\_mult
  - glucat::index\_set, 176
  - PyClical::index\_set, 188
- sign\_of\_square
  - glucat, 60
  - glucat::index\_set, 176
  - PyClical::index\_set, 188
- signature\_t
  - glucat::gen, 71
- signed\_perm\_nork
  - glucat::matrix, 78
- sin
  - glucat, 61
  - glucat::numeric\_traits, 219
- sinh
  - glucat, 61
  - glucat::numeric\_traits, 219
- size\_type
  - glucat::framed\_multi, 143
- sorted\_begin
  - glucat::sorted\_range, 258
  - glucat::sorted\_range< Sorted\_Map\_T, Sorted\_↔  
Map\_T >, 260
- sorted\_end
  - glucat::sorted\_range, 259
  - glucat::sorted\_range< Sorted\_Map\_T, Sorted\_↔  
Map\_T >, 261
- sorted\_iterator
  - glucat::sorted\_range, 258
  - glucat::sorted\_range< Sorted\_Map\_T, Sorted\_↔  
Map\_T >, 260
- sorted\_map\_t
  - glucat::framed\_multi, 143
  - glucat::sorted\_range, 258
  - glucat::sorted\_range< Sorted\_Map\_T, Sorted\_↔  
Map\_T >, 260
- sorted\_range
  - glucat::sorted\_range, 258
  - glucat::sorted\_range< Sorted\_Map\_T, Sorted\_↔  
Map\_T >, 260
- sparse\_prod
  - glucat::matrix, 78
- sqrt
  - glucat, 62, 63
- glucat::numeric\_traits, 219
- star
  - glucat, 63, 64
  - glucat::framed\_multi, 154
  - glucat::matrix\_multi, 206
- std, 89
  - std::numeric\_limits< glucat::framed\_multi< Scalar\_T,  
LO, HI, Tune\_P > >, 208
  - std::numeric\_limits< glucat::matrix\_multi< Scalar\_T,  
LO, HI, Tune\_P > >, 209
- String
  - PyClical.h, 320
- tan
  - glucat, 64
  - glucat::numeric\_traits, 219
- tanh
  - glucat, 65
  - glucat::numeric\_traits, 220
- tau
  - PyClical, 88
- term\_t
  - glucat::framed\_multi, 143
  - glucat::matrix\_multi, 196
- test
  - glucat::index\_set, 177
- test/control.h, 324
- test/driver.h, 325
- test/timing.h, 325
- test/try\_catch.h, 326
- test/tuning.h, 317
- threshold
  - PyClical, 88
- to\_demote
  - glucat, 65
- to\_double
  - glucat::numeric\_traits, 220
- to\_int
  - glucat::numeric\_traits, 220
- to\_lapack
  - glucat::matrix, 78
- to\_promote
  - glucat, 65
- to\_scalar\_t
  - glucat::numeric\_traits, 220–222
- tol
  - PyClical, 89
- trace
  - glucat::matrix, 79
- truncated
  - glucat::clifford\_algebra, 124
  - PyClical::clifford, 111
- try\_catch
  - glucat, 66
- tune\_p
  - glucat::framed\_multi, 144
  - glucat::matrix\_multi, 196
- Tuning\_Fast\_Basis\_Max\_Count
  - glucat, 67



- Tuning\_Fast\_CR\_Sqrt\_Max\_Steps
  - glucat, [68](#)
- Tuning\_Fast\_DB\_Sqrt\_Max\_Steps
  - glucat, [68](#)
- Tuning\_Fast\_Div\_Max\_Steps
  - glucat, [68](#)
- Tuning\_Fast\_Fast\_Size\_Threshold
  - glucat, [68](#)
- Tuning\_Fast\_Inv\_Fast\_Dim\_Threshold
  - glucat, [68](#)
- Tuning\_Fast\_Log\_Max\_Inner\_Steps
  - glucat, [68](#)
- Tuning\_Fast\_Log\_Max\_Outer\_Steps
  - glucat, [69](#)
- Tuning\_Fast\_Mult\_Matrix\_Threshold
  - glucat, [69](#)
- Tuning\_Fast\_Products\_Size\_Threshold
  - glucat, [69](#)
- Tuning\_Int\_Digits
  - glucat, [69](#)
- Tuning\_Max\_Threshold
  - glucat, [69](#)
- Tuning\_Naive\_Basis\_Max\_Count
  - glucat, [69](#)
- Tuning\_Naive\_Fast\_Size\_Threshold
  - glucat, [70](#)
- Tuning\_Naive\_Inv\_Fast\_Dim\_Threshold
  - glucat, [70](#)
- Tuning\_Naive\_Mult\_Matrix\_Threshold
  - glucat, [70](#)
- Tuning\_Slow\_Basis\_Max\_Count
  - glucat, [70](#)
- Tuning\_Slow\_Fast\_Size\_Threshold
  - glucat, [70](#)
- Tuning\_Slow\_Inv\_Fast\_Dim\_Threshold
  - glucat, [70](#)
- Tuning\_Slow\_Mult\_Matrix\_Threshold
  - glucat, [71](#)
- Tuning\_Slow\_Products\_Size\_Threshold
  - glucat, [71](#)
- tuning\_fast
  - glucat, [23](#)
- tuning\_naive
  - glucat, [24](#)
- tuning\_slow
  - glucat, [24](#)
- type
  - glucat::numeric\_traits::demoted, [133](#)
  - glucat::numeric\_traits::promoted, [248](#)
- UBLAS\_ABS
  - portability.h, [310](#)
- UBLAS\_SQRT
  - portability.h, [310](#)
- uint\_gen
  - glucat::random\_generator, [252](#)
- unfold
  - glucat::framed\_multi, [152](#)
  - glucat::index\_set, [177](#)
- uniform
  - glucat::random\_generator, [251](#)
- uniform\_dist
  - glucat::random\_generator, [252](#)
- unit
  - glucat::matrix, [79](#)
- v\_hi
  - glucat::clifford\_algebra, [125](#)
  - glucat::index\_set, [179](#)
- v\_lo
  - glucat::clifford\_algebra, [125](#)
  - glucat::index\_set, [179](#)
- valid
  - glucat::control\_t, [130](#)
- value\_of\_fold
  - glucat::index\_set, [177](#)
- var\_pair\_t
  - glucat::framed\_multi::var\_term, [263](#)
- var\_term
  - glucat::framed\_multi::var\_term, [263](#)
- var\_term\_t
  - glucat::framed\_multi, [144](#)
- vector\_part
  - glucat, [66](#)
  - glucat::clifford\_algebra, [124](#)
  - PyClical::clifford, [112](#)
- vector\_t
  - glucat::clifford\_algebra, [116](#)
  - glucat::framed\_multi, [144](#)
  - glucat::matrix\_multi, [196](#)
- verbose
  - glucat::control\_t, [130](#)
- write
  - glucat::clifford\_algebra, [124](#)