

MQCPack

Generated by Doxygen 1.8.16



<b>1 Modules Index</b>	<b>1</b>
1.1 Modules List	1
<b>2 Data Type Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Data Type Index</b>	<b>5</b>
3.1 Data Types List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Module Documentation</b>	<b>9</b>
5.1 mqc_algebra Module Reference	9
5.1.1 Function/Subroutine Documentation	15
5.1.1.1 bin_coeff()	15
5.1.1.2 factorial()	15
5.1.1.3 matrix_symm2sq_complex()	15
5.1.1.4 matrix_symm2sq_integer()	16
5.1.1.5 matrix_symm2sq_real()	16
5.1.1.6 mqc_allocate_matrix()	16
5.1.1.7 mqc_allocate_r4tensor()	16
5.1.1.8 mqc_allocate_scalar()	17
5.1.1.9 mqc_allocate_vector()	17
5.1.1.10 mqc_complexscalaradd()	18
5.1.1.11 mqc_complexscalardivide()	18
5.1.1.12 mqc_complexscalarmultiply()	18
5.1.1.13 mqc_complexscalarsubtract()	18
5.1.1.14 mqc_complexvectorproduct()	18
5.1.1.15 mqc_crossproduct()	18
5.1.1.16 mqc_deallocate_matrix()	19
5.1.1.17 mqc_deallocate_r4tensor()	19
5.1.1.18 mqc_deallocate_scalar()	19
5.1.1.19 mqc_deallocate_vector()	19
5.1.1.20 mqc_elementmatrixdivide()	19
5.1.1.21 mqc_elementmatrixproduct()	19
5.1.1.22 mqc_elementvectorproduct()	20
5.1.1.23 mqc_givens_matrix()	20
5.1.1.24 mqc_input_complex_scalar()	20
5.1.1.25 mqc_input_integer_scalar()	20
5.1.1.26 mqc_input_real_scalar()	21

---

5.1.1.27 <code>mqc_integertscalar()</code> . . . . .	21
5.1.1.28 <code>mqc_integerlescalar()</code> . . . . .	21
5.1.1.29 <code>mqc_integerscalaradd()</code> . . . . .	21
5.1.1.30 <code>mqc_integerscalardivide()</code> . . . . .	22
5.1.1.31 <code>mqc_integerscalarmultiply()</code> . . . . .	22
5.1.1.32 <code>mqc_integerscalarsubtract()</code> . . . . .	22
5.1.1.33 <code>mqc_integervectorproduct()</code> . . . . .	22
5.1.1.34 <code>mqc_length_vector()</code> . . . . .	22
5.1.1.35 <code>mqc_matrix_cast_complex()</code> . . . . .	22
5.1.1.36 <code>mqc_matrix_cast_real()</code> . . . . .	23
5.1.1.37 <code>mqc_matrix_columns()</code> . . . . .	23
5.1.1.38 <code>mqc_matrix_conjugate_transpose()</code> . . . . .	23
5.1.1.39 <code>mqc_matrix_copy_complex2int()</code> . . . . .	23
5.1.1.40 <code>mqc_matrix_copy_complex2real()</code> . . . . .	23
5.1.1.41 <code>mqc_matrix_copy_int2complex()</code> . . . . .	23
5.1.1.42 <code>mqc_matrix_copy_int2real()</code> . . . . .	24
5.1.1.43 <code>mqc_matrix_copy_real2complex()</code> . . . . .	24
5.1.1.44 <code>mqc_matrix_copy_real2int()</code> . . . . .	24
5.1.1.45 <code>mqc_matrix_determinant()</code> . . . . .	24
5.1.1.46 <code>mqc_matrix_diag2full()</code> . . . . .	24
5.1.1.47 <code>mqc_matrix_diag2symm()</code> . . . . .	24
5.1.1.48 <code>mqc_matrix_diagmatrix_put_complex()</code> . . . . .	25
5.1.1.49 <code>mqc_matrix_diagmatrix_put_integer()</code> . . . . .	25
5.1.1.50 <code>mqc_matrix_diagmatrix_put_real()</code> . . . . .	25
5.1.1.51 <code>mqc_matrix_diagmatrix_put_vector()</code> . . . . .	25
5.1.1.52 <code>mqc_matrix_diagonalize()</code> . . . . .	25
5.1.1.53 <code>mqc_matrix_full2diag()</code> . . . . .	25
5.1.1.54 <code>mqc_matrix_full2symm()</code> . . . . .	26
5.1.1.55 <code>mqc_matrix_generalized_eigensystem()</code> . . . . .	26
5.1.1.56 <code>mqc_matrix_havecomplex()</code> . . . . .	26
5.1.1.57 <code>mqc_matrix_havediagonal()</code> . . . . .	26
5.1.1.58 <code>mqc_matrix_havefull()</code> . . . . .	26
5.1.1.59 <code>mqc_matrix_haveinteger()</code> . . . . .	26
5.1.1.60 <code>mqc_matrix_havereal()</code> . . . . .	27
5.1.1.61 <code>mqc_matrix_havesymmetric()</code> . . . . .	27
5.1.1.62 <code>mqc_matrix_identity()</code> . . . . .	27
5.1.1.63 <code>mqc_matrix_initialize()</code> . . . . .	27
5.1.1.64 <code>mqc_matrix_inverse()</code> . . . . .	27
5.1.1.65 <code>mqc_matrix_isallocated()</code> . . . . .	27

---

---

5.1.1.66 mqc_matrix_matrix_at()	28
5.1.1.67 mqc_matrix_matrix_contraction()	29
5.1.1.68 mqc_matrix_matrix_put()	29
5.1.1.69 mqc_matrix_norm()	30
5.1.1.70 mqc_matrix_rms_max()	30
5.1.1.71 mqc_matrix_rows()	30
5.1.1.72 mqc_matrix_scalar_at()	30
5.1.1.73 mqc_matrix_scalar_put()	30
5.1.1.74 mqc_matrix_set()	31
5.1.1.75 mqc_matrix_sqrt()	31
5.1.1.76 mqc_matrix_storagetype()	31
5.1.1.77 mqc_matrix_svd()	31
5.1.1.78 mqc_matrix_symm2diag()	31
5.1.1.79 mqc_matrix_symm2full()	32
5.1.1.80 mqc_matrix_symm2full_func()	32
5.1.1.81 mqc_matrix_symmetrize()	32
5.1.1.82 mqc_matrix_symmmatrix_put_complex()	32
5.1.1.83 mqc_matrix_symmmatrix_put_integer()	32
5.1.1.84 mqc_matrix_symmmatrix_put_real()	32
5.1.1.85 mqc_matrix_symmsymmr4tensor_put_complex()	33
5.1.1.86 mqc_matrix_symmsymmr4tensor_put_real()	33
5.1.1.87 mqc_matrix_test_diagonal()	33
5.1.1.88 mqc_matrix_test_symmetric()	33
5.1.1.89 mqc_matrix_trace()	33
5.1.1.90 mqc_matrix_transpose()	33
5.1.1.91 mqc_matrix_vector_at()	34
5.1.1.92 mqc_matrix_vector_put()	34
5.1.1.93 mqc_matrixmatrixdotproduct()	34
5.1.1.94 mqc_matrixmatrixproduct()	34
5.1.1.95 mqc_matrixmatrixsubtract()	34
5.1.1.96 mqc_matrixmatrixsum()	35
5.1.1.97 mqc_matrixscalarproduct()	35
5.1.1.98 mqc_matrixvectordotproduct()	35
5.1.1.99 mqc_outer()	35
5.1.1.100 mqc_output_complex_scalar()	35
5.1.1.101 mqc_output_integer_scalar()	35
5.1.1.102 mqc_output_mqcscalar_scalar()	36
5.1.1.103 mqc_output_real_scalar()	36
5.1.1.104 mqc_print_matrix_algebra1()	36

---

---

5.1.1.105 mqc_print_r4tensor_algebra1()	36
5.1.1.106 mqc_print_scalar_algebra1()	36
5.1.1.107 mqc_print_vector_algebra1()	37
5.1.1.108 mqc_r4tensor_at()	37
5.1.1.109 mqc_r4tensor_havecomplex()	37
5.1.1.110 mqc_r4tensor_haveinteger()	37
5.1.1.111 mqc_r4tensor_havereal()	37
5.1.1.112 mqc_r4tensor_initialize()	38
5.1.1.113 mqc_r4tensor_put()	38
5.1.1.114 mqc_realgtscalar()	38
5.1.1.115 mqc_reallescalar()	38
5.1.1.116 mqc_realltscalar()	38
5.1.1.117 mqc_realscalaradd()	39
5.1.1.118 mqc_realscalardivide()	39
5.1.1.119 mqc_realscalarmultiply()	39
5.1.1.120 mqc_realscalarsubtract()	39
5.1.1.121 mqc_realvectorproduct()	39
5.1.1.122 mqc_scalar_acos()	39
5.1.1.123 mqc_scalar_asin()	40
5.1.1.124 mqc_scalar_atan()	40
5.1.1.125 mqc_scalar_atan2()	40
5.1.1.126 mqc_scalar_cmplx()	40
5.1.1.127 mqc_scalar_complex_conjugate()	40
5.1.1.128 mqc_scalar_complex_imagpart()	40
5.1.1.129 mqc_scalar_complex_realpart()	41
5.1.1.130 mqc_scalar_cos()	41
5.1.1.131 mqc_scalar_get_abs_value()	41
5.1.1.132 mqc_scalar_get_intrinsic_complex()	41
5.1.1.133 mqc_scalar_get_intrinsic_integer()	41
5.1.1.134 mqc_scalar_get_intrinsic_real()	41
5.1.1.135 mqc_scalar_get_random_value()	42
5.1.1.136 mqc_scalar_havecomplex()	42
5.1.1.137 mqc_scalar_haveinteger()	42
5.1.1.138 mqc_scalar_havereal()	42
5.1.1.139 mqc_scalar_isallocated()	42
5.1.1.140 mqc_scalar_sin()	42
5.1.1.141 mqc_scalar_sqrt()	43
5.1.1.142 mqc_scalar_tan()	43
5.1.1.143 mqc_scalaradd()	43

---

---

5.1.1.144 mqc_scalarcomplexadd()	43
5.1.1.145 mqc_scalarcomplexdivide()	43
5.1.1.146 mqc_scalarcomplexexponent()	43
5.1.1.147 mqc_scalarcomplexmultiply()	44
5.1.1.148 mqc_scalarcomplexsubtract()	44
5.1.1.149 mqc_scalardivide()	44
5.1.1.150 mqc_scalareq()	44
5.1.1.151 mqc_scalarexponent()	44
5.1.1.152 mqc_scalarge()	44
5.1.1.153 mqc_scalargt()	45
5.1.1.154 mqc_scalargtinteger()	45
5.1.1.155 mqc_scalargtreal()	45
5.1.1.156 mqc_scalarintegeradd()	45
5.1.1.157 mqc_scalarintegerdivide()	45
5.1.1.158 mqc_scalarintegrexponent()	45
5.1.1.159 mqc_scalarintegermultiply()	46
5.1.1.160 mqc_scalarintegersubtract()	46
5.1.1.161 mqc_scalarle()	46
5.1.1.162 mqc_scalarleinteger()	46
5.1.1.163 mqc_scalarlereal()	46
5.1.1.164 mqc_scalarlt()	46
5.1.1.165 mqc_scalarltreal()	47
5.1.1.166 mqc_scalarmatrixproduct()	47
5.1.1.167 mqc_scalarmultiply()	47
5.1.1.168 mqc_scalarne()	47
5.1.1.169 mqc_scalarrealadd()	47
5.1.1.170 mqc_scalarrealddivide()	47
5.1.1.171 mqc_scalarrealexponent()	48
5.1.1.172 mqc_scalarrealmultiply()	48
5.1.1.173 mqc_scalarrealsubtract()	48
5.1.1.174 mqc_scalarsubtract()	48
5.1.1.175 mqc_scalarvectordifference()	48
5.1.1.176 mqc_scalarvectorproduct()	48
5.1.1.177 mqc_scalarvectorsum()	49
5.1.1.178 mqc_set_array2tensor()	49
5.1.1.179 mqc_set_array2vector_complex()	49
5.1.1.180 mqc_set_array2vector_integer()	49
5.1.1.181 mqc_set_array2vector_real()	49
5.1.1.182 mqc_set_complexarray2matrix()	49

---

5.1.1.183 mqc_set_integerarray2matrix()	50
5.1.1.184 mqc_set_matrix2complexarray()	50
5.1.1.185 mqc_set_matrix2integerarray()	50
5.1.1.186 mqc_set_matrix2matrix()	50
5.1.1.187 mqc_set_matrix2realarray()	50
5.1.1.188 mqc_set_realarray2matrix()	50
5.1.1.189 mqc_set_vector2complexarray()	51
5.1.1.190 mqc_set_vector2integerarray()	51
5.1.1.191 mqc_set_vector2realarray()	51
5.1.1.192 mqc_set_vector2vector()	51
5.1.1.193 mqc_vector2diagmatrix()	51
5.1.1.194 mqc_vector_abs()	51
5.1.1.195 mqc_vector_argsort()	52
5.1.1.196 mqc_vector_cast_complex()	52
5.1.1.197 mqc_vector_cast_real()	52
5.1.1.198 mqc_vector_cmplx()	52
5.1.1.199 mqc_vector_complex_imagpart()	52
5.1.1.200 mqc_vector_complex_realpart()	52
5.1.1.201 mqc_vector_conjugate_transpose()	53
5.1.1.202 mqc_vector_copy_complex2int()	53
5.1.1.203 mqc_vector_copy_complex2real()	53
5.1.1.204 mqc_vector_copy_int2complex()	53
5.1.1.205 mqc_vector_copy_int2real()	53
5.1.1.206 mqc_vector_copy_real2complex()	53
5.1.1.207 mqc_vector_copy_real2int()	54
5.1.1.208 mqc_vector_havecomplex()	54
5.1.1.209 mqc_vector_haveinteger()	54
5.1.1.210 mqc_vector_havereal()	54
5.1.1.211 mqc_vector_initialize()	54
5.1.1.212 mqc_vector_isallocated()	54
5.1.1.213 mqc_vector_iscolumn()	55
5.1.1.214 mqc_vector_maxloc()	55
5.1.1.215 mqc_vector_maxval()	55
5.1.1.216 mqc_vector_minloc()	55
5.1.1.217 mqc_vector_minval()	55
5.1.1.218 mqc_vector_norm()	55
5.1.1.219 mqc_vector_pop()	56
5.1.1.220 mqc_vector_power()	56
5.1.1.221 mqc_vector_push()	56

---



5.1.1.222 mqc_vector_scalar_at()	56
5.1.1.223 mqc_vector_scalar_increment()	56
5.1.1.224 mqc_vector_scalar_put()	56
5.1.1.225 mqc_vector_shift()	57
5.1.1.226 mqc_vector_sort()	57
5.1.1.227 mqc_vector_sqrt()	57
5.1.1.228 mqc_vector_transpose()	57
5.1.1.229 mqc_vector_unshift()	57
5.1.1.230 mqc_vector_vector_at()	57
5.1.1.231 mqc_vector_vector_put()	58
5.1.1.232 mqc_vectorcomplexdivide()	58
5.1.1.233 mqc_vectorcomplexproduct()	58
5.1.1.234 mqc_vectorintegerdivide()	58
5.1.1.235 mqc_vectorintegerproduct()	58
5.1.1.236 mqc_vectormatrixdotproduct()	58
5.1.1.237 mqc_vectorrealdivide()	59
5.1.1.238 mqc_vectorrealproduct()	59
5.1.1.239 mqc_vectorscalardivide()	59
5.1.1.240 mqc_vectorscalarproduct()	59
5.1.1.241 mqc_vectorvectordifference()	59
5.1.1.242 mqc_vectorvectordotproduct()	59
5.1.1.243 mqc_vectorvectorsum()	60
5.1.1.244 symindexhash()	60
5.2 mqc_est Module Reference	60
5.2.1 Function/Subroutine Documentation	62
5.2.1.1 gen_det_str()	62
5.2.1.2 get_one_gamma_matrix()	62
5.2.1.3 mqc_build_ci_hamiltonian()	63
5.2.1.4 mqc_eigenvalue_eigenvalue_dotproduct()	63
5.2.1.5 mqc_eigenvalues_add_name()	63
5.2.1.6 mqc_eigenvalues_allocate()	63
5.2.1.7 mqc_eigenvalues_array_name()	63
5.2.1.8 mqc_eigenvalues_array_type()	64
5.2.1.9 mqc_eigenvalues_at()	64
5.2.1.10 mqc_eigenvalues_dimension()	64
5.2.1.11 mqc_eigenvalues_eigenvalues_multiply()	64
5.2.1.12 mqc_eigenvalues_has_alpha()	64
5.2.1.13 mqc_eigenvalues_has_beta()	64
5.2.1.14 mqc_eigenvalues_integral_multiply()	65

5.2.1.15 mqc_eigenvalues_isallocated()	65
5.2.1.16 mqc_eigenvalues_output_array()	65
5.2.1.17 mqc_eigenvalues_output_block()	65
5.2.1.18 mqc_eri_integral_contraction()	65
5.2.1.19 mqc_integral_add_name()	65
5.2.1.20 mqc_integral_allocate()	66
5.2.1.21 mqc_integral_array_name()	66
5.2.1.22 mqc_integral_array_type()	66
5.2.1.23 mqc_integral_at()	66
5.2.1.24 mqc_integral_conjugate_transpose()	66
5.2.1.25 mqc_integral_delete_energy_list()	67
5.2.1.26 mqc_integral_difference()	67
5.2.1.27 mqc_integral_dimension()	67
5.2.1.28 mqc_integral_eigenvalues_multiply()	67
5.2.1.29 mqc_integral_get_energy_list()	67
5.2.1.30 mqc_integral_has_alpha()	67
5.2.1.31 mqc_integral_has_alphabeta()	68
5.2.1.32 mqc_integral_has_beta()	68
5.2.1.33 mqc_integral_has_betaalpha()	68
5.2.1.34 mqc_integral_identity()	68
5.2.1.35 mqc_integral_initialize()	68
5.2.1.36 mqc_integral_integral_multiply()	69
5.2.1.37 mqc_integral_isallocated()	69
5.2.1.38 mqc_integral_matrix_multiply()	69
5.2.1.39 mqc_integral_norm()	69
5.2.1.40 mqc_integral_output_array()	69
5.2.1.41 mqc_integral_output_block()	69
5.2.1.42 mqc_integral_output_orbitals()	70
5.2.1.43 mqc_integral_scalar_multiply()	70
5.2.1.44 mqc_integral_set_energy_list()	70
5.2.1.45 mqc_integral_sum()	70
5.2.1.46 mqc_integral_swap_orbitals()	70
5.2.1.47 mqc_integral_transpose()	71
5.2.1.48 mqc_matrix_integral_multiply()	71
5.2.1.49 mqc_matrix_spinblockghf()	71
5.2.1.50 mqc_matrix_undospinblockghf_eigenvalues()	71
5.2.1.51 mqc_matrix_undospinblockghf_integral()	71
5.2.1.52 mqc_print_eigenvalues()	72
5.2.1.53 mqc_print_integral()	72

5.2.1.54 mqc_print_twoeris()	72
5.2.1.55 mqc_print_wavefunction()	72
5.2.1.56 mqc_scalar_integral_multiply()	72
5.2.1.57 mqc_scf_eigenvalues_power()	73
5.2.1.58 mqc_scf_integral_contraction()	73
5.2.1.59 mqc_scf_integral_determinant()	73
5.2.1.60 mqc_scf_integral_diagonalize()	73
5.2.1.61 mqc_scf_integral_generalized_eigensystem()	73
5.2.1.62 mqc_scf_integral_inverse()	74
5.2.1.63 mqc_scf_integral_trace()	74
5.2.1.64 mqc_scf_transformation_matrix()	74
5.2.1.65 mqc_twoeris_allocate()	74
5.2.1.66 mqc_twoeris_at()	74
5.2.1.67 slater_condon()	75
5.2.1.68 twoeri_trans()	75
<b>6 Data Type Documentation</b>	<b>77</b>
6.1 mqc_algebra::abs Interface Reference	77
6.1.1 Member Function/Subroutine Documentation	77
6.1.1.1 mqc_scalar_get_abs_value()	77
6.1.1.2 mqc_vector_abs()	77
6.2 mqc_algebra::acos Interface Reference	78
6.2.1 Member Function/Subroutine Documentation	78
6.2.1.1 mqc_scalar_acos()	78
6.3 mqc_algebra::aimag Interface Reference	78
6.3.1 Member Function/Subroutine Documentation	78
6.3.1.1 mqc_scalar_complex_imagpart()	78
6.3.1.2 mqc_vector_complex_imagpart()	79
6.4 mqc_algebra::asin Interface Reference	79
6.4.1 Member Function/Subroutine Documentation	79
6.4.1.1 mqc_scalar_asin()	79
6.5 mqc_algebra::assignment(=) Interface Reference	79
6.5.1 Member Function/Subroutine Documentation	80
6.5.1.1 mqc_input_complex_scalar()	80
6.5.1.2 mqc_input_integer_scalar()	80
6.5.1.3 mqc_input_real_scalar()	81
6.5.1.4 mqc_output_complex_scalar()	81
6.5.1.5 mqc_output_integer_scalar()	81
6.5.1.6 mqc_output_mqcscalar_scalar()	81

6.5.1.7 mqc_output_real_scalar()	82
6.5.1.8 mqc_set_array2tensor()	82
6.5.1.9 mqc_set_array2vector_complex()	82
6.5.1.10 mqc_set_array2vector_integer()	82
6.5.1.11 mqc_set_array2vector_real()	82
6.5.1.12 mqc_set_complexarray2matrix()	82
6.5.1.13 mqc_set_integerarray2matrix()	83
6.5.1.14 mqc_set_matrix2complexarray()	83
6.5.1.15 mqc_set_matrix2integerarray()	83
6.5.1.16 mqc_set_matrix2matrix()	83
6.5.1.17 mqc_set_matrix2realarray()	83
6.5.1.18 mqc_set_realarray2matrix()	83
6.5.1.19 mqc_set_vector2complexarray()	84
6.5.1.20 mqc_set_vector2integerarray()	84
6.5.1.21 mqc_set_vector2realarray()	84
6.5.1.22 mqc_set_vector2vector()	84
6.6 mqc_est::assignment(=) Interface Reference	84
6.6.1 Member Function/Subroutine Documentation	84
6.6.1.1 mqc_eigenvalues_output_array()	85
6.6.1.2 mqc_integral_output_array()	85
6.7 mqc_algebra::atan Interface Reference	85
6.7.1 Member Function/Subroutine Documentation	85
6.7.1.1 mqc_scalar_atan()	85
6.8 mqc_algebra::atan2 Interface Reference	85
6.8.1 Member Function/Subroutine Documentation	86
6.8.1.1 mqc_scalar_atan2()	86
6.9 mqc_algebra::cmplx Interface Reference	86
6.9.1 Member Function/Subroutine Documentation	86
6.9.1.1 mqc_scalar_cmplx()	86
6.9.1.2 mqc_vector_cmplx()	86
6.10 mqc_algebra::conjg Interface Reference	87
6.10.1 Member Function/Subroutine Documentation	87
6.10.1.1 mqc_scalar_complex_conjugate()	87
6.11 mqc_algebra::contraction Interface Reference	87
6.11.1 Member Function/Subroutine Documentation	87
6.11.1.1 mqc_matrix_matrix_contraction()	87
6.12 mqc_est::contraction Interface Reference	88
6.12.1 Member Function/Subroutine Documentation	88
6.12.1.1 mqc_eri_integral_contraction()	88

6.12.1.2 mqc_scf_integral_contraction()	88
6.13 mqc_algebra::cos Interface Reference	88
6.13.1 Member Function/Subroutine Documentation	88
6.13.1.1 mqc_scalar_cos()	89
6.14 mqc_algebra::dagger Interface Reference	89
6.14.1 Member Function/Subroutine Documentation	89
6.14.1.1 mqc_matrix_conjugate_transpose()	89
6.14.1.2 mqc_vector_conjugate_transpose()	89
6.15 mqc_est::dagger Interface Reference	89
6.15.1 Member Function/Subroutine Documentation	90
6.15.1.1 mqc_integral_conjugate_transpose()	90
6.16 mqc_algebra::dot_product Interface Reference	90
6.16.1 Member Function/Subroutine Documentation	90
6.16.1.1 mqc_vectorvectordotproduct()	90
6.17 mqc_est::dot_product Interface Reference	90
6.17.1 Member Function/Subroutine Documentation	91
6.17.1.1 mqc_eigenvalue_eigenvalue_dotproduct()	91
6.18 mqc_algebra::matmul Interface Reference	91
6.18.1 Member Function/Subroutine Documentation	91
6.18.1.1 mqc_matrixmatrixdotproduct()	91
6.18.1.2 mqc_matrixvectordotproduct()	91
6.18.1.3 mqc_vectormatrixdotproduct()	92
6.19 mqc_est::matmul Interface Reference	92
6.19.1 Member Function/Subroutine Documentation	92
6.19.1.1 mqc_eigenvalues_eigenvalues_multiply()	92
6.19.1.2 mqc_eigenvalues_integral_multiply()	92
6.19.1.3 mqc_integral_eigenvalues_multiply()	93
6.19.1.4 mqc_integral_integral_multiply()	93
6.19.1.5 mqc_integral_matrix_multiply()	93
6.19.1.6 mqc_matrix_integral_multiply()	93
6.20 mqc_algebra::matrix_symm2sq Interface Reference	93
6.20.1 Member Function/Subroutine Documentation	94
6.20.1.1 matrix_symm2sq_complex()	94
6.20.1.2 matrix_symm2sq_integer()	94
6.20.1.3 matrix_symm2sq_real()	94
6.21 mqc_algebra::mqc_cast_complex Interface Reference	94
6.21.1 Member Function/Subroutine Documentation	94
6.21.1.1 mqc_matrix_cast_complex()	95
6.21.1.2 mqc_vector_cast_complex()	95

6.22 mqc_algebra::mqc_cast_real Interface Reference	95
6.22.1 Member Function/Subroutine Documentation	95
6.22.1.1 mqc_matrix_cast_real()	95
6.22.1.2 mqc_vector_cast_real()	95
6.23 mqc_est::mqc_determinant Type Reference	96
6.23.1 Member Data Documentation	96
6.23.1.1 nalpstr	96
6.23.1.2 nbetstr	96
6.23.1.3 ndets	96
6.23.1.4 order	96
6.23.1.5 strings	96
6.24 mqc_est::mqc_determinant_string Type Reference	97
6.24.1 Member Data Documentation	97
6.24.1.1 alpha	97
6.24.1.2 beta	97
6.25 mqc_algebra::mqc_have_complex Interface Reference	97
6.25.1 Member Function/Subroutine Documentation	97
6.25.1.1 mqc_matrix_havecomplex()	97
6.25.1.2 mqc_vector_havecomplex()	98
6.26 mqc_algebra::mqc_have_int Interface Reference	98
6.26.1 Member Function/Subroutine Documentation	98
6.26.1.1 mqc_matrix_haveinteger()	98
6.26.1.2 mqc_vector_haveinteger()	98
6.27 mqc_algebra::mqc_have_real Interface Reference	98
6.27.1 Member Function/Subroutine Documentation	99
6.27.1.1 mqc_matrix_havereal()	99
6.27.1.2 mqc_vector_havereal()	99
6.28 mqc_algebra::mqc_matrix Type Reference	99
6.28.1 Member Function/Subroutine Documentation	100
6.28.1.1 at()	100
6.28.1.2 dagger()	100
6.28.1.3 det()	100
6.28.1.4 diag()	100
6.28.1.5 eigensys()	100
6.28.1.6 identity()	101
6.28.1.7 init()	101
6.28.1.8 initialize()	101
6.28.1.9 inv()	101
6.28.1.10 mat()	101

6.28.1.11 mput()	101
6.28.1.12 norm()	101
6.28.1.13 print()	102
6.28.1.14 put()	102
6.28.1.15 rmsmax()	102
6.28.1.16 s_type()	102
6.28.1.17 set()	102
6.28.1.18 sqrt()	102
6.28.1.19 svd()	102
6.28.1.20 trace()	103
6.28.1.21 transpose()	103
6.28.1.22 vat()	103
6.28.1.23 vput()	103
6.28.2 Member Data Documentation	103
6.28.2.1 matc	103
6.28.2.2 mati	103
6.28.2.3 matr	104
6.29 mqc_algebra::mqc_matrix_diagmatrix_put Interface Reference	104
6.29.1 Member Function/Subroutine Documentation	104
6.29.1.1 mqc_matrix_diagmatrix_put_complex()	104
6.29.1.2 mqc_matrix_diagmatrix_put_integer()	104
6.29.1.3 mqc_matrix_diagmatrix_put_real()	104
6.29.1.4 mqc_matrix_diagmatrix_put_vector()	105
6.30 mqc_algebra::mqc_matrix_symmmatrix_put Interface Reference	105
6.30.1 Member Function/Subroutine Documentation	105
6.30.1.1 mqc_matrix_symmmatrix_put_complex()	105
6.30.1.2 mqc_matrix_symmmatrix_put_integer()	105
6.30.1.3 mqc_matrix_symmmatrix_put_real()	105
6.31 mqc_est::mqc_matrix_undospinblockghf Interface Reference	106
6.31.1 Member Function/Subroutine Documentation	106
6.31.1.1 mqc_matrix_undospinblockghf_eigenvalues()	106
6.31.1.2 mqc_matrix_undospinblockghf_integral()	106
6.32 mqc_algebra::mqc_print Interface Reference	106
6.32.1 Member Function/Subroutine Documentation	106
6.32.1.1 mqc_print_matrix_algebra1()	107
6.32.1.2 mqc_print_r4tensor_algebra1()	107
6.32.1.3 mqc_print_scalar_algebra1()	107
6.32.1.4 mqc_print_vector_algebra1()	107
6.33 mqc_est::mqc_print Interface Reference	108

6.33.1 Member Function/Subroutine Documentation	108
6.33.1.1 mqc_print_eigenvalues()	108
6.33.1.2 mqc_print_integral()	108
6.33.1.3 mqc_print_twoeris()	108
6.33.1.4 mqc_print_wavefunction()	109
6.34 mqc_est::mqc_pscf_wavefunction Type Reference	109
6.34.1 Member Data Documentation	109
6.34.1.1 nactive	109
6.34.1.2 ncore	110
6.34.1.3 nfrz	110
6.34.1.4 nval	110
6.34.1.5 pscf_amplitudes	110
6.34.1.6 pscf_energies	110
6.35 mqc_algebra::mqc_r4tensor Type Reference	110
6.35.1 Member Function/Subroutine Documentation	111
6.35.1.1 at()	111
6.35.1.2 init()	111
6.35.1.3 initialize()	111
6.35.1.4 print()	111
6.35.1.5 put()	111
6.36 mqc_algebra::mqc_scalar Type Reference	111
6.36.1 Member Function/Subroutine Documentation	112
6.36.1.1 abs()	112
6.36.1.2 cval()	112
6.36.1.3 ival()	112
6.36.1.4 print()	112
6.36.1.5 random()	112
6.36.1.6 rval()	112
6.37 mqc_est::mqc_scf_eigenvalues Type Reference	113
6.37.1 Member Function/Subroutine Documentation	113
6.37.1.1 addlabel()	113
6.37.1.2 at()	113
6.37.1.3 getblock()	113
6.37.1.4 getlabel()	113
6.37.1.5 power()	113
6.37.1.6 print()	114
6.38 mqc_est::mqc_scf_integral Type Reference	114
6.38.1 Member Function/Subroutine Documentation	114
6.38.1.1 addlabel()	114



6.38.1.2 deleteelist()	114
6.38.1.3 det()	115
6.38.1.4 diag()	115
6.38.1.5 eigensys()	115
6.38.1.6 getblock()	115
6.38.1.7 getelist()	115
6.38.1.8 getlabel()	115
6.38.1.9 identity()	115
6.38.1.10 init()	116
6.38.1.11 inv()	116
6.38.1.12 norm()	116
6.38.1.13 orbitals()	116
6.38.1.14 print()	116
6.38.1.15 setelist()	116
6.38.1.16 swap()	116
6.38.1.17 trace()	117
6.39 mqc_algebra::mqc_set_array2vector Interface Reference	117
6.39.1 Member Function/Subroutine Documentation	117
6.39.1.1 mqc_set_array2vector_complex()	117
6.39.1.2 mqc_set_array2vector_integer()	117
6.39.1.3 mqc_set_array2vector_real()	117
6.40 mqc_est::mqc_twoeris Type Reference	118
6.40.1 Member Function/Subroutine Documentation	118
6.40.1.1 print()	118
6.41 mqc_algebra::mqc_vector Type Reference	118
6.41.1 Member Function/Subroutine Documentation	119
6.41.1.1 abs()	119
6.41.1.2 argsort()	119
6.41.1.3 at()	119
6.41.1.4 dagger()	119
6.41.1.5 diag()	119
6.41.1.6 init()	120
6.41.1.7 initialize()	120
6.41.1.8 maxloc()	120
6.41.1.9 maxval()	120
6.41.1.10 minloc()	120
6.41.1.11 minval()	120
6.41.1.12 norm()	120
6.41.1.13 pop()	121

6.41.1.14 power()	121
6.41.1.15 print()	121
6.41.1.16 push()	121
6.41.1.17 put()	121
6.41.1.18 shift()	121
6.41.1.19 size()	121
6.41.1.20 sort()	122
6.41.1.21 sqrt()	122
6.41.1.22 transpose()	122
6.41.1.23 unshift()	122
6.41.1.24 vat()	122
6.41.1.25 vput()	122
6.41.2 Member Data Documentation	122
6.41.2.1 data_type	122
6.41.2.2 length	123
6.41.2.3 vecc	123
6.41.2.4 veci	123
6.41.2.5 vecr	123
6.42 mqc_est::mqc_wavefunction Type Reference	123
6.42.1 Member Function/Subroutine Documentation	124
6.42.1.1 print()	124
6.42.2 Member Data Documentation	124
6.42.2.1 basis	124
6.42.2.2 charge	124
6.42.2.3 core_hamiltonian	125
6.42.2.4 density_matrix	125
6.42.2.5 fock_matrix	125
6.42.2.6 mo_coefficients	125
6.42.2.7 mo_energies	125
6.42.2.8 mo_symmetries	125
6.42.2.9 multiplicity	125
6.42.2.10 nalpha	126
6.42.2.11 nbasis	126
6.42.2.12 nbeta	126
6.42.2.13 nelectrons	126
6.42.2.14 overlap_matrix	126
6.42.2.15 scf_density_matrix	126
6.42.2.16 symmetry	126
6.42.2.17 wf_complex	127

6.42.2.18 wf_type . . . . .	127
6.43 mqc_algebra::operator(*) Interface Reference . . . . .	127
6.43.1 Member Function/Subroutine Documentation . . . . .	127
6.43.1.1 mqc_complexscalarmultiply() . . . . .	128
6.43.1.2 mqc_complexvectorproduct() . . . . .	128
6.43.1.3 mqc_integerscalarmultiply() . . . . .	128
6.43.1.4 mqc_integervectorproduct() . . . . .	128
6.43.1.5 mqc_matrixmatrixproduct() . . . . .	128
6.43.1.6 mqc_matrixscalarproduct() . . . . .	128
6.43.1.7 mqc_realscalarmultiply() . . . . .	129
6.43.1.8 mqc_realvectorproduct() . . . . .	129
6.43.1.9 mqc_scalarcomplexmultiply() . . . . .	129
6.43.1.10 mqc_scalarintegermultiply() . . . . .	129
6.43.1.11 mqc_scalarmatrixproduct() . . . . .	129
6.43.1.12 mqc_scalarmultiply() . . . . .	129
6.43.1.13 mqc_scalarrealmultiply() . . . . .	130
6.43.1.14 mqc_scalarvectorproduct() . . . . .	130
6.43.1.15 mqc_vectorcomplexproduct() . . . . .	130
6.43.1.16 mqc_vectorintegerproduct() . . . . .	130
6.43.1.17 mqc_vectorrealproduct() . . . . .	130
6.43.1.18 mqc_vectorscalarproduct() . . . . .	130
6.44 mqc_est::operator(*) Interface Reference . . . . .	131
6.44.1 Member Function/Subroutine Documentation . . . . .	131
6.44.1.1 mqc_integral_scalar_multiply() . . . . .	131
6.44.1.2 mqc_scalar_integral_multiply() . . . . .	131
6.45 mqc_algebra::operator(**) Interface Reference . . . . .	131
6.45.1 Member Function/Subroutine Documentation . . . . .	131
6.45.1.1 mqc_scalarcomplexexponent() . . . . .	132
6.45.1.2 mqc_scalarexponent() . . . . .	132
6.45.1.3 mqc_scalarintegerexponent() . . . . .	132
6.45.1.4 mqc_scalarrealexponent() . . . . .	132
6.46 mqc_est::operator(+) Interface Reference . . . . .	132
6.46.1 Member Function/Subroutine Documentation . . . . .	132
6.46.1.1 mqc_integral_sum() . . . . .	133
6.47 mqc_algebra::operator(+) Interface Reference . . . . .	133
6.47.1 Member Function/Subroutine Documentation . . . . .	133
6.47.1.1 mqc_complexscalaradd() . . . . .	133
6.47.1.2 mqc_integerscalaradd() . . . . .	133
6.47.1.3 mqc_matrixmatrixsum() . . . . .	134

6.47.1.4 mqc_realscalaradd()	134
6.47.1.5 mqc_scalaradd()	134
6.47.1.6 mqc_scalarcomplexadd()	134
6.47.1.7 mqc_scalarintegeradd()	134
6.47.1.8 mqc_scalarrealadd()	134
6.47.1.9 mqc_scalarvectorsum()	135
6.47.1.10 mqc_vectorvectorsum()	135
6.48 mqc_est::operator(-) Interface Reference	135
6.48.1 Member Function/Subroutine Documentation	135
6.48.1.1 mqc_integral_difference()	135
6.49 mqc_algebra::operator(-) Interface Reference	136
6.49.1 Member Function/Subroutine Documentation	136
6.49.1.1 mqc_complexscalarsubtract()	136
6.49.1.2 mqc_integerscalarsubtract()	136
6.49.1.3 mqc_matrixmatrixsubtract()	136
6.49.1.4 mqc_realscalarsubtract()	137
6.49.1.5 mqc_scalarcomplexsubtract()	137
6.49.1.6 mqc_scalarintegersubtract()	137
6.49.1.7 mqc_scalarrealsubtract()	137
6.49.1.8 mqc_scalarsubtract()	137
6.49.1.9 mqc_scalarvectordifference()	137
6.49.1.10 mqc_vectorvectordifference()	138
6.50 mqc_algebra::operator(.dot.) Interface Reference	138
6.50.1 Member Function/Subroutine Documentation	138
6.50.1.1 mqc_matrixmatrixdotproduct()	138
6.50.1.2 mqc_matrixvectordotproduct()	138
6.50.1.3 mqc_vectormatrixdotproduct()	138
6.50.1.4 mqc_vectorvectordotproduct()	139
6.51 mqc_algebra::operator(.eq.) Interface Reference	139
6.51.1 Member Function/Subroutine Documentation	139
6.51.1.1 mqc_scalareq()	139
6.52 mqc_algebra::operator(.ewd.) Interface Reference	139
6.52.1 Member Function/Subroutine Documentation	139
6.52.1.1 mqc_elementmatrixdivide()	140
6.53 mqc_algebra::operator(.ewp.) Interface Reference	140
6.53.1 Member Function/Subroutine Documentation	140
6.53.1.1 mqc_elementmatrixproduct()	140
6.53.1.2 mqc_elementvectorproduct()	140
6.54 mqc_algebra::operator(.ge.) Interface Reference	140

6.54.1 Member Function/Subroutine Documentation	141
6.54.1.1 mqc_scalarge()	141
6.55 mqc_algebra::operator(.gt.) Interface Reference	141
6.55.1 Member Function/Subroutine Documentation	141
6.55.1.1 mqc_integertscalar()	141
6.55.1.2 mqc_realgtscalar()	141
6.55.1.3 mqc_scalargt()	142
6.55.1.4 mqc_scalargtinteger()	142
6.55.1.5 mqc_scalargtreal()	142
6.56 mqc_algebra::operator(.le.) Interface Reference	142
6.56.1 Member Function/Subroutine Documentation	142
6.56.1.1 mqc_integerlescalar()	143
6.56.1.2 mqc_reallescalar()	143
6.56.1.3 mqc_scalarle()	143
6.56.1.4 mqc_scalarleinteger()	143
6.56.1.5 mqc_scalarlereal()	143
6.57 mqc_algebra::operator(.lt.) Interface Reference	144
6.57.1 Member Function/Subroutine Documentation	144
6.57.1.1 mqc_realltscalar()	144
6.57.1.2 mqc_scalarlt()	144
6.57.1.3 mqc_scalarltreal()	144
6.58 mqc_algebra::operator(.ne.) Interface Reference	144
6.58.1 Member Function/Subroutine Documentation	145
6.58.1.1 mqc_scalarne()	145
6.59 mqc_algebra::operator(.outer.) Interface Reference	145
6.59.1 Member Function/Subroutine Documentation	145
6.59.1.1 mqc_outer()	145
6.60 mqc_algebra::operator(.x.) Interface Reference	145
6.60.1 Member Function/Subroutine Documentation	146
6.60.1.1 mqc_crossproduct()	146
6.61 mqc_algebra::operator(/) Interface Reference	146
6.61.1 Member Function/Subroutine Documentation	146
6.61.1.1 mqc_complexscalardivide()	146
6.61.1.2 mqc_integerscalardivide()	147
6.61.1.3 mqc_realscalardivide()	147
6.61.1.4 mqc_scalarcomplexdivide()	147
6.61.1.5 mqc_scalardivide()	147
6.61.1.6 mqc_scalarintegerdivide()	147
6.61.1.7 mqc_scalarrealdivide()	147

6.61.1.8 mqc_vectorcomplexdivide()	148
6.61.1.9 mqc_vectorintegerdivide()	148
6.61.1.10 mqc_vectorrealddivide()	148
6.61.1.11 mqc_vectorscalardivide()	148
6.62 mqc_algebra::real Interface Reference	148
6.62.1 Member Function/Subroutine Documentation	148
6.62.1.1 mqc_scalar_complex_realpart()	149
6.62.1.2 mqc_vector_complex_realpart()	149
6.63 mqc_algebra::sin Interface Reference	149
6.63.1 Member Function/Subroutine Documentation	149
6.63.1.1 mqc_scalar_sin()	149
6.64 mqc_algebra::sqrt Interface Reference	149
6.64.1 Member Function/Subroutine Documentation	150
6.64.1.1 mqc_scalar_sqrt()	150
6.65 mqc_algebra::tan Interface Reference	150
6.65.1 Member Function/Subroutine Documentation	150
6.65.1.1 mqc_scalar_tan()	150
6.66 mqc_est::transpose Interface Reference	150
6.66.1 Member Function/Subroutine Documentation	150
6.66.1.1 mqc_integral_transpose()	151
6.67 mqc_algebra::transpose Interface Reference	151
6.67.1 Member Function/Subroutine Documentation	151
6.67.1.1 mqc_matrix_transpose()	151
6.67.1.2 mqc_vector_transpose()	151
<b>7 File Documentation</b>	<b>153</b>
7.1 src/mqc_algebra.F03 File Reference	153
7.2 src/mqc_est.F03 File Reference	159
<b>Index</b>	<b>163</b>

# Chapter 1

## Modules Index

### 1.1 Modules List

Here is a list of all modules with brief descriptions:

<a href="#">mqc_algebra</a>	9
<a href="#">mqc_est</a>	60





## Chapter 2

# Data Type Index

### 2.1 Class Hierarchy

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

<code>mqc_algebra::abs</code>	77
<code>mqc_algebra::acos</code>	78
<code>mqc_algebra::aimag</code>	78
<code>mqc_algebra::asin</code>	79
<code>mqc_algebra::assignment(=)</code>	79
<code>mqc_est::assignment(=)</code>	84
<code>mqc_algebra::atan</code>	85
<code>mqc_algebra::atan2</code>	85
<code>mqc_algebra::cmplx</code>	86
<code>mqc_algebra::conjg</code>	87
<code>mqc_algebra::contraction</code>	87
<code>mqc_est::contraction</code>	88
<code>mqc_algebra::cos</code>	88
<code>mqc_algebra::dagger</code>	89
<code>mqc_est::dagger</code>	89
<code>mqc_algebra::dot_product</code>	90
<code>mqc_est::dot_product</code>	90
<code>mqc_algebra::matmul</code>	91
<code>mqc_est::matmul</code>	92
<code>mqc_algebra::matrix_symm2sq</code>	93
<code>mqc_algebra::mqc_cast_complex</code>	94
<code>mqc_algebra::mqc_cast_real</code>	95
<code>mqc_est::mqc_determinant</code>	96
<code>mqc_est::mqc_determinant_string</code>	97
<code>mqc_algebra::mqc_have_complex</code>	97
<code>mqc_algebra::mqc_have_int</code>	98
<code>mqc_algebra::mqc_have_real</code>	98
<code>mqc_algebra::mqc_matrix</code>	99
<code>mqc_algebra::mqc_matrix_diagmatrix_put</code>	104
<code>mqc_algebra::mqc_matrix_symmmatrix_put</code>	105
<code>mqc_est::mqc_matrix_undospinblockghf</code>	106

mqc_algebra::mqc_print . . . . .	106
mqc_est::mqc_print . . . . .	108
mqc_algebra::mqc_r4tensor . . . . .	110
mqc_algebra::mqc_scalar . . . . .	111
mqc_est::mqc_scf_eigenvalues . . . . .	113
mqc_est::mqc_scf_integral . . . . .	114
mqc_algebra::mqc_set_array2vector . . . . .	117
mqc_est::mqc_twoeris . . . . .	118
mqc_algebra::mqc_vector . . . . .	118
mqc_est::mqc_wavefunction . . . . .	123
mqc_est::mqc_pscf_wavefunction . . . . .	109
mqc_algebra::operator(*) . . . . .	127
mqc_est::operator(*) . . . . .	131
mqc_algebra::operator(**) . . . . .	131
mqc_est::operator(+) . . . . .	132
mqc_algebra::operator(+) . . . . .	133
mqc_est::operator(-) . . . . .	135
mqc_algebra::operator(-) . . . . .	136
mqc_algebra::operator(.dot.) . . . . .	138
mqc_algebra::operator(.eq.) . . . . .	139
mqc_algebra::operator(.ewd.) . . . . .	139
mqc_algebra::operator(.ewp.) . . . . .	140
mqc_algebra::operator(.ge.) . . . . .	140
mqc_algebra::operator(.gt.) . . . . .	141
mqc_algebra::operator(.le.) . . . . .	142
mqc_algebra::operator(.lt.) . . . . .	144
mqc_algebra::operator(.ne.) . . . . .	144
mqc_algebra::operator(.outer.) . . . . .	145
mqc_algebra::operator(.x.) . . . . .	145
mqc_algebra::operator(/) . . . . .	146
mqc_algebra::real . . . . .	148
mqc_algebra::sin . . . . .	149
mqc_algebra::sqrt . . . . .	149
mqc_algebra::tan . . . . .	150
mqc_est::transpose . . . . .	150
mqc_algebra::transpose . . . . .	151

## Chapter 3

# Data Type Index

### 3.1 Data Types List

Here are the data types with brief descriptions:

<a href="#">mqc_algebra::abs</a>	77
<a href="#">mqc_algebra::acos</a>	78
<a href="#">mqc_algebra::aimag</a>	78
<a href="#">mqc_algebra::asin</a>	79
<a href="#">mqc_algebra::assignment(=)</a>	79
<a href="#">mqc_est::assignment(=)</a>	84
<a href="#">mqc_algebra::atan</a>	85
<a href="#">mqc_algebra::atan2</a>	85
<a href="#">mqc_algebra::cmplx</a>	86
<a href="#">mqc_algebra::conjg</a>	87
<a href="#">mqc_algebra::contraction</a>	87
<a href="#">mqc_est::contraction</a>	88
<a href="#">mqc_algebra::cos</a>	88
<a href="#">mqc_algebra::dagger</a>	89
<a href="#">mqc_est::dagger</a>	89
<a href="#">mqc_algebra::dot_product</a>	90
<a href="#">mqc_est::dot_product</a>	90
<a href="#">mqc_algebra::matmul</a>	91
<a href="#">mqc_est::matmul</a>	92
<a href="#">mqc_algebra::matrix_symm2sq</a>	93
<a href="#">mqc_algebra::mqc_cast_complex</a>	94
<a href="#">mqc_algebra::mqc_cast_real</a>	95
<a href="#">mqc_est::mqc_determinant</a>	96
<a href="#">mqc_est::mqc_determinant_string</a>	97
<a href="#">mqc_algebra::mqc_have_complex</a>	97
<a href="#">mqc_algebra::mqc_have_int</a>	98
<a href="#">mqc_algebra::mqc_have_real</a>	98
<a href="#">mqc_algebra::mqc_matrix</a>	99
<a href="#">mqc_algebra::mqc_matrix_diagmatrix_put</a>	104
<a href="#">mqc_algebra::mqc_matrix_symmmatrix_put</a>	105
<a href="#">mqc_est::mqc_matrix_undospinblockghf</a>	106

<a href="#">mqc_algebra::mqc_print</a>	106
<a href="#">mqc_est::mqc_print</a>	108
<a href="#">mqc_est::mqc_pscf_wavefunction</a>	109
<a href="#">mqc_algebra::mqc_r4tensor</a>	110
<a href="#">mqc_algebra::mqc_scalar</a>	111
<a href="#">mqc_est::mqc_scf_eigenvalues</a>	113
<a href="#">mqc_est::mqc_scf_integral</a>	114
<a href="#">mqc_algebra::mqc_set_array2vector</a>	117
<a href="#">mqc_est::mqc_twoeris</a>	118
<a href="#">mqc_algebra::mqc_vector</a>	118
<a href="#">mqc_est::mqc_wavefunction</a>	123
<a href="#">mqc_algebra::operator(*)</a>	127
<a href="#">mqc_est::operator(*)</a>	131
<a href="#">mqc_algebra::operator(**)</a>	131
<a href="#">mqc_est::operator(+)</a>	132
<a href="#">mqc_algebra::operator(+)</a>	133
<a href="#">mqc_est::operator(-)</a>	135
<a href="#">mqc_algebra::operator(-)</a>	136
<a href="#">mqc_algebra::operator(.dot.)</a>	138
<a href="#">mqc_algebra::operator(.eq.)</a>	139
<a href="#">mqc_algebra::operator(.ewd.)</a>	139
<a href="#">mqc_algebra::operator(.ewp.)</a>	140
<a href="#">mqc_algebra::operator(.ge.)</a>	140
<a href="#">mqc_algebra::operator(.gt.)</a>	141
<a href="#">mqc_algebra::operator(.le.)</a>	142
<a href="#">mqc_algebra::operator(.lt.)</a>	144
<a href="#">mqc_algebra::operator(.ne.)</a>	144
<a href="#">mqc_algebra::operator(.outer.)</a>	145
<a href="#">mqc_algebra::operator(.x.)</a>	145
<a href="#">mqc_algebra::operator(/)</a>	146
<a href="#">mqc_algebra::real</a>	148
<a href="#">mqc_algebra::sin</a>	149
<a href="#">mqc_algebra::sqrt</a>	149
<a href="#">mqc_algebra::tan</a>	150
<a href="#">mqc_est::transpose</a>	150
<a href="#">mqc_algebra::transpose</a>	151

## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">mqc_algebra.F03</a> . . . . .	153
src/ <a href="#">mqc_est.F03</a> . . . . .	159



## Chapter 5

# Module Documentation

### 5.1 mqc\_algebra Module Reference

#### Data Types

- interface [abs](#)
- interface [acos](#)
- interface [aimag](#)
- interface [asin](#)
- interface [assignment\(=\)](#)
- interface [atan](#)
- interface [atan2](#)
- interface [cmplx](#)
- interface [conjg](#)
- interface [contraction](#)
- interface [cos](#)
- interface [dagger](#)
- interface [dot\\_product](#)
- interface [matmul](#)
- interface [matrix\\_symm2sq](#)
- interface [mqc\\_cast\\_complex](#)
- interface [mqc\\_cast\\_real](#)
- interface [mqc\\_have\\_complex](#)
- interface [mqc\\_have\\_int](#)
- interface [mqc\\_have\\_real](#)
- type [mqc\\_matrix](#)
- interface [mqc\\_matrix\\_diagmatrix\\_put](#)
- interface [mqc\\_matrix\\_symmmatrix\\_put](#)
- interface [mqc\\_print](#)
- type [mqc\\_r4tensor](#)
- type [mqc\\_scalar](#)
- interface [mqc\\_set\\_array2vector](#)
- type [mqc\\_vector](#)
- interface [operator\(\\*\)](#)

- interface `operator(**)`
- interface `operator(+)`
- interface `operator(-)`
- interface `operator(.dot.)`
- interface `operator(.eq.)`
- interface `operator(.ewd.)`
- interface `operator(.ewp.)`
- interface `operator(.ge.)`
- interface `operator(.gt.)`
- interface `operator(.le.)`
- interface `operator(.lt.)`
- interface `operator(.ne.)`
- interface `operator(.outer.)`
- interface `operator(.x.)`
- interface `operator(/)`
- interface `real`
- interface `sin`
- interface `sqrt`
- interface `tan`
- interface `transpose`

## Functions/Subroutines

- integer(kind=int64) function `factorial` (n)
- integer(kind=int64) function `bin_coeff` (N, K)
- subroutine `mqc_allocate_scalar` (Scalar, Data\_type)

***MQC\_Allocate\_Scalar is used to allocate a scalar type variable of the MQC\_Scalar class***

- subroutine `mqc_deallocate_scalar` (Scalar)
- logical function `mqc_scalar_isallocated` (Scalar)
- subroutine `mqc_input_integer_scalar` (ScalarOut, ScalarIn)

***MQC\_Input\_Integer\_Scalar is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar***

- subroutine `mqc_input_real_scalar` (ScalarOut, ScalarIn)
- subroutine `mqc_input_complex_scalar` (ScalarOut, ScalarIn)
- subroutine `mqc_output_mqcscalar_scalar` (ScalarOut, ScalarIn)
- subroutine `mqc_output_integer_scalar` (ScalarOut, ScalarIn)
- subroutine `mqc_output_real_scalar` (ScalarOut, ScalarIn)
- subroutine `mqc_output_complex_scalar` (ScalarOut, ScalarIn)
- subroutine `mqc_print_scalar_algebra1` (Scalar, IOut, Header, Blank\_At\_Top, Blank\_At\_Bottom)
- type(`mqc_scalar`) function `mqc_scalar_cmplx` (Scalar1, Scalar2)
- type(`mqc_scalar`) function `mqc_scalar_sqrt` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_sin` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_cos` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_tan` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_asin` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_acos` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_atan` (Scalar)
- type(`mqc_scalar`) function `mqc_scalar_atan2` (Scalar)
- logical function `mqc_scalar_havereal` (Scalar)
- logical function `mqc_scalar_haveinteger` (Scalar)



- logical function [mqc\\_scalar\\_havecomplex](#) (Scalar)
- [real](#)(kind=real64) function [mqc\\_scalar\\_get\\_intrinsic\\_real](#) (Scalar)
- [integer](#)(kind=int64) function [mqc\\_scalar\\_get\\_intrinsic\\_integer](#) (Scalar)
- [complex](#)(kind=real64) function [mqc\\_scalar\\_get\\_intrinsic\\_complex](#) (Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_get\\_abs\\_value](#) (Scalar)
- subroutine [mqc\\_scalar\\_get\\_random\\_value](#) (Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalaradd](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_scalarsubtract](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_scalarmultiply](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_scalardivide](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_scalarexponent](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegerexponent](#) (Scalar, IntIn)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealexponent](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexexponent](#) (Scalar, ComplIn)
- logical function [mqc\\_scalarne](#) (Scalar1, Scalar2)
- logical function [mqc\\_scalareq](#) (Scalar1, Scalar2)
- logical function [mqc\\_scalarlt](#) (Scalar1, Scalar2)
- logical function [mqc\\_realltscalar](#) (RealIn, Scalar)
- logical function [mqc\\_scalarltreal](#) (Scalar, RealIn)
- logical function [mqc\\_scalargt](#) (Scalar1, Scalar2)
- logical function [mqc\\_integertgtscalar](#) (IntIn, Scalar)
- logical function [mqc\\_scalargtinteger](#) (Scalar, IntIn)
- logical function [mqc\\_realgtscalar](#) (RealIn, Scalar)
- logical function [mqc\\_scalargtreal](#) (Scalar, RealIn)
- logical function [mqc\\_scalarle](#) (Scalar1, Scalar2)
- logical function [mqc\\_reallescalar](#) (RealIn, Scalar)
- logical function [mqc\\_scalarlereal](#) (Scalar, RealIn)
- logical function [mqc\\_integerlescalar](#) (IntIn, Scalar)
- logical function [mqc\\_scalarleinteger](#) (Scalar, IntIn)
- logical function [mqc\\_scalarge](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_complex\\_conjugate](#) (ScalarIn)
- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_complex\\_realpart](#) (ScalarIn)
- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_complex\\_imagpart](#) (ScalarIn)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalarmultiply](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegermultiply](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalarmultiply](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealmultiply](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalarmultiply](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexmultiply](#) (Scalar, ComplexIn)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalardivide](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegerdivide](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalardivide](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealddivide](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalardivide](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexdivide](#) (Scalar, ComplexIn)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalaradd](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegeradd](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalaradd](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealadd](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalaradd](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexadd](#) (Scalar, ComplexIn)

- type([mqc\\_scalar](#)) function [mqc\\_integerscalarsubtract](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegersubtract](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalarsubtract](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealsubtract](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalarsubtract](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexsubtract](#) (Scalar, ComplexIn)
- subroutine [mqc\\_allocate\\_vector](#) (N, Vector, Data\_Type)
- subroutine [mqc\\_deallocate\\_vector](#) (Vector)
- integer(kind=int64) function [mqc\\_length\\_vector](#) (Vector)
- logical function [mqc\\_vector\\_havereal](#) (Vector)
- logical function [mqc\\_vector\\_haveinteger](#) (Vector)
- logical function [mqc\\_vector\\_havecomplex](#) (Vector)
- logical function [mqc\\_vector\\_iscolumn](#) (Vector)
- subroutine [mqc\\_vector\\_copy\\_int2real](#) (Vector)
- subroutine [mqc\\_vector\\_copy\\_int2complex](#) (Vector)
- subroutine [mqc\\_vector\\_copy\\_real2int](#) (Vector)
- subroutine [mqc\\_vector\\_copy\\_real2complex](#) (Vector)
- subroutine [mqc\\_vector\\_copy\\_complex2int](#) (Vector)
- subroutine [mqc\\_vector\\_copy\\_complex2real](#) (Vector)
- type([mqc\\_scalar](#)) function [mqc\\_vector\\_scalar\\_at](#) (Vec, I)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_vector\\_at](#) (Vec, I, J)
- subroutine [mqc\\_set\\_vector2integerarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_set\\_vector2realarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_set\\_vector2complexarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_set\\_array2vector\\_integer](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_array2vector\\_real](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_array2vector\\_complex](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_vector2vector](#) (VectorOut, VectorIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorvectorsum](#) (Vector1In, Vector2In)
- type([mqc\\_vector](#)) function [mqc\\_vectorvectordifference](#) (Vector1In, Vector2In)
- type([mqc\\_vector](#)) function [mqc\\_scalarvectorsum](#) (ScalarIn, VectorIn)
- type([mqc\\_vector](#)) function [mqc\\_scalarvectordifference](#) (ScalarIn, VectorIn)
- type([mqc\\_vector](#)) function [mqc\\_elementvectorproduct](#) (Vector1In, Vector2In)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_transpose](#) (Vector)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_conjugate\\_transpose](#) (Vector)
- type([mqc\\_scalar](#)) function [mqc\\_vectorvectordotproduct](#) (Vector1, Vector2)
- type([mqc\\_matrix](#)) function [mqc\\_outer](#) (VA, VB)
- type([mqc\\_vector](#)) function [mqc\\_crossproduct](#) (Vector1In, Vector2In)
- subroutine [mqc\\_print\\_vector\\_algebra1](#) (Vector, IOut, Header, Verbose, Blank\_At\_Top, Blank\_At\_Bottom)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_cast\\_real](#) (VA)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_cast\\_complex](#) (VA)
- subroutine [mqc\\_vector\\_scalar\\_put](#) (Vector, Scalar, I)
- subroutine [mqc\\_vector\\_scalar\\_increment](#) (Vector, Scalar, I)
- subroutine [mqc\\_vector\\_vector\\_put](#) (Vector, VectorIn, I)
- subroutine [mqc\\_vector\\_initialize](#) (Vector, Length, Scalar)
- type([mqc\\_vector](#)) function [mqc\\_scalarvectorproduct](#) (Scalar, Vector)
- type([mqc\\_vector](#)) function [mqc\\_vectorscalarproduct](#) (vector, scalar)
- type([mqc\\_vector](#)) function [mqc\\_vectorscalardivide](#) (vector, scalar)
- type([mqc\\_vector](#)) function [mqc\\_realvectorproduct](#) (RealIn, Vector)
- type([mqc\\_vector](#)) function [mqc\\_vectorrealproduct](#) (vector, realIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorrealddivide](#) (vector, realIn)

- type([mqc\\_vector](#)) function [mqc\\_integervectorproduct](#) (intIn, Vector)
  - type([mqc\\_vector](#)) function [mqc\\_vectorintegerproduct](#) (vector, intIn)
  - type([mqc\\_vector](#)) function [mqc\\_vectorintegerdivide](#) (vector, intIn)
  - type([mqc\\_vector](#)) function [mqc\\_complexvectorproduct](#) (Compln, Vector)
  - type([mqc\\_vector](#)) function [mqc\\_vectorcomplexproduct](#) (vector, compln)
  - type([mqc\\_vector](#)) function [mqc\\_vectorcomplexdivide](#) (vector, compln)
  - type([mqc\\_scalar](#)) function [mqc\\_vector\\_norm](#) (vector, methodIn)
  - logical function [mqc\\_vector\\_isallocated](#) (Vector)
  - subroutine [mqc\\_vector\\_push](#) (Vector, Scalar)
  - subroutine [mqc\\_vector\\_unshift](#) (Vector, Scalar)
  - type([mqc\\_scalar](#)) function [mqc\\_vector\\_pop](#) (Vector)
  - type([mqc\\_scalar](#)) function [mqc\\_vector\\_shift](#) (Vector)
  - type([mqc\\_scalar](#)) function [mqc\\_vector\\_maxval](#) (Vector)
  - type([mqc\\_scalar](#)) function [mqc\\_vector\\_minval](#) (Vector)
  - integer function [mqc\\_vector\\_maxloc](#) (Vector)
  - integer function [mqc\\_vector\\_minloc](#) (Vector)
  - type([mqc\\_vector](#)) function [mqc\\_vector\\_argsort](#) (Vector)
  - subroutine [mqc\\_vector\\_sort](#) (Vector, idx)
  - subroutine [mqc\\_vector\\_sqrt](#) (A)
  - type([mqc\\_vector](#)) function [mqc\\_vector\\_abs](#) (A)
  - subroutine [mqc\\_vector\\_power](#) (A, P)
  - type([mqc\\_vector](#)) function [mqc\\_vector\\_complex\\_realpart](#) (A)
  - type([mqc\\_vector](#)) function [mqc\\_vector\\_complex\\_imagpart](#) (A)
  - type([mqc\\_vector](#)) function [mqc\\_vector\\_cmplx](#) (Vector1, Vector2)
  - character(len=64) function [mqc\\_matrix\\_storage\\_type](#) (Matrix)
  - subroutine [mqc\\_matrix\\_diagonalize](#) (A, EVals, EVecs)
  - type([mqc\\_matrix](#)) function [mqc\\_matrix\\_cast\\_real](#) (MA)
  - type([mqc\\_matrix](#)) function [mqc\\_matrix\\_cast\\_complex](#) (MA)
  - type([mqc\\_scalar](#)) function [mqc\\_matrix\\_scalar\\_at](#) (Mat, I, J)
  - type([mqc\\_vector](#)) function [mqc\\_matrix\\_vector\\_at](#) (Mat, Rows, Cols)
  - recursive subroutine [mqc\\_matrix\\_vector\\_put](#) (Mat, VectorIn, Rows, Cols)
  - type([mqc\\_matrix](#)) function [mqc\\_matrix\\_matrix\\_at](#) (Mat, Rows, Cols)
- Mqc\_Matrix\_Matrix\_At is a function that returns a submatrix of the matrix***
- subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_vector](#) (diagVectorIn, mat)
  - subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_integer](#) (mat, diagMatrixIn)
  - subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_real](#) (mat, diagMatrixIn)
  - subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_complex](#) (mat, diagMatrixIn)
  - subroutine [mqc\\_matrix\\_symmmatrix\\_put\\_integer](#) (mat, symmMatrixIn)
  - subroutine [mqc\\_matrix\\_symmmatrix\\_put\\_real](#) (mat, symmMatrixIn)
  - subroutine [mqc\\_matrix\\_symmmatrix\\_put\\_complex](#) (mat, symmMatrixIn)
  - recursive subroutine [mqc\\_matrix\\_matrix\\_put](#) (Mat, MatrixIn, Rows, Cols)
  - integer(kind=int64) function [symindexhash](#) (i, j, k, l)
  - type([mqc\\_matrix](#)) function [mqc\\_elementmatrixproduct](#) (A, B)
  - type([mqc\\_matrix](#)) function [mqc\\_elementmatrixdivide](#) (A, B)
  - logical function [mqc\\_matrix\\_test\\_symmetric](#) (Matrix, Option)
  - logical function [mqc\\_matrix\\_test\\_diagonal](#) (Matrix)
  - subroutine [mqc\\_allocate\\_matrix](#) (M, N, Matrix, Data\_Type, Storage)
  - subroutine [mqc\\_deallocate\\_matrix](#) (Matrix)
  - logical function [mqc\\_matrix\\_isallocated](#) (Matrix)
  - subroutine [mqc\\_set\\_integerarray2matrix](#) (MatrixOut, ArrayIn)

- subroutine [mqc\\_set\\_realarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_set\\_complexarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_set\\_matrix2integerarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_set\\_matrix2realarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_set\\_matrix2complexarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_set\\_matrix2matrix](#) (MatrixOut, MatrixIn)
- subroutine [mqc\\_print\\_matrix\\_algebra1](#) (Matrix, IOut, Header, Blank\_At\_Top, Blank\_At\_Bottom)
- subroutine [mqc\\_matrix\\_copy\\_int2real](#) (Matrix)
- subroutine [mqc\\_matrix\\_copy\\_int2complex](#) (Matrix)
- subroutine [mqc\\_matrix\\_copy\\_real2int](#) (Matrix)
- subroutine [mqc\\_matrix\\_copy\\_real2complex](#) (Matrix)
- subroutine [mqc\\_matrix\\_copy\\_complex2int](#) (Matrix)
- subroutine [mqc\\_matrix\\_copy\\_complex2real](#) (Matrix)
- integer(kind=int64) function [mqc\\_matrix\\_rows](#) (Matrix)
- integer(kind=int64) function [mqc\\_matrix\\_columns](#) (Matrix)
- logical function [mqc\\_matrix\\_havereal](#) (Matrix)
- logical function [mqc\\_matrix\\_haveinteger](#) (Matrix)
- logical function [mqc\\_matrix\\_havecomplex](#) (Matrix)
- logical function [mqc\\_matrix\\_havfull](#) (Matrix)
- logical function [mqc\\_matrix\\_havesymmetric](#) (Matrix)
- logical function [mqc\\_matrix\\_havediagonal](#) (Matrix)
- type([mqc\\_matrix](#)) function [mqc\\_matrix\\_transpose](#) (Matrix)
- type([mqc\\_matrix](#)) function [mqc\\_matrix\\_conjugate\\_transpose](#) (Matrix)
- type([mqc\\_matrix](#)) function [mqc\\_matrix\\_symmetrize](#) (Matrix)
- subroutine [mqc\\_matrix\\_full2symm](#) (Matrix)
- subroutine [mqc\\_matrix\\_symm2full](#) (Matrix, Option)
- subroutine [mqc\\_matrix\\_full2diag](#) (Matrix)
- subroutine [mqc\\_matrix\\_diag2full](#) (Matrix)
- subroutine [mqc\\_matrix\\_symm2diag](#) (Matrix)
- subroutine [mqc\\_matrix\\_diag2symm](#) (Matrix)
- type([mqc\\_matrix](#)) function [mqc\\_matrix\\_symm2full\\_func](#) (Matrix)
- subroutine [matrix\\_symm2sq\\_integer](#) (N, I\_Symm, I\_Sq)
- subroutine [matrix\\_symm2sq\\_real](#) (N, A\_Symm, A\_Sq)
- subroutine [matrix\\_symm2sq\\_complex](#) (N, A\_Symm, A\_Sq)
- type([mqc\\_matrix](#)) function [mqc\\_vector2diagmatrix](#) (vector)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixsum](#) (MA, MB)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixsubtract](#) (MA, MB)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixproduct](#) (MA, MB)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixdotproduct](#) (MA, MB)
- type([mqc\\_vector](#)) function [mqc\\_matrixvectordotproduct](#) (MA, VB)
- type([mqc\\_vector](#)) function [mqc\\_vectormatrixdotproduct](#) (VA, MB)
- type([mqc\\_matrix](#)) function [mqc\\_matrixscalarproduct](#) (Matrix, Scalar)
- type([mqc\\_matrix](#)) function [mqc\\_scalarmatrixproduct](#) (Scalar, Matrix)
- type([mqc\\_scalar](#)) function [mqc\\_matrix\\_matrix\\_contraction](#) (Matrix1, Matrix2)
- subroutine [mqc\\_matrix\\_scalar\\_put](#) (Matrix, Scalar, I, J)
- subroutine [mqc\\_matrix\\_initialize](#) (Matrix, Rows, Columns, Scalar, Storage)
- subroutine [mqc\\_matrix\\_identity](#) (matrix, n, m)
- subroutine [mqc\\_matrix\\_set](#) (matrix, scalar, storage)
- type([mqc\\_scalar](#)) function [mqc\\_matrix\\_norm](#) (matrix, methodIn)
- type([mqc\\_scalar](#)) function [mqc\\_matrix\\_determinant](#) (a)
- type([mqc\\_matrix](#)) function [mqc\\_matrix\\_inverse](#) (a)

- type([mqc\\_scalar](#)) function [mqc\\_matrix\\_trace](#) (matrix)
- subroutine [mqc\\_matrix\\_generalized\\_eigensystem](#) (a, bIn, eigenvals, reigenvecs, leigenvecs)
- subroutine [mqc\\_matrix\\_svd](#) (A, EVals, EUVecs, EVVecs)
- subroutine [mqc\\_matrix\\_rms\\_max](#) (A, rms\_A, max\_A)
- subroutine [mqc\\_matrix\\_sqrt](#) (A, eVals, eVecs)
- subroutine [mqc\\_allocate\\_r4tensor](#) (I, J, K, L, Tensor, Data\_Type, Storage)
- subroutine [mqc\\_deallocate\\_r4tensor](#) (Tensor)
- type([mqc\\_scalar](#)) function [mqc\\_r4tensor\\_at](#) (Tensor, I, J, K, L)
- subroutine [mqc\\_r4tensor\\_put](#) (Tensor, Element, I, J, K, L)
- subroutine [mqc\\_print\\_r4tensor\\_algebra1](#) (Tensor, IOut, Header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_set\\_array2tensor](#) (TensorOut, ArrayIn)
- subroutine [mqc\\_r4tensor\\_initialize](#) (R4Tensor, I, J, K, L, Scalar)
- subroutine [mqc\\_matrix\\_symmsymmr4tensor\\_put\\_real](#) (r4Tensor, symmSymmMatrixIn)
- subroutine [mqc\\_matrix\\_symmsymmr4tensor\\_put\\_complex](#) (r4Tensor, symmSymmMatrixIn)
- logical function [mqc\\_r4tensor\\_haveinteger](#) (R4Tensor)
- logical function [mqc\\_r4tensor\\_havereal](#) (R4Tensor)
- logical function [mqc\\_r4tensor\\_havecomplex](#) (R4Tensor)
- type([mqc\\_matrix](#)) function [mqc\\_givens\\_matrix](#) (m\_size, angle, p, q)

### 5.1.1 Function/Subroutine Documentation

#### 5.1.1.1 bin\_coeff()

```
integer(kind=int64) function mqc_algebra::bin_coeff (
    integer(kind=int64), intent(in) N,
    integer(kind=int64), intent(in) K )
```

#### 5.1.1.2 factorial()

```
integer(kind=int64) function mqc_algebra::factorial (
    integer(kind=int64), intent(in) n )
```

#### 5.1.1.3 matrix\_symm2sq\_complex()

```
subroutine mqc_algebra::matrix_symm2sq_complex (
    integer(kind=int64), intent(in) N,
    complex(kind=real64), dimension(:), intent(in) A_Symm,
    complex(kind=real64), dimension(n,n), intent(out) A_Sq )
```

#### 5.1.1.4 matrix\_symm2sq\_integer()

```
subroutine mqc_algebra::matrix_symm2sq_integer (
    integer(kind=int64), intent(in) N,
    integer(kind=int64), dimension(:), intent(in) I_Symm,
    integer(kind=int64), dimension(n,n), intent(out) I_Sq )
```

#### 5.1.1.5 matrix\_symm2sq\_real()

```
subroutine mqc_algebra::matrix_symm2sq_real (
    integer(kind=int64), intent(in) N,
    real(kind=real64), dimension(:), intent(in) A_Symm,
    real(kind=real64), dimension(n,n), intent(out) A_Sq )
```

#### 5.1.1.6 mqc\_allocate\_matrix()

```
subroutine mqc_algebra::mqc_allocate_matrix (
    integer(kind=int64), intent(in) M,
    integer(kind=int64), intent(in) N,
    class(mqc_matrix), intent(inout) Matrix,
    character(len=*), intent(in) Data_Type,
    character(len=*), intent(in) Storage )
```

#### 5.1.1.7 mqc\_allocate\_r4tensor()

```
subroutine mqc_algebra::mqc_allocate_r4tensor (
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in) J,
    integer(kind=int64), intent(in) K,
    integer(kind=int64), intent(in) L,
    type(mqc_r4tensor), intent(inout) Tensor,
    character(len=*), intent(in) Data_Type,
    character(len=*), intent(in) Storage )
```

### 5.1.1.8 mqc\_allocate\_scalar()

```
subroutine mqc_algebra::mqc_allocate_scalar (
    type(mqc_scalar), intent(inout) Scalar,
    character(len=*), intent(in) Data_Type )
```

**MQC\_Allocate\_Scalar** is used to allocate a scalar type variable of the **MQC\_Scalar** class

#### Purpose:

MQC\_Allocate\_Scalar is a subroutine used to allocate a scalar type variable of the MQC\_Scalar class. The following options are available:

1. Data\_Type = 'Real' declares the MQC\_Scalar variable to be of real type.
2. Data\_Type = 'Integer' declares the MQC\_Scalar variable to be of integer type.
3. Data\_Type = 'Complex' declares the MQC\_Scalar variable to be of complex type.

#### Parameters

in, out	<i>Scalar</i>	Scalar is Type(MQC_Scalar) The name of the MQC_Scalar variable
in	<i>Data_Type</i>	Data_Type is Character(Len=*) = 'Real': the MQC_Scalar is real = 'Integer': the MQC_Scalar is integer = 'Complex': the MQC_Scalar is complex

#### Author

L. M. Thompson

#### Date

2016

### 5.1.1.9 mqc\_allocate\_vector()

```
subroutine mqc_algebra::mqc_allocate_vector (
    integer(kind=int64), intent(in) N,
    type(mqc_vector), intent(inout) Vector,
    character(len=*), intent(in) Data_Type )
```

#### 5.1.1.10 mqc\_complexscalaradd()

```
type(mqc_scalar) function mqc_algebra::mqc_complexscalaradd (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.11 mqc\_complexscalardivide()

```
type(mqc_scalar) function mqc_algebra::mqc_complexscalardivide (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.12 mqc\_complexscalarmultiply()

```
type(mqc_scalar) function mqc_algebra::mqc_complexscalarmultiply (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.13 mqc\_complexscalarsubtract()

```
type(mqc_scalar) function mqc_algebra::mqc_complexscalarsubtract (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.14 mqc\_complexvectorproduct()

```
type(mqc_vector) function mqc_algebra::mqc_complexvectorproduct (
    complex(kind=real64), intent(in) CompIn,
    type(mqc_vector), intent(in) Vector )
```

#### 5.1.1.15 mqc\_crossproduct()

```
type(mqc_vector) function mqc_algebra::mqc_crossproduct (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```



**5.1.1.16 mqc\_deallocate\_matrix()**

```
subroutine mqc_algebra::mqc_deallocate_matrix (
    class(mqc_matrix), intent(inout) Matrix )
```

**5.1.1.17 mqc\_deallocate\_r4tensor()**

```
subroutine mqc_algebra::mqc_deallocate_r4tensor (
    type(mqc_r4tensor), intent(inout) Tensor )
```

**5.1.1.18 mqc\_deallocate\_scalar()**

```
subroutine mqc_algebra::mqc_deallocate_scalar (
    type(mqc_scalar), intent(inout) Scalar )
```

**5.1.1.19 mqc\_deallocate\_vector()**

```
subroutine mqc_algebra::mqc_deallocate_vector (
    type(mqc_vector), intent(inout) Vector )
```

**5.1.1.20 mqc\_elementmatrixdivide()**

```
type(mqc_matrix) function mqc_algebra::mqc_elementmatrixdivide (
    type(mqc_matrix), intent(in) A,
    type(mqc_matrix), intent(in) B )
```

**5.1.1.21 mqc\_elementmatrixproduct()**

```
type(mqc_matrix) function mqc_algebra::mqc_elementmatrixproduct (
    type(mqc_matrix), intent(in) A,
    type(mqc_matrix), intent(in) B )
```

#### 5.1.1.22 mqc\_elementvectorproduct()

```
type(mqc_vector) function mqc_algebra::mqc_elementvectorproduct (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

#### 5.1.1.23 mqc\_givens\_matrix()

```
type(mqc_matrix) function mqc_algebra::mqc_givens_matrix (
    integer(kind=int64), intent(in) m_size,
    real(kind=real64), intent(in) angle,
    integer(kind=int64), intent(in) p,
    integer(kind=int64), intent(in) q )
```

#### 5.1.1.24 mqc\_input\_complex\_scalar()

```
subroutine mqc_algebra::mqc_input_complex_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    complex(kind=real64), intent(in) ScalarIn )
```

#### 5.1.1.25 mqc\_input\_integer\_scalar()

```
subroutine mqc_algebra::mqc_input_integer_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    integer(kind=int64), intent(in) ScalarIn )
```

**MQC\_Input\_Integer\_Scalar is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar**

##### Purpose:

MQC\_Input\_Integer\_Scalar is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar.

##### Parameters

in, out	<i>ScalarOut</i>	ScalarOut is Type(MQC_Scalar) The name of the output variable
in	<i>ScalarIn</i>	ScalarIn is Integer(kind=int64) The value of the input variable

**Author**

L. M. Thompson

**Date**

2016

**5.1.1.26 mqc\_input\_real\_scalar()**

```
subroutine mqc_algebra::mqc_input_real_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    real(kind=real64), intent(in) ScalarIn )
```

**5.1.1.27 mqc\_integergtscalar()**

```
logical function mqc_algebra::mqc_integergtscalar (
    integer(kind=int64), intent(in) IntIn,
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.28 mqc\_integerlescalar()**

```
logical function mqc_algebra::mqc_integerlescalar (
    integer(kind=int64), intent(in) IntIn,
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.29 mqc\_integerscalaradd()**

```
type(mqc_scalar) function mqc_algebra::mqc_integerscalaradd (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.30 mqc\_integerscalardivide()

```
type(mqc_scalar) function mqc_algebra::mqc_integerscalardivide (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.31 mqc\_integerscalarmultiply()

```
type(mqc_scalar) function mqc_algebra::mqc_integerscalarmultiply (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.32 mqc\_integerscalarsubtract()

```
type(mqc_scalar) function mqc_algebra::mqc_integerscalarsubtract (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.33 mqc\_integervectorproduct()

```
type(mqc_vector) function mqc_algebra::mqc_integervectorproduct (
    integer(kind=int64), intent(in) intIn,
    type(mqc_vector), intent(in) Vector )
```

#### 5.1.1.34 mqc\_length\_vector()

```
integer(kind=int64) function mqc_algebra::mqc_length_vector (
    class(mqc_vector) Vector )
```

#### 5.1.1.35 mqc\_matrix\_cast\_complex()

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_cast_complex (
    type(mqc_matrix), intent(in) MA )
```

#### 5.1.1.36 mqc\_matrix\_cast\_real()

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_cast_real (
    type(mqc_matrix), intent(in) MA )
```

#### 5.1.1.37 mqc\_matrix\_columns()

```
integer(kind=int64) function mqc_algebra::mqc_matrix_columns (
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.38 mqc\_matrix\_conjugate\_transpose()

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_conjugate_transpose (
    class(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.39 mqc\_matrix\_copy\_complex2int()

```
subroutine mqc_algebra::mqc_matrix_copy_complex2int (
    type(mqc_matrix) Matrix )
```

#### 5.1.1.40 mqc\_matrix\_copy\_complex2real()

```
subroutine mqc_algebra::mqc_matrix_copy_complex2real (
    type(mqc_matrix) Matrix )
```

#### 5.1.1.41 mqc\_matrix\_copy\_int2complex()

```
subroutine mqc_algebra::mqc_matrix_copy_int2complex (
    type(mqc_matrix) Matrix )
```

#### 5.1.1.42 mqc\_matrix\_copy\_int2real()

```
subroutine mqc_algebra::mqc_matrix_copy_int2real (  
    type(mqc_matrix) Matrix )
```

#### 5.1.1.43 mqc\_matrix\_copy\_real2complex()

```
subroutine mqc_algebra::mqc_matrix_copy_real2complex (  
    type(mqc_matrix) Matrix )
```

#### 5.1.1.44 mqc\_matrix\_copy\_real2int()

```
subroutine mqc_algebra::mqc_matrix_copy_real2int (  
    type(mqc_matrix) Matrix )
```

#### 5.1.1.45 mqc\_matrix\_determinant()

```
type(mqc_scalar) function mqc_algebra::mqc_matrix_determinant (  
    class(mqc_matrix) a )
```

#### 5.1.1.46 mqc\_matrix\_diag2full()

```
subroutine mqc_algebra::mqc_matrix_diag2full (  
    type(mqc_matrix), intent(inout) Matrix )
```

#### 5.1.1.47 mqc\_matrix\_diag2symm()

```
subroutine mqc_algebra::mqc_matrix_diag2symm (  
    type(mqc_matrix), intent(inout) Matrix )
```

#### 5.1.1.48 mqc\_matrix\_diagmatrix\_put\_complex()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put_complex (
    class(mqc_matrix), intent(inout) mat,
    complex(kind=real64), dimension(:), intent(in) diagMatrixIn )
```

#### 5.1.1.49 mqc\_matrix\_diagmatrix\_put\_integer()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put_integer (
    class(mqc_matrix), intent(inout) mat,
    integer(kind=int64), dimension(:), intent(in) diagMatrixIn )
```

#### 5.1.1.50 mqc\_matrix\_diagmatrix\_put\_real()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put_real (
    class(mqc_matrix), intent(inout) mat,
    real(kind=real64), dimension(:), intent(in) diagMatrixIn )
```

#### 5.1.1.51 mqc\_matrix\_diagmatrix\_put\_vector()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put_vector (
    class(mqc_vector), intent(in) diagVectorIn,
    class(mqc_matrix), intent(inout) mat )
```

#### 5.1.1.52 mqc\_matrix\_diagonalize()

```
subroutine mqc_algebra::mqc_matrix_diagonalize (
    class(mqc_matrix), intent(in) A,
    type(mqc_vector), intent(inout), optional EVals,
    type(mqc_matrix), intent(inout), optional EVecs )
```

#### 5.1.1.53 mqc\_matrix\_full2diag()

```
subroutine mqc_algebra::mqc_matrix_full2diag (
    type(mqc_matrix), intent(inout) Matrix )
```

#### 5.1.1.54 mqc\_matrix\_full2symm()

```
subroutine mqc_algebra::mqc_matrix_full2symm (  
    type(mqc_matrix), intent(inout) Matrix )
```

#### 5.1.1.55 mqc\_matrix\_generalized\_eigensystem()

```
subroutine mqc_algebra::mqc_matrix_generalized_eigensystem (  
    class(mqc_matrix), intent(inout) a,  
    type(mqc_matrix), intent(inout), optional bIn,  
    type(mqc_vector), intent(out), optional eigenvals,  
    type(mqc_matrix), intent(out), optional reigenvecs,  
    type(mqc_matrix), intent(out), optional leigenvecs )
```

#### 5.1.1.56 mqc\_matrix\_havecomplex()

```
logical function mqc_algebra::mqc_matrix_havecomplex (  
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.57 mqc\_matrix\_havediagonal()

```
logical function mqc_algebra::mqc_matrix_havediagonal (  
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.58 mqc\_matrix\_havefull()

```
logical function mqc_algebra::mqc_matrix_havefull (  
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.59 mqc\_matrix\_haveinteger()

```
logical function mqc_algebra::mqc_matrix_haveinteger (  
    type(mqc_matrix), intent(in) Matrix )
```



**5.1.1.60 mqc\_matrix\_havereal()**

```
logical function mqc_algebra::mqc_matrix_havereal (  
    type(mqc_matrix), intent(in) Matrix )
```

**5.1.1.61 mqc\_matrix\_havesymmetric()**

```
logical function mqc_algebra::mqc_matrix_havesymmetric (  
    type(mqc_matrix), intent(in) Matrix )
```

**5.1.1.62 mqc\_matrix\_identity()**

```
subroutine mqc_algebra::mqc_matrix_identity (  
    class(mqc_matrix), intent(inout) matrix,  
    integer(kind=int64) n,  
    integer(kind=int64) m )
```

**5.1.1.63 mqc\_matrix\_initialize()**

```
subroutine mqc_algebra::mqc_matrix_initialize (  
    class(mqc_matrix), intent(inout) Matrix,  
    integer(kind=int64), intent(in) Rows,  
    integer(kind=int64), intent(in) Columns,  
    class(*), optional Scalar,  
    character(len=*), intent(in), optional Storage )
```

**5.1.1.64 mqc\_matrix\_inverse()**

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_inverse (  
    class(mqc_matrix) a )
```

**5.1.1.65 mqc\_matrix\_isallocated()**

```
logical function mqc_algebra::mqc_matrix_isallocated (  
    class(mqc_matrix), intent(inout) Matrix )
```

#### 5.1.1.66 `mqc_matrix_matrix_at()`

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_matrix_at (  
    class(mqc_matrix), intent(in) Mat,  
    integer(kind=int64), dimension(:), intent(in) Rows,  
    integer(kind=int64), dimension(:), intent(in) Cols )
```

**MQC\_Matrix\_Matrix\_At** is a function that returns a submatrix of the matrix

## Parameters

in	<i>Mat</i>	<p>Mat is Class(MQC_Matrix)</p> <p>Name of the input matrix variable</p>
in	<i>rows</i>	<p>Rows is Integer(kind=int64),Dimension(:)</p> <p>If = [A,B]: output is submatrix of rows A to B</p> <p>If (A,B)&gt;0 row count is from first index</p> <p>If (A,B)&lt;0 row count is from last index</p> <p>If = [0]: submatrix of rows equivalent to [1,-1]</p>
in	<i>Cols</i>	<p>Cols is Integer(kind=int64),Dimension(:)</p> <p>If = [A,B]: output is submatrix of columns A to B</p> <p>If (A,B)&gt;0 column count is from first index</p> <p>If (A,B)&lt;0 column count is from last index</p> <p>If = [0]: submatrix of columns equivalent to [1,-1]</p>

## Author

L. M. Thompson

## Date

2017

## 5.1.1.67 mqc\_matrix\_matrix\_contraction()

```

type(mqc_scalar) function mqc_algebra::mqc_matrix_matrix_contraction (
    type(mqc_matrix), intent(in) Matrix1,
    type(mqc_matrix), intent(in) Matrix2 )

```

## 5.1.1.68 mqc\_matrix\_matrix\_put()

```

recursive subroutine mqc_algebra::mqc_matrix_matrix_put (
    class(mqc_matrix), intent(inout) Mat,
    type(mqc_matrix), intent(in) MatrixIn,
    integer(kind=int64), dimension(:), intent(in) Rows,
    integer(kind=int64), dimension(:), intent(in) Cols )

```

#### 5.1.1.69 mqc\_matrix\_norm()

```
type(mqc_scalar) function mqc_algebra::mqc_matrix_norm (
    class(mqc_matrix), intent(inout) matrix,
    character(len=1), intent(in), optional methodIn )
```

#### 5.1.1.70 mqc\_matrix\_rms\_max()

```
subroutine mqc_algebra::mqc_matrix_rms_max (
    class(mqc_matrix), intent(inout) A,
    type(mqc_scalar), intent(out) rms_A,
    type(mqc_scalar), intent(out) max_A )
```

#### 5.1.1.71 mqc\_matrix\_rows()

```
integer(kind=int64) function mqc_algebra::mqc_matrix_rows (
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.72 mqc\_matrix\_scalar\_at()

```
type(mqc_scalar) function mqc_algebra::mqc_matrix_scalar_at (
    class(mqc_matrix), intent(in) Mat,
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in) J )
```

#### 5.1.1.73 mqc\_matrix\_scalar\_put()

```
subroutine mqc_algebra::mqc_matrix_scalar_put (
    class(mqc_matrix), intent(inout) Matrix,
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in) J )
```

#### 5.1.1.74 mqc\_matrix\_set()

```
subroutine mqc_algebra::mqc_matrix_set (
    class(mqc_matrix), intent(inout) matrix,
    class(*), optional scalar,
    character(len=*), intent(in), optional storage )
```

#### 5.1.1.75 mqc\_matrix\_sqrt()

```
subroutine mqc_algebra::mqc_matrix_sqrt (
    class(mqc_matrix), intent(inout) A,
    type(mqc_vector), intent(inout), optional eVals,
    type(mqc_matrix), intent(inout), optional eVecs )
```

#### 5.1.1.76 mqc\_matrix\_storagetype()

```
character(len=64) function mqc_algebra::mqc_matrix_storagetype (
    class(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.77 mqc\_matrix\_svd()

```
subroutine mqc_algebra::mqc_matrix_svd (
    class(mqc_matrix), intent(inout) A,
    type(mqc_vector), intent(inout), optional EVals,
    type(mqc_matrix), intent(inout), optional EUVecs,
    type(mqc_matrix), intent(inout), optional EVVecs )
```

#### 5.1.1.78 mqc\_matrix\_symm2diag()

```
subroutine mqc_algebra::mqc_matrix_symm2diag (
    type(mqc_matrix), intent(inout) Matrix )
```

#### 5.1.1.79 mqc\_matrix\_symm2full()

```
subroutine mqc_algebra::mqc_matrix_symm2full (
    type(mqc_matrix), intent(inout) Matrix,
    character(len=*), intent(in), optional Option )
```

#### 5.1.1.80 mqc\_matrix\_symm2full\_func()

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_symm2full_func (
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.81 mqc\_matrix\_symmetrize()

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_symmetrize (
    type(mqc_matrix), intent(in) Matrix )
```

#### 5.1.1.82 mqc\_matrix\_symmmatrix\_put\_complex()

```
subroutine mqc_algebra::mqc_matrix_symmmatrix_put_complex (
    class(mqc_matrix), intent(inout) mat,
    complex(kind=real64), dimension(:), intent(in) symmMatrixIn )
```

#### 5.1.1.83 mqc\_matrix\_symmmatrix\_put\_integer()

```
subroutine mqc_algebra::mqc_matrix_symmmatrix_put_integer (
    class(mqc_matrix), intent(inout) mat,
    integer(kind=int64), dimension(:), intent(in) symmMatrixIn )
```

#### 5.1.1.84 mqc\_matrix\_symmmatrix\_put\_real()

```
subroutine mqc_algebra::mqc_matrix_symmmatrix_put_real (
    class(mqc_matrix), intent(inout) mat,
    real(kind=real64), dimension(:), intent(in) symmMatrixIn )
```

**5.1.1.85 mqc\_matrix\_symmsymmr4tensor\_put\_complex()**

```
subroutine mqc_algebra::mqc_matrix_symmsymmr4tensor_put_complex (
    class(mqc_r4tensor), intent(inout) r4Tensor,
    complex(kind=real64), dimension(:), intent(in) symmSymmMatrixIn )
```

**5.1.1.86 mqc\_matrix\_symmsymmr4tensor\_put\_real()**

```
subroutine mqc_algebra::mqc_matrix_symmsymmr4tensor_put_real (
    class(mqc_r4tensor), intent(inout) r4Tensor,
    real(kind=real64), dimension(:), intent(in) symmSymmMatrixIn )
```

**5.1.1.87 mqc\_matrix\_test\_diagonal()**

```
logical function mqc_algebra::mqc_matrix_test_diagonal (
    class(mqc_matrix), intent(in) Matrix )
```

**5.1.1.88 mqc\_matrix\_test\_symmetric()**

```
logical function mqc_algebra::mqc_matrix_test_symmetric (
    class(mqc_matrix), intent(in) Matrix,
    character(len=*), intent(in), optional Option )
```

**5.1.1.89 mqc\_matrix\_trace()**

```
type(mqc_scalar) function mqc_algebra::mqc_matrix_trace (
    class(mqc_matrix), intent(in) matrix )
```

**5.1.1.90 mqc\_matrix\_transpose()**

```
type(mqc_matrix) function mqc_algebra::mqc_matrix_transpose (
    class(mqc_matrix), intent(in) Matrix )
```

**5.1.1.91 mqc\_matrix\_vector\_at()**

```

type(mqc_vector) function mqc_algebra::mqc_matrix_vector_at (
    class(mqc_matrix), intent(in) Mat,
    integer(kind=int64), dimension(:), intent(in) Rows,
    integer(kind=int64), dimension(:), intent(in) Cols )

```

**5.1.1.92 mqc\_matrix\_vector\_put()**

```

recursive subroutine mqc_algebra::mqc_matrix_vector_put (
    class(mqc_matrix), intent(inout) Mat,
    type(mqc_vector), intent(in) VectorIn,
    integer(kind=int64), dimension(:), intent(in) Rows,
    integer(kind=int64), dimension(:), intent(in) Cols )

```

**5.1.1.93 mqc\_matrixmatrixdotproduct()**

```

type(mqc_matrix) function mqc_algebra::mqc_matrixmatrixdotproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )

```

**5.1.1.94 mqc\_matrixmatrixproduct()**

```

type(mqc_matrix) function mqc_algebra::mqc_matrixmatrixproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )

```

**5.1.1.95 mqc\_matrixmatrixsubtract()**

```

type(mqc_matrix) function mqc_algebra::mqc_matrixmatrixsubtract (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )

```



**5.1.1.96 mqc\_matrixmatrixsum()**

```
type(mqc_matrix) function mqc_algebra::mqc_matrixmatrixsum (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )
```

**5.1.1.97 mqc\_matrixscalarproduct()**

```
type(mqc_matrix) function mqc_algebra::mqc_matrixscalarproduct (
    type(mqc_matrix), intent(in) Matrix,
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.98 mqc\_matrixvectordotproduct()**

```
type(mqc_vector) function mqc_algebra::mqc_matrixvectordotproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_vector), intent(in) VB )
```

**5.1.1.99 mqc\_outer()**

```
type(mqc_matrix) function mqc_algebra::mqc_outer (
    type(mqc_vector), intent(in) VA,
    type(mqc_vector), intent(in) VB )
```

**5.1.1.100 mqc\_output\_complex\_scalar()**

```
subroutine mqc_algebra::mqc_output_complex_scalar (
    complex(kind=real64), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )
```

**5.1.1.101 mqc\_output\_integer\_scalar()**

```
subroutine mqc_algebra::mqc_output_integer_scalar (
    integer(kind=int64), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )
```

**5.1.1.102 mqc\_output\_mqcscalar\_scalar()**

```

subroutine mqc_algebra::mqc_output_mqcscalar_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )

```

**5.1.1.103 mqc\_output\_real\_scalar()**

```

subroutine mqc_algebra::mqc_output_real_scalar (
    real(kind=real64), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )

```

**5.1.1.104 mqc\_print\_matrix\_algebra1()**

```

subroutine mqc_algebra::mqc_print_matrix_algebra1 (
    class(mqc_matrix), intent(in) Matrix,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in) Header,
    logical, intent(in), optional Blank_At_Top,
    logical, intent(in), optional Blank_At_Bottom )

```

**5.1.1.105 mqc\_print\_r4tensor\_algebra1()**

```

subroutine mqc_algebra::mqc_print_r4tensor_algebra1 (
    class(mqc_r4tensor), intent(in) Tensor,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in), optional Header,
    logical, optional blank_at_top,
    logical, optional blank_at_bottom )

```

**5.1.1.106 mqc\_print\_scalar\_algebra1()**

```

subroutine mqc_algebra::mqc_print_scalar_algebra1 (
    class(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in) Header,
    logical, intent(in), optional Blank_At_Top,
    logical, intent(in), optional Blank_At_Bottom )

```

**5.1.1.107 mqc\_print\_vector\_algebra1()**

```
subroutine mqc_algebra::mqc_print_vector_algebra1 (
    class(mqc_vector), intent(in) Vector,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in) Header,
    logical, intent(in), optional Verbose,
    logical, intent(in), optional Blank_At_Top,
    logical, intent(in), optional Blank_At_Bottom )
```

**5.1.1.108 mqc\_r4tensor\_at()**

```
type(mqc_scalar) function mqc_algebra::mqc_r4tensor_at (
    class(mqc_r4tensor), intent(in) Tensor,
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in) J,
    integer(kind=int64), intent(in) K,
    integer(kind=int64), intent(in) L )
```

**5.1.1.109 mqc\_r4tensor\_havecomplex()**

```
logical function mqc_algebra::mqc_r4tensor_havecomplex (
    type(mqc_r4tensor), intent(in) R4Tensor )
```

**5.1.1.110 mqc\_r4tensor\_haveinteger()**

```
logical function mqc_algebra::mqc_r4tensor_haveinteger (
    type(mqc_r4tensor), intent(in) R4Tensor )
```

**5.1.1.111 mqc\_r4tensor\_havereal()**

```
logical function mqc_algebra::mqc_r4tensor_havereal (
    type(mqc_r4tensor), intent(in) R4Tensor )
```

**5.1.1.112 mqc\_r4tensor\_initialize()**

```

subroutine mqc_algebra::mqc_r4tensor_initialize (
    class(mqc_r4tensor), intent(inout) R4Tensor,
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in) J,
    integer(kind=int64), intent(in) K,
    integer(kind=int64), intent(in) L,
    class(*), optional Scalar )

```

**5.1.1.113 mqc\_r4tensor\_put()**

```

subroutine mqc_algebra::mqc_r4tensor_put (
    class(mqc_r4tensor), intent(inout) Tensor,
    type(mqc_scalar), intent(in) Element,
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in) J,
    integer(kind=int64), intent(in) K,
    integer(kind=int64), intent(in) L )

```

**5.1.1.114 mqc\_realgtscalar()**

```

logical function mqc_algebra::mqc_realgtscalar (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )

```

**5.1.1.115 mqc\_reallscalar()**

```

logical function mqc_algebra::mqc_reallscalar (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )

```

**5.1.1.116 mqc\_realltscalar()**

```

logical function mqc_algebra::mqc_realltscalar (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )

```

**5.1.1.117 mqc\_realscalaradd()**

```
type(mqc_scalar) function mqc_algebra::mqc_realscalaradd (  
    real(kind=real64), intent(in) RealIn,  
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.118 mqc\_realscalardivide()**

```
type(mqc_scalar) function mqc_algebra::mqc_realscalardivide (  
    real(kind=real64), intent(in) RealIn,  
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.119 mqc\_realscalarmultiply()**

```
type(mqc_scalar) function mqc_algebra::mqc_realscalarmultiply (  
    real(kind=real64), intent(in) RealIn,  
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.120 mqc\_realscalarsubtract()**

```
type(mqc_scalar) function mqc_algebra::mqc_realscalarsubtract (  
    real(kind=real64), intent(in) RealIn,  
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.121 mqc\_realvectorproduct()**

```
type(mqc_vector) function mqc_algebra::mqc_realvectorproduct (  
    real(kind=real64), intent(in) RealIn,  
    type(mqc_vector), intent(in) Vector )
```

**5.1.1.122 mqc\_scalar\_acos()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_acos (  
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.123 mqc\_scalar\_asin()

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_asin (  
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.124 mqc\_scalar\_atan()

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_atan (  
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.125 mqc\_scalar\_atan2()

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_atan2 (  
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.126 mqc\_scalar\_cmplx()

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_cmplx (  
    type(mqc_scalar), intent(in) Scalar1,  
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 5.1.1.127 mqc\_scalar\_complex\_conjugate()

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_complex_conjugate (  
    type(mqc_scalar), intent(in) ScalarIn )
```

#### 5.1.1.128 mqc\_scalar\_complex\_imagpart()

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_complex_imagpart (  
    type(mqc_scalar), intent(in) ScalarIn )
```

**5.1.1.129 mqc\_scalar\_complex\_realpart()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_complex_realpart (  
    type(mqc_scalar), intent(in) ScalarIn )
```

**5.1.1.130 mqc\_scalar\_cos()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_cos (  
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.131 mqc\_scalar\_get\_abs\_value()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_get_abs_value (  
    class(mqc_scalar), intent(in) Scalar )
```

**5.1.1.132 mqc\_scalar\_get\_intrinsic\_complex()**

```
complex(kind=real64) function mqc_algebra::mqc_scalar_get_intrinsic_complex (  
    class(mqc_scalar), intent(in) Scalar )
```

**5.1.1.133 mqc\_scalar\_get\_intrinsic\_integer()**

```
integer(kind=int64) function mqc_algebra::mqc_scalar_get_intrinsic_integer (  
    class(mqc_scalar), intent(in) Scalar )
```

**5.1.1.134 mqc\_scalar\_get\_intrinsic\_real()**

```
real(kind=real64) function mqc_algebra::mqc_scalar_get_intrinsic_real (  
    class(mqc_scalar), intent(in) Scalar )
```

**5.1.1.135 mqc\_scalar\_get\_random\_value()**

```
subroutine mqc_algebra::mqc_scalar_get_random_value (
    class(mqc_scalar) Scalar )
```

**5.1.1.136 mqc\_scalar\_havecomplex()**

```
logical function mqc_algebra::mqc_scalar_havecomplex (
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.137 mqc\_scalar\_haveinteger()**

```
logical function mqc_algebra::mqc_scalar_haveinteger (
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.138 mqc\_scalar\_havereal()**

```
logical function mqc_algebra::mqc_scalar_havereal (
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.139 mqc\_scalar\_isallocated()**

```
logical function mqc_algebra::mqc_scalar_isallocated (
    type(mqc_scalar), intent(inout) Scalar )
```

**5.1.1.140 mqc\_scalar\_sin()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_sin (
    type(mqc_scalar), intent(in) Scalar )
```



**5.1.1.141 mqc\_scalar\_sqrt()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_sqrt (
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.142 mqc\_scalar\_tan()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalar_tan (
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.143 mqc\_scalaradd()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalaradd (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

**5.1.1.144 mqc\_scalarcomplexadd()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarcomplexadd (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

**5.1.1.145 mqc\_scalarcomplexdivide()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarcomplexdivide (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

**5.1.1.146 mqc\_scalarcomplexexponent()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarcomplexexponent (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) CompIn )
```

#### 5.1.1.147 mqc\_scalarcomplexmultiply()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarcomplexmultiply (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

#### 5.1.1.148 mqc\_scalarcomplexsubtract()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarcomplexsubtract (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

#### 5.1.1.149 mqc\_scalardivide()

```
type(mqc_scalar) function mqc_algebra::mqc_scalardivide (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 5.1.1.150 mqc\_scalareq()

```
logical function mqc_algebra::mqc_scalareq (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 5.1.1.151 mqc\_scalarexponent()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarexponent (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 5.1.1.152 mqc\_scalarge()

```
logical function mqc_algebra::mqc_scalarge (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

**5.1.1.153 mqc\_scalargt()**

```
logical function mqc_algebra::mqc_scalargt (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

**5.1.1.154 mqc\_scalargtinteger()**

```
logical function mqc_algebra::mqc_scalargtinteger (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntIn )
```

**5.1.1.155 mqc\_scalargtreal()**

```
logical function mqc_algebra::mqc_scalargtreal (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

**5.1.1.156 mqc\_scalarintegeradd()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarintegeradd (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

**5.1.1.157 mqc\_scalarintegerdivide()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarintegerdivide (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

**5.1.1.158 mqc\_scalarintegerexponent()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarintegerexponent (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntIn )
```

#### 5.1.1.159 mqc\_scalarintegermultiply()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarintegermultiply (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

#### 5.1.1.160 mqc\_scalarintegersubtract()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarintegersubtract (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

#### 5.1.1.161 mqc\_scalarle()

```
logical function mqc_algebra::mqc_scalarle (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 5.1.1.162 mqc\_scalarleinteger()

```
logical function mqc_algebra::mqc_scalarleinteger (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntIn )
```

#### 5.1.1.163 mqc\_scalarlereal()

```
logical function mqc_algebra::mqc_scalarlereal (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 5.1.1.164 mqc\_scalarlt()

```
logical function mqc_algebra::mqc_scalarlt (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

**5.1.1.165 mqc\_scalarltreal()**

```
logical function mqc_algebra::mqc_scalarltreal (  
    type(mqc_scalar), intent(in) Scalar,  
    real(kind=real64), intent(in) RealIn )
```

**5.1.1.166 mqc\_scalarmatrixproduct()**

```
type(mqc_matrix) function mqc_algebra::mqc_scalarmatrixproduct (  
    type(mqc_scalar), intent(in) Scalar,  
    type(mqc_matrix), intent(in) Matrix )
```

**5.1.1.167 mqc\_scalarmultiply()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarmultiply (  
    type(mqc_scalar), intent(in) Scalar1,  
    type(mqc_scalar), intent(in) Scalar2 )
```

**5.1.1.168 mqc\_scalarne()**

```
logical function mqc_algebra::mqc_scalarne (  
    type(mqc_scalar), intent(in) Scalar1,  
    type(mqc_scalar), intent(in) Scalar2 )
```

**5.1.1.169 mqc\_scalarrealadd()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarrealadd (  
    type(mqc_scalar), intent(in) Scalar,  
    real(kind=real64), intent(in) RealIn )
```

**5.1.1.170 mqc\_scalarrealddivide()**

```
type(mqc_scalar) function mqc_algebra::mqc_scalarrealddivide (  
    type(mqc_scalar), intent(in) Scalar,  
    real(kind=real64), intent(in) RealIn )
```

#### 5.1.1.171 mqc\_scalarrealexponent()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarrealexponent (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 5.1.1.172 mqc\_scalarrealmultiply()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarrealmultiply (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 5.1.1.173 mqc\_scalarrealsubtract()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarrealsubtract (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 5.1.1.174 mqc\_scalarsubtract()

```
type(mqc_scalar) function mqc_algebra::mqc_scalarsubtract (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 5.1.1.175 mqc\_scalarvectordifference()

```
type(mqc_vector) function mqc_algebra::mqc_scalarvectordifference (
    type(mqc_scalar), intent(in) ScalarIn,
    type(mqc_vector), intent(in) VectorIn )
```

#### 5.1.1.176 mqc\_scalarvectorproduct()

```
type(mqc_vector) function mqc_algebra::mqc_scalarvectorproduct (
    type(mqc_scalar), intent(in) Scalar,
    type(mqc_vector), intent(in) Vector )
```

**5.1.1.177 mqc\_scalarvectorsum()**

```
type(mqc_vector) function mqc_algebra::mqc_scalarvectorsum (
    type(mqc_scalar), intent(in) ScalarIn,
    type(mqc_vector), intent(in) VectorIn )
```

**5.1.1.178 mqc\_set\_array2tensor()**

```
subroutine mqc_algebra::mqc_set_array2tensor (
    type(mqc_r4tensor), intent(inout) TensorOut,
    class(*), dimension(:, :, :, :), intent(in) ArrayIn )
```

**5.1.1.179 mqc\_set\_array2vector\_complex()**

```
subroutine mqc_algebra::mqc_set_array2vector_complex (
    type(mqc_vector), intent(inout) VectorOut,
    complex(kind=real64), dimension(:), intent(in) ArrayIn )
```

**5.1.1.180 mqc\_set\_array2vector\_integer()**

```
subroutine mqc_algebra::mqc_set_array2vector_integer (
    type(mqc_vector), intent(inout) VectorOut,
    integer(kind=int64), dimension(:), intent(in) ArrayIn )
```

**5.1.1.181 mqc\_set\_array2vector\_real()**

```
subroutine mqc_algebra::mqc_set_array2vector_real (
    type(mqc_vector), intent(inout) VectorOut,
    real(kind=real64), dimension(:), intent(in) ArrayIn )
```

**5.1.1.182 mqc\_set\_complexarray2matrix()**

```
subroutine mqc_algebra::mqc_set_complexarray2matrix (
    type(mqc_matrix), intent(inout) MatrixOut,
    complex(kind=real64), dimension(:, :), intent(in) ArrayIn )
```

**5.1.1.183 mqc\_set\_integerarray2matrix()**

```
subroutine mqc_algebra::mqc_set_integerarray2matrix (  
    type(mqc_matrix), intent(inout) MatrixOut,  
    integer(kind=int64), dimension(:,:), intent(in) ArrayIn )
```

**5.1.1.184 mqc\_set\_matrix2complexarray()**

```
subroutine mqc_algebra::mqc_set_matrix2complexarray (  
    complex(kind=real64), dimension(:,:), intent(inout), allocatable ArrayOut,  
    type(mqc_matrix), intent(in) MatrixIn )
```

**5.1.1.185 mqc\_set\_matrix2integerarray()**

```
subroutine mqc_algebra::mqc_set_matrix2integerarray (  
    integer(kind=int64), dimension(:,:), intent(inout), allocatable ArrayOut,  
    type(mqc_matrix), intent(in) MatrixIn )
```

**5.1.1.186 mqc\_set\_matrix2matrix()**

```
subroutine mqc_algebra::mqc_set_matrix2matrix (  
    class(mqc_matrix), intent(inout) MatrixOut,  
    class(mqc_matrix), intent(in) MatrixIn )
```

**5.1.1.187 mqc\_set\_matrix2realarray()**

```
subroutine mqc_algebra::mqc_set_matrix2realarray (  
    real(kind=real64), dimension(:,:), intent(inout), allocatable ArrayOut,  
    type(mqc_matrix), intent(in) MatrixIn )
```

**5.1.1.188 mqc\_set\_realarray2matrix()**

```
subroutine mqc_algebra::mqc_set_realarray2matrix (  
    type(mqc_matrix), intent(inout) MatrixOut,  
    real(kind=real64), dimension(:,:), intent(in) ArrayIn )
```



**5.1.1.189 mqc\_set\_vector2complexarray()**

```
subroutine mqc_algebra::mqc_set_vector2complexarray (
    complex(kind=real64), dimension(:), intent(inout), allocatable ArrayOut,
    type(mqc_vector), intent(in) VectorIn )
```

**5.1.1.190 mqc\_set\_vector2integerarray()**

```
subroutine mqc_algebra::mqc_set_vector2integerarray (
    integer(kind=int64), dimension(:), intent(inout), allocatable ArrayOut,
    type(mqc_vector), intent(in) VectorIn )
```

**5.1.1.191 mqc\_set\_vector2realarray()**

```
subroutine mqc_algebra::mqc_set_vector2realarray (
    real(kind=real64), dimension(:), intent(inout), allocatable ArrayOut,
    type(mqc_vector), intent(in) VectorIn )
```

**5.1.1.192 mqc\_set\_vector2vector()**

```
subroutine mqc_algebra::mqc_set_vector2vector (
    class(mqc_vector), intent(inout) VectorOut,
    class(mqc_vector), intent(in) VectorIn )
```

**5.1.1.193 mqc\_vector2diagmatrix()**

```
type(mqc_matrix) function mqc_algebra::mqc_vector2diagmatrix (
    type(mqc_vector), intent(in) vector )
```

**5.1.1.194 mqc\_vector\_abs()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_abs (
    class(mqc_vector), intent(in) A )
```

**5.1.1.195 mqc\_vector\_argsort()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_argsort (
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.196 mqc\_vector\_cast\_complex()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_cast_complex (
    type(mqc_vector), intent(in) VA )
```

**5.1.1.197 mqc\_vector\_cast\_real()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_cast_real (
    type(mqc_vector), intent(in) VA )
```

**5.1.1.198 mqc\_vector\_cmplx()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_cmplx (
    type(mqc_vector), intent(in) Vector1,
    type(mqc_vector), intent(in) Vector2 )
```

**5.1.1.199 mqc\_vector\_complex\_imagpart()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_complex_imagpart (
    class(mqc_vector), intent(in) A )
```

**5.1.1.200 mqc\_vector\_complex\_realpart()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_complex_realpart (
    class(mqc_vector), intent(in) A )
```

**5.1.1.201 mqc\_vector\_conjugate\_transpose()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_conjugate_transpose (  
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.202 mqc\_vector\_copy\_complex2int()**

```
subroutine mqc_algebra::mqc_vector_copy_complex2int (  
    type(mqc_vector) Vector )
```

**5.1.1.203 mqc\_vector\_copy\_complex2real()**

```
subroutine mqc_algebra::mqc_vector_copy_complex2real (  
    type(mqc_vector) Vector )
```

**5.1.1.204 mqc\_vector\_copy\_int2complex()**

```
subroutine mqc_algebra::mqc_vector_copy_int2complex (  
    type(mqc_vector) Vector )
```

**5.1.1.205 mqc\_vector\_copy\_int2real()**

```
subroutine mqc_algebra::mqc_vector_copy_int2real (  
    type(mqc_vector) Vector )
```

**5.1.1.206 mqc\_vector\_copy\_real2complex()**

```
subroutine mqc_algebra::mqc_vector_copy_real2complex (  
    type(mqc_vector) Vector )
```

**5.1.1.207 mqc\_vector\_copy\_real2int()**

```
subroutine mqc_algebra::mqc_vector_copy_real2int (  
    type(mqc_vector) Vector )
```

**5.1.1.208 mqc\_vector\_havecomplex()**

```
logical function mqc_algebra::mqc_vector_havecomplex (  
    type(mqc_vector), intent(in) Vector )
```

**5.1.1.209 mqc\_vector\_haveinteger()**

```
logical function mqc_algebra::mqc_vector_haveinteger (  
    type(mqc_vector), intent(in) Vector )
```

**5.1.1.210 mqc\_vector\_havereal()**

```
logical function mqc_algebra::mqc_vector_havereal (  
    type(mqc_vector), intent(in) Vector )
```

**5.1.1.211 mqc\_vector\_initialize()**

```
subroutine mqc_algebra::mqc_vector_initialize (  
    class(mqc_vector), intent(inout) Vector,  
    integer(kind=int64), intent(in) Length,  
    class(*), optional Scalar )
```

**5.1.1.212 mqc\_vector\_isallocated()**

```
logical function mqc_algebra::mqc_vector_isallocated (  
    class(mqc_vector), intent(inout) Vector )
```

**5.1.1.213 mqc\_vector\_iscolumn()**

```
logical function mqc_algebra::mqc_vector_iscolumn (  
    type(mqc_vector), intent(in) Vector )
```

**5.1.1.214 mqc\_vector\_maxloc()**

```
integer function mqc_algebra::mqc_vector_maxloc (  
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.215 mqc\_vector\_maxval()**

```
type(mqc_scalar) function mqc_algebra::mqc_vector_maxval (  
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.216 mqc\_vector\_minloc()**

```
integer function mqc_algebra::mqc_vector_minloc (  
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.217 mqc\_vector\_minval()**

```
type(mqc_scalar) function mqc_algebra::mqc_vector_minval (  
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.218 mqc\_vector\_norm()**

```
type(mqc_scalar) function mqc_algebra::mqc_vector_norm (  
    class(mqc_vector), intent(inout) vector,  
    character(len=1), intent(in), optional methodIn )
```

#### 5.1.1.219 mqc\_vector\_pop()

```
type(mqc_scalar) function mqc_algebra::mqc_vector_pop (
    class(mqc_vector), intent(inout) Vector )
```

#### 5.1.1.220 mqc\_vector\_power()

```
subroutine mqc_algebra::mqc_vector_power (
    class(mqc_vector), intent(inout) A,
    class(*) P )
```

#### 5.1.1.221 mqc\_vector\_push()

```
subroutine mqc_algebra::mqc_vector_push (
    class(mqc_vector), intent(inout) Vector,
    type(mqc_scalar), intent(in) Scalar )
```

#### 5.1.1.222 mqc\_vector\_scalar\_at()

```
type(mqc_scalar) function mqc_algebra::mqc_vector_scalar_at (
    class(mqc_vector) Vec,
    integer(kind=int64), intent(in) I )
```

#### 5.1.1.223 mqc\_vector\_scalar\_increment()

```
subroutine mqc_algebra::mqc_vector_scalar_increment (
    class(mqc_vector), intent(inout) Vector,
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) I )
```

#### 5.1.1.224 mqc\_vector\_scalar\_put()

```
subroutine mqc_algebra::mqc_vector_scalar_put (
    class(mqc_vector), intent(inout) Vector,
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) I )
```

**5.1.1.225 mqc\_vector\_shift()**

```
type(mqc_scalar) function mqc_algebra::mqc_vector_shift (
    class(mqc_vector), intent(inout) Vector )
```

**5.1.1.226 mqc\_vector\_sort()**

```
subroutine mqc_algebra::mqc_vector_sort (
    class(mqc_vector), intent(inout) Vector,
    type(mqc_vector), intent(in), optional idx )
```

**5.1.1.227 mqc\_vector\_sqrt()**

```
subroutine mqc_algebra::mqc_vector_sqrt (
    class(mqc_vector), intent(inout) A )
```

**5.1.1.228 mqc\_vector\_transpose()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_transpose (
    class(mqc_vector), intent(in) Vector )
```

**5.1.1.229 mqc\_vector\_unshift()**

```
subroutine mqc_algebra::mqc_vector_unshift (
    class(mqc_vector), intent(inout) Vector,
    type(mqc_scalar), intent(in) Scalar )
```

**5.1.1.230 mqc\_vector\_vector\_at()**

```
type(mqc_vector) function mqc_algebra::mqc_vector_vector_at (
    class(mqc_vector) Vec,
    integer(kind=int64), intent(in) I,
    integer(kind=int64), intent(in), optional J )
```

#### 5.1.1.231 mqc\_vector\_vector\_put()

```
subroutine mqc_algebra::mqc_vector_vector_put (
    class(mqc_vector), intent(inout) Vector,
    type(mqc_vector), intent(in) VectorIn,
    integer(kind=int64), intent(in), optional I )
```

#### 5.1.1.232 mqc\_vectorcomplexdivide()

```
type(mqc_vector) function mqc_algebra::mqc_vectorcomplexdivide (
    type(mqc_vector), intent(in) vector,
    complex(kind=real64), intent(in) compIn )
```

#### 5.1.1.233 mqc\_vectorcomplexproduct()

```
type(mqc_vector) function mqc_algebra::mqc_vectorcomplexproduct (
    type(mqc_vector), intent(in) vector,
    complex(kind=real64), intent(in) compIn )
```

#### 5.1.1.234 mqc\_vectorintegerdivide()

```
type(mqc_vector) function mqc_algebra::mqc_vectorintegerdivide (
    type(mqc_vector), intent(in) vector,
    integer(kind=int64), intent(in) intIn )
```

#### 5.1.1.235 mqc\_vectorintegerproduct()

```
type(mqc_vector) function mqc_algebra::mqc_vectorintegerproduct (
    type(mqc_vector), intent(in) vector,
    integer(kind=int64), intent(in) intIn )
```

#### 5.1.1.236 mqc\_vectormatrixdotproduct()

```
type(mqc_vector) function mqc_algebra::mqc_vectormatrixdotproduct (
    type(mqc_vector), intent(in) VA,
    type(mqc_matrix), intent(in) MB )
```



**5.1.1.237 mqc\_vectorrealdivide()**

```
type(mqc_vector) function mqc_algebra::mqc_vectorrealdivide (
    type(mqc_vector), intent(in) vector,
    real(kind=real64), intent(in) realIn )
```

**5.1.1.238 mqc\_vectorrealproduct()**

```
type(mqc_vector) function mqc_algebra::mqc_vectorrealproduct (
    type(mqc_vector), intent(in) vector,
    real(kind=real64), intent(in) realIn )
```

**5.1.1.239 mqc\_vectorscalardivide()**

```
type(mqc_vector) function mqc_algebra::mqc_vectorscalardivide (
    type(mqc_vector), intent(in) vector,
    type(mqc_scalar), intent(in) scalar )
```

**5.1.1.240 mqc\_vectorscalarproduct()**

```
type(mqc_vector) function mqc_algebra::mqc_vectorscalarproduct (
    type(mqc_vector), intent(in) vector,
    type(mqc_scalar), intent(in) scalar )
```

**5.1.1.241 mqc\_vectorvectordifference()**

```
type(mqc_vector) function mqc_algebra::mqc_vectorvectordifference (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

**5.1.1.242 mqc\_vectorvectordotproduct()**

```
type(mqc_scalar) function mqc_algebra::mqc_vectorvectordotproduct (
    type(mqc_vector), intent(in) Vector1,
    type(mqc_vector), intent(in) Vector2 )
```

### 5.1.1.243 mqc\_vectorvectorsum()

```
type(mqc_vector) function mqc_algebra::mqc_vectorvectorsum (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

### 5.1.1.244 symindexhash()

```
integer(kind=int64) function mqc_algebra::symindexhash (
    integer(kind=int64), intent(in) i,
    integer(kind=int64), intent(in) j,
    integer(kind=int64), intent(in), optional k,
    integer(kind=int64), intent(in), optional l )
```

## 5.2 mqc\_est Module Reference

### Data Types

- interface [assignment\(=\)](#)
- interface [contraction](#)
- interface [dagger](#)
- interface [dot\\_product](#)
- interface [matmul](#)
- type [mqc\\_determinant](#)
- type [mqc\\_determinant\\_string](#)
- interface [mqc\\_matrix\\_undospinblockghf](#)
- interface [mqc\\_print](#)
- type [mqc\\_pscf\\_wavefunction](#)
- type [mqc\\_scf\\_eigenvalues](#)
- type [mqc\\_scf\\_integral](#)
- type [mqc\\_twoeris](#)
- type [mqc\\_wavefunction](#)
- interface [operator\(\\*\)](#)
- interface [operator\(+\)](#)
- interface [operator\(-\)](#)
- interface [transpose](#)

## Functions/Subroutines

- subroutine [mqc\\_print\\_wavefunction](#) (wavefunction, iOut, label)
- subroutine [mqc\\_print\\_integral](#) (integral, iOut, header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_print\\_eigenvalues](#) (eigenvalues, iOut, header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_print\\_twoeris](#) (twoERIs, iOut, header, blank\_at\_top, blank\_at\_bottom)
- logical function [mqc\\_integral\\_isallocated](#) (Integral)
- logical function [mqc\\_eigenvalues\\_isallocated](#) (Eigenvalues)
- logical function [mqc\\_integral\\_has\\_alpha](#) (integral)
- logical function [mqc\\_integral\\_has\\_beta](#) (integral)
- logical function [mqc\\_integral\\_has\\_alphabeta](#) (integral)
- logical function [mqc\\_integral\\_has\\_betaalpha](#) (integral)
- logical function [mqc\\_eigenvalues\\_has\\_alpha](#) (eigenvalues)
- logical function [mqc\\_eigenvalues\\_has\\_beta](#) (eigenvalues)
- character(len=64) function [mqc\\_integral\\_array\\_type](#) (integral)
- character(len=64) function [mqc\\_eigenvalues\\_array\\_type](#) (eigenvalues)
- character(len=64) function [mqc\\_integral\\_array\\_name](#) (integral)
- character(len=64) function [mqc\\_eigenvalues\\_array\\_name](#) (eigenvalues)
- subroutine [mqc\\_integral\\_add\\_name](#) (integral, arrayName)
- subroutine [mqc\\_eigenvalues\\_add\\_name](#) (eigenvalues, arrayName)
- integer(kind=int64) function [mqc\\_integral\\_dimension](#) (integral, label, axis)
- integer(kind=int64) function [mqc\\_eigenvalues\\_dimension](#) (eigenvalues, label)
- subroutine [mqc\\_twoeris\\_allocate](#) (twoERIs, storageType, integralType, alpha, beta, alphaBeta, betaAlpha)
- subroutine [mqc\\_integral\\_allocate](#) (integral, arrayName, arrayType, alpha, beta, alphaBeta, betaAlpha)
- subroutine [mqc\\_eigenvalues\\_allocate](#) (eigenvalues, arrayName, arrayType, alpha, beta)
- subroutine [mqc\\_integral\\_identity](#) (integral, nAlpha, nBeta, label, nAlpha2, nBeta2)
- subroutine [mqc\\_integral\\_initialize](#) (integral, nAlpha, nBeta, scalar, label, nAlpha2, nBeta2)
- type(mqc\_matrix) function [mqc\\_integral\\_output\\_block](#) (integral, blockName)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_output\\_orbitals](#) (integral, orbString, alphaOrbsIn, betaOrbsIn, axis)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_swap\\_orbitals](#) (integral, alphaOrbsIn, betaOrbsIn, axis)
- type(mqc\_vector) function [mqc\\_eigenvalues\\_output\\_block](#) (eigenvalues, blockName)
- subroutine [mqc\\_integral\\_output\\_array](#) (matrixOut, integralln)
- subroutine [mqc\\_eigenvalues\\_output\\_array](#) (vectorOut, eigenvaluesIn)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_matrix\\_multiply](#) (integralA, matrixB, label)
- type(mqc\_scf\_integral) function [mqc\\_matrix\\_integral\\_multiply](#) (matrixA, integralB, label)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_sum](#) (integralA, integralB)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_difference](#) (integralA, integralB)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_integral\\_multiply](#) (integralA, integralB, label)
- type(mqc\_scf\_integral) function [mqc\\_scalar\\_integral\\_multiply](#) (scalar, integral)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_scalar\\_multiply](#) (integral, scalar)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_eigenvalues\\_multiply](#) (integralA, eigenvaluesB, label)
- type(mqc\_scf\_integral) function [mqc\\_eigenvalues\\_integral\\_multiply](#) (eigenvaluesA, integralB, label)
- type(mqc\_scf\_eigenvalues) function [mqc\\_eigenvalues\\_eigenvalues\\_multiply](#) (eigenvaluesA, eigenvaluesB, label)
- type(mqc\_scalar) function [mqc\\_eigenvalue\\_eigenvalue\\_dotproduct](#) (eigenvalueA, eigenvalueB)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_transpose](#) (integral, label)
- type(mqc\_scf\_integral) function [mqc\\_integral\\_conjugate\\_transpose](#) (integral, label)
- type(mqc\_scalar) function [mqc\\_integral\\_norm](#) (integral, methodIn)
- subroutine [mqc\\_matrix\\_spinblockghf](#) (array, nelec, multi, elist)
- subroutine [mqc\\_matrix\\_undospinblockghf\\_eigenvalues](#) (eigenvaluesIn, vectorOut)
- subroutine [mqc\\_matrix\\_undospinblockghf\\_integral](#) (integralln, matrixOut)
- type(mqc\_scalar) function [mqc\\_scf\\_integral\\_contraction](#) (integral1, integral2)

- type([mqc\\_scf\\_integral](#)) function [mqc\\_eri\\_integral\\_contraction](#) (eris, integral, label)
- subroutine [mqc\\_scf\\_integral\\_generalized\\_eigensystem](#) (integralA, integralB, eVals, rEVecs, IEVecs)
- subroutine [mqc\\_scf\\_integral\\_diagonalize](#) (integral, eVals, eVecs)
- type([mqc\\_scf\\_integral](#)) function [mqc\\_scf\\_integral\\_inverse](#) (integral)
- type(mqc\_scalar) function [mqc\\_scf\\_integral\\_trace](#) (integral)
- type(mqc\_scalar) function [mqc\\_scf\\_integral\\_determinant](#) (integral)
- subroutine [mqc\\_integral\\_set\\_energy\\_list](#) (integral, elist)
- integer(kind=int64) function, dimension(:), allocatable [mqc\\_integral\\_get\\_energy\\_list](#) (integral)
- subroutine [mqc\\_integral\\_delete\\_energy\\_list](#) (integral)
- subroutine [mqc\\_scf\\_eigenvalues\\_power](#) (eigenvalues, power)
- type(mqc\_scalar) function [mqc\\_twoeris\\_at](#) (twoERIs, i, j, k, l, spinBlock)
- type(mqc\_scalar) function [mqc\\_integral\\_at](#) (integral, i, j, spinBlock)
- type(mqc\_scalar) function [mqc\\_eigenvalues\\_at](#) (eigenvalues, i, spinBlock)
- subroutine [mqc\\_scf\\_transformation\\_matrix](#) (overlap, transform\_matrix, nBasUse)
- subroutine [gen\\_det\\_str](#) (IOut, IPrint, NBasisIn, NAlphaIn, NBetaIn, Determinants, NCoreIn)
- type(mqc\_scalar) function [slater\\_condon](#) (IOut, IPrint, NBasisIn, Determinants, L\_A\_String, L\_B\_String, R\_A\_String, R\_B\_String, Core\_Hamiltonian, ERIs, UHF)
- subroutine [twoeri\\_trans](#) (IOut, IPrint, MO\_Coeff, ERIs, MO\_ERIs, UHF)
- subroutine [mqc\\_build\\_ci\\_hamiltonian](#) (IOut, IPrint, NBasis, Determinants, MO\_Core\_Ham, MO\_ERIs, UHF, CI\_Hamiltonian)
- type(mqc\_matrix) function [get\\_one\\_gamma\\_matrix](#) (iOut, iPrint, nBasisIn, nState, determinants, ci\_amplitudes, nCoreIn, nOrbsIn)

## 5.2.1 Function/Subroutine Documentation

### 5.2.1.1 [gen\\_det\\_str\(\)](#)

```
subroutine mqc_est::gen_det_str (
    integer(kind=int64) IOut,
    integer(kind=int64) IPrint,
    type(mqc_scalar) NBasisIn,
    type(mqc_scalar) NAlphaIn,
    type(mqc_scalar) NBetaIn,
    type(mqc\_determinant) Determinants,
    type(mqc_scalar), optional NCoreIn )
```

### 5.2.1.2 [get\\_one\\_gamma\\_matrix\(\)](#)

```
type(mqc_matrix) function mqc_est::get_one_gamma_matrix (
    integer(kind=int64), intent(in) iOut,
    integer(kind=int64), intent(in) iPrint,
    type(mqc_scalar), intent(in) nBasisIn,
    integer(kind=int64), intent(in) nState,
    type(mqc\_determinant), intent(in) determinants,
    type(mqc_matrix), intent(in) ci_amplitudes,
    integer(kind=int64), intent(in), optional nCoreIn,
    integer(kind=int64), intent(in), optional nOrbsIn )
```

### 5.2.1.3 mqc\_build\_ci\_hamiltonian()

```
subroutine mqc_est::mqc_build_ci_hamiltonian (
    integer(kind=int64), intent(in) IOut,
    integer(kind=int64), intent(in) IPrint,
    type(mqc_scalar), intent(in) NBasis,
    type(mqc_determinant), intent(in) Determinants,
    type(mqc_scf_integral), intent(in) MO_Core_Ham,
    type(mqc_twoeris), intent(in) MO_ERIs,
    logical, intent(in) UHF,
    type(mqc_matrix), intent(out) CI_Hamiltonian )
```

### 5.2.1.4 mqc\_eigenvalue\_eigenvalue\_dotproduct()

```
type(mqc_scalar) function mqc_est::mqc_eigenvalue_eigenvalue_dotproduct (
    type(mqc_scf_eigenvalues), intent(in) eigenvalueA,
    type(mqc_scf_eigenvalues), intent(in) eigenvalueB )
```

### 5.2.1.5 mqc\_eigenvalues\_add\_name()

```
subroutine mqc_est::mqc_eigenvalues_add_name (
    class(mqc_scf_eigenvalues) eigenvalues,
    character(len=*) arrayName )
```

### 5.2.1.6 mqc\_eigenvalues\_allocate()

```
subroutine mqc_est::mqc_eigenvalues_allocate (
    class(mqc_scf_eigenvalues) eigenvalues,
    character(len=*) arrayName,
    character(len=*) arrayType,
    type(mqc_vector), optional alpha,
    type(mqc_vector), optional beta )
```

### 5.2.1.7 mqc\_eigenvalues\_array\_name()

```
character(len=64) function mqc_est::mqc_eigenvalues_array_name (
    class(mqc_scf_eigenvalues) eigenvalues )
```

### 5.2.1.8 mqc\_eigenvalues\_array\_type()

```
character(len=64) function mqc_est::mqc_eigenvalues_array_type (
    class(mqc_scf_eigenvalues) eigenvalues )
```

### 5.2.1.9 mqc\_eigenvalues\_at()

```
type(mqc_scalar) function mqc_est::mqc_eigenvalues_at (
    class(mqc_scf_eigenvalues) eigenvalues,
    integer(kind=int64) i,
    character(len=64), optional spinBlock )
```

### 5.2.1.10 mqc\_eigenvalues\_dimension()

```
integer(kind=int64) function mqc_est::mqc_eigenvalues_dimension (
    class(mqc_scf_eigenvalues), intent(in) eigenvalues,
    character(len=*), intent(in) label )
```

### 5.2.1.11 mqc\_eigenvalues\_eigenvalues\_multiply()

```
type(mqc_scf_eigenvalues) function mqc_est::mqc_eigenvalues_eigenvalues_multiply (
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesA,
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesB,
    character(len=*), intent(in), optional label )
```

### 5.2.1.12 mqc\_eigenvalues\_has\_alpha()

```
logical function mqc_est::mqc_eigenvalues_has_alpha (
    class(mqc_scf_eigenvalues) eigenvalues )
```

### 5.2.1.13 mqc\_eigenvalues\_has\_beta()

```
logical function mqc_est::mqc_eigenvalues_has_beta (
    class(mqc_scf_eigenvalues) eigenvalues )
```

#### 5.2.1.14 mqc\_eigenvalues\_integral\_multiply()

```
type(mqc_scf_integral) function mqc_est::mqc_eigenvalues_integral_multiply (
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesA,
    type(mqc_scf_integral), intent(in) integralB,
    character(len=*), intent(in), optional label )
```

#### 5.2.1.15 mqc\_eigenvalues\_isallocated()

```
logical function mqc_est::mqc_eigenvalues_isallocated (
    class(mqc_scf_eigenvalues), intent(inout) Eigenvalues )
```

#### 5.2.1.16 mqc\_eigenvalues\_output\_array()

```
subroutine mqc_est::mqc_eigenvalues_output_array (
    type(mqc_vector), intent(inout) vectorOut,
    class(mqc_scf_eigenvalues), intent(in) eigenvaluesIn )
```

#### 5.2.1.17 mqc\_eigenvalues\_output\_block()

```
type(mqc_vector) function mqc_est::mqc_eigenvalues_output_block (
    class(mqc_scf_eigenvalues) eigenvalues,
    character(len=*), optional blockName )
```

#### 5.2.1.18 mqc\_eri\_integral\_contraction()

```
type(mqc_scf_integral) function mqc_est::mqc_eri_integral_contraction (
    type(mqc_twoeris), intent(in) eris,
    type(mqc_scf_integral), intent(in) integral,
    character(len=*), optional label )
```

#### 5.2.1.19 mqc\_integral\_add\_name()

```
subroutine mqc_est::mqc_integral_add_name (
    class(mqc_scf_integral) integral,
    character(len=*) arrayName )
```

#### 5.2.1.20 `mqc_integral_allocate()`

```
subroutine mqc_est::mqc_integral_allocate (
    class(mqc_scf_integral) integral,
    character(len=*) arrayName,
    character(len=*) arrayType,
    type(mqc_matrix), optional alpha,
    type(mqc_matrix), optional beta,
    type(mqc_matrix), optional alphaBeta,
    type(mqc_matrix), optional betaAlpha )
```

#### 5.2.1.21 `mqc_integral_array_name()`

```
character(len=64) function mqc_est::mqc_integral_array_name (
    class(mqc_scf_integral) integral )
```

#### 5.2.1.22 `mqc_integral_array_type()`

```
character(len=64) function mqc_est::mqc_integral_array_type (
    class(mqc_scf_integral) integral )
```

#### 5.2.1.23 `mqc_integral_at()`

```
type(mqc_scalar) function mqc_est::mqc_integral_at (
    class(mqc_scf_integral) integral,
    integer(kind=int64) i,
    integer(kind=int64) j,
    character(len=64), optional spinBlock )
```

#### 5.2.1.24 `mqc_integral_conjugate_transpose()`

```
type(mqc_scf_integral) function mqc_est::mqc_integral_conjugate_transpose (
    type(mqc_scf_integral), intent(in) integral,
    character(len=*), intent(in), optional label )
```



#### 5.2.1.25 mqc\_integral\_delete\_energy\_list()

```
subroutine mqc_est::mqc_integral_delete_energy_list (
    class(mqc_scf_integral) integral )
```

#### 5.2.1.26 mqc\_integral\_difference()

```
type(mqc_scf_integral) function mqc_est::mqc_integral_difference (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), intent(in) integralB )
```

#### 5.2.1.27 mqc\_integral\_dimension()

```
integer(kind=int64) function mqc_est::mqc_integral_dimension (
    class(mqc_scf_integral), intent(in) integral,
    character(len=*), intent(in) label,
    integer(kind=int64), intent(in), optional axis )
```

#### 5.2.1.28 mqc\_integral\_eigenvalues\_multiply()

```
type(mqc_scf_integral) function mqc_est::mqc_integral_eigenvalues_multiply (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesB,
    character(len=*), intent(in), optional label )
```

#### 5.2.1.29 mqc\_integral\_get\_energy\_list()

```
integer(kind=int64) function, dimension(:), allocatable mqc_est::mqc_integral_get_energy_list (
    class(mqc_scf_integral) integral )
```

#### 5.2.1.30 mqc\_integral\_has\_alpha()

```
logical function mqc_est::mqc_integral_has_alpha (
    class(mqc_scf_integral) integral )
```

#### 5.2.1.31 mqc\_integral\_has\_alphabeta()

```
logical function mqc_est::mqc_integral_has_alphabeta (  
    class(mqc_scf_integral) integral )
```

#### 5.2.1.32 mqc\_integral\_has\_beta()

```
logical function mqc_est::mqc_integral_has_beta (  
    class(mqc_scf_integral) integral )
```

#### 5.2.1.33 mqc\_integral\_has\_betaalpha()

```
logical function mqc_est::mqc_integral_has_betaalpha (  
    class(mqc_scf_integral) integral )
```

#### 5.2.1.34 mqc\_integral\_identity()

```
subroutine mqc_est::mqc_integral_identity (  
    class(mqc_scf_integral), intent(inout) integral,  
    integer, intent(in) nAlpha,  
    integer, intent(in) nBeta,  
    character(len=*), intent(in), optional label,  
    integer, intent(in), optional nAlpha2,  
    integer, intent(in), optional nBeta2 )
```

#### 5.2.1.35 mqc\_integral\_initialize()

```
subroutine mqc_est::mqc_integral_initialize (  
    class(mqc_scf_integral), intent(inout) integral,  
    integer, intent(in) nAlpha,  
    integer, intent(in) nBeta,  
    class(*), intent(in), optional scalar,  
    character(len=*), intent(in), optional label,  
    integer, intent(in), optional nAlpha2,  
    integer, intent(in), optional nBeta2 )
```

### 5.2.1.36 mqc\_integral\_integral\_multiply()

```
type(mqc_scf_integral) function mqc_est::mqc_integral_integral_multiply (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), intent(in) integralB,
    character(len=*), intent(in), optional label )
```

### 5.2.1.37 mqc\_integral\_isallocated()

```
logical function mqc_est::mqc_integral_isallocated (
    class(mqc_scf_integral), intent(inout) Integral )
```

### 5.2.1.38 mqc\_integral\_matrix\_multiply()

```
type(mqc_scf_integral) function mqc_est::mqc_integral_matrix_multiply (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_matrix), intent(in) matrixB,
    character(len=*), intent(in), optional label )
```

### 5.2.1.39 mqc\_integral\_norm()

```
type(mqc_scalar) function mqc_est::mqc_integral_norm (
    class(mqc_scf_integral), intent(in) integral,
    character(len=1), intent(in), optional methodIn )
```

### 5.2.1.40 mqc\_integral\_output\_array()

```
subroutine mqc_est::mqc_integral_output_array (
    type(mqc_matrix), intent(inout) matrixOut,
    class(mqc_scf_integral), intent(in) integralIn )
```

### 5.2.1.41 mqc\_integral\_output\_block()

```
type(mqc_matrix) function mqc_est::mqc_integral_output_block (
    class(mqc_scf_integral) integral,
    character(len=*), optional blockName )
```

**5.2.1.42 mqc\_integral\_output\_orbitals()**

```

type(mqc_scf_integral) function mqc_est::mqc_integral_output_orbitals (
    class(mqc_scf_integral), intent(in) integral,
    character(len=*), optional orbString,
    integer(kind=int64), dimension(:), optional alphaOrbsIn,
    integer(kind=int64), dimension(:), optional betaOrbsIn,
    integer(kind=int64), intent(in), optional axis )

```

**5.2.1.43 mqc\_integral\_scalar\_multiply()**

```

type(mqc_scf_integral) function mqc_est::mqc_integral_scalar_multiply (
    type(mqc_scf_integral), intent(in) integral,
    type(mqc_scalar), intent(in) scalar )

```

**5.2.1.44 mqc\_integral\_set\_energy\_list()**

```

subroutine mqc_est::mqc_integral_set_energy_list (
    class(mqc_scf_integral) integral,
    integer(kind=int64), dimension(:), allocatable elist )

```

**5.2.1.45 mqc\_integral\_sum()**

```

type(mqc_scf_integral) function mqc_est::mqc_integral_sum (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), intent(in) integralB )

```

**5.2.1.46 mqc\_integral\_swap\_orbitals()**

```

type(mqc_scf_integral) function mqc_est::mqc_integral_swap_orbitals (
    class(mqc_scf_integral), intent(in) integral,
    integer(kind=int64), dimension(2), optional alphaOrbsIn,
    integer(kind=int64), dimension(2), optional betaOrbsIn,
    integer(kind=int64), intent(in), optional axis )

```

#### 5.2.1.47 mqc\_integral\_transpose()

```
type(mqc_scf_integral) function mqc_est::mqc_integral_transpose (
    type(mqc_scf_integral), intent(in) integral,
    character(len=*), intent(in), optional label )
```

#### 5.2.1.48 mqc\_matrix\_integral\_multiply()

```
type(mqc_scf_integral) function mqc_est::mqc_matrix_integral_multiply (
    type(mqc_matrix), intent(in) matrixA,
    type(mqc_scf_integral), intent(in) integralB,
    character(len=*), intent(in), optional label )
```

#### 5.2.1.49 mqc\_matrix\_spinblockghf()

```
subroutine mqc_est::mqc_matrix_spinblockghf (
    class(*), intent(inout) array,
    integer(kind=int64), optional nelec,
    integer(kind=int64), optional multi,
    integer(kind=int64), dimension(:), optional, allocatable elist )
```

#### 5.2.1.50 mqc\_matrix\_undospinblockghf\_eigenvalues()

```
subroutine mqc_est::mqc_matrix_undospinblockghf_eigenvalues (
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesIn,
    type(mqc_vector), intent(out) vectorOut )
```

#### 5.2.1.51 mqc\_matrix\_undospinblockghf\_integral()

```
subroutine mqc_est::mqc_matrix_undospinblockghf_integral (
    type(mqc_scf_integral), intent(in) integralIn,
    type(mqc_matrix), intent(out) matrixOut )
```

**5.2.1.52 mqc\_print\_eigenvalues()**

```

subroutine mqc_est::mqc_print_eigenvalues (
    class(mqc_scf_eigenvalues) eigenvalues,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in) header,
    logical, intent(in), optional blank_at_top,
    logical, intent(in), optional blank_at_bottom )

```

**5.2.1.53 mqc\_print\_integral()**

```

subroutine mqc_est::mqc_print_integral (
    class(mqc_scf_integral) integral,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in) header,
    logical, intent(in), optional blank_at_top,
    logical, intent(in), optional blank_at_bottom )

```

**5.2.1.54 mqc\_print\_twoeris()**

```

subroutine mqc_est::mqc_print_twoeris (
    class(mqc_twoeris) twoERIs,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in) header,
    logical, intent(in), optional blank_at_top,
    logical, intent(in), optional blank_at_bottom )

```

**5.2.1.55 mqc\_print\_wavefunction()**

```

subroutine mqc_est::mqc_print_wavefunction (
    class(mqc_wavefunction) wavefunction,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in), optional label )

```

**5.2.1.56 mqc\_scalar\_integral\_multiply()**

```

type(mqc_scf_integral) function mqc_est::mqc_scalar_integral_multiply (
    type(mqc_scalar), intent(in) scalar,
    type(mqc_scf_integral), intent(in) integral )

```

**5.2.1.57 mqc\_scf\_eigenvalues\_power()**

```
subroutine mqc_est::mqc_scf_eigenvalues_power (
    class(mqc_scf_eigenvalues), intent(inout) eigenvalues,
    class(*) power )
```

**5.2.1.58 mqc\_scf\_integral\_contraction()**

```
type(mqc_scalar) function mqc_est::mqc_scf_integral_contraction (
    type(mqc_scf_integral), intent(in) integral1,
    type(mqc_scf_integral), intent(in) integral2 )
```

**5.2.1.59 mqc\_scf\_integral\_determinant()**

```
type(mqc_scalar) function mqc_est::mqc_scf_integral_determinant (
    class(mqc_scf_integral), intent(in) integral )
```

**5.2.1.60 mqc\_scf\_integral\_diagonalize()**

```
subroutine mqc_est::mqc_scf_integral_diagonalize (
    class(mqc_scf_integral), intent(in) integral,
    type(mqc_scf_eigenvalues), intent(inout), optional eVals,
    type(mqc_scf_integral), intent(inout), optional eVecs )
```

**5.2.1.61 mqc\_scf\_integral\_generalized\_eigensystem()**

```
subroutine mqc_est::mqc_scf_integral_generalized_eigensystem (
    class(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), optional integralB,
    type(mqc_scf_eigenvalues), intent(inout), optional eVals,
    type(mqc_scf_integral), intent(inout), optional rEVecs,
    type(mqc_scf_integral), intent(inout), optional lEVecs )
```

**5.2.1.62 mqc\_scf\_integral\_inverse()**

```
type(mqc_scf_integral) function mqc_est::mqc_scf_integral_inverse (
    class(mqc_scf_integral), intent(in) integral )
```

**5.2.1.63 mqc\_scf\_integral\_trace()**

```
type(mqc_scalar) function mqc_est::mqc_scf_integral_trace (
    class(mqc_scf_integral), intent(in) integral )
```

**5.2.1.64 mqc\_scf\_transformation\_matrix()**

```
subroutine mqc_est::mqc_scf_transformation_matrix (
    type(mqc_scf_integral), intent(in) overlap,
    type(mqc_scf_integral), intent(out) transform_matrix,
    integer(kind=int64), intent(out), optional nBasUse )
```

**5.2.1.65 mqc\_twoeris\_allocate()**

```
subroutine mqc_est::mqc_twoeris_allocate (
    class(mqc_twoeris) twoERIs,
    character(len=*) storageType,
    character(len=*) integralType,
    type(mqc_r4tensor), optional alpha,
    type(mqc_r4tensor), optional beta,
    type(mqc_r4tensor), optional alphaBeta,
    type(mqc_r4tensor), optional betaAlpha )
```

**5.2.1.66 mqc\_twoeris\_at()**

```
type(mqc_scalar) function mqc_est::mqc_twoeris_at (
    class(mqc_twoeris) twoERIs,
    integer(kind=int64), intent(in) i,
    integer(kind=int64), intent(in) j,
    integer(kind=int64), intent(in) k,
    integer(kind=int64), intent(in) l,
    character(len=64), optional spinBlock )
```



### 5.2.1.67 slater\_condon()

```
type(mqc_scalar) function mqc_est::slater_condon (
    integer(kind=int64), intent(in) IOut,
    integer(kind=int64), intent(in) IPrint,
    type(mqc_scalar), intent(in) NBasisIn,
    type(mqc_determinant), intent(in) Determinants,
    integer(kind=int64), intent(in) L_A_String,
    integer(kind=int64), intent(in) L_B_String,
    integer(kind=int64), intent(in) R_A_String,
    integer(kind=int64), intent(in) R_B_String,
    type(mqc_scf_integral), intent(in) Core_Hamiltonian,
    type(mqc_twoeris), intent(in) ERIs,
    logical, intent(in) UHF )
```

### 5.2.1.68 twoeri\_trans()

```
subroutine mqc_est::twoeri_trans (
    integer(kind=int64) IOut,
    integer(kind=int64) IPrint,
    type(mqc_scf_integral), intent(in) MO_Coeff,
    type(mqc_twoeris), intent(in) ERIs,
    type(mqc_twoeris), intent(out) MO_ERIs,
    logical UHF )
```



## Chapter 6

# Data Type Documentation

### 6.1 `mqc_algebra::abs` Interface Reference

#### Public Member Functions

- `type(mqc_scalar)` function `mqc_scalar_get_abs_value` (Scalar)
- `type(mqc_vector)` function `mqc_vector_abs` (A)

#### 6.1.1 Member Function/Subroutine Documentation

##### 6.1.1.1 `mqc_scalar_get_abs_value()`

```
type(mqc_scalar) function mqc_algebra::abs::mqc_scalar_get_abs_value (  
    class(mqc_scalar), intent(in) Scalar )
```

##### 6.1.1.2 `mqc_vector_abs()`

```
type(mqc_vector) function mqc_algebra::abs::mqc_vector_abs (  
    class(mqc_vector), intent(in) A )
```

The documentation for this interface was generated from the following file:

- `src/mqc_algebra.F03`

## 6.2 mqc\_algebra::acos Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_acos](#) (Scalar)

### 6.2.1 Member Function/Subroutine Documentation

#### 6.2.1.1 mqc\_scalar\_acos()

```
type(mqc\_scalar) function mqc_algebra::acos::mqc_scalar_acos (  
    type(mqc\_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.3 mqc\_algebra::aimag Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_complex\\_imagpart](#) (ScalarIn)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_complex\\_imagpart](#) (A)

### 6.3.1 Member Function/Subroutine Documentation

#### 6.3.1.1 mqc\_scalar\_complex\_imagpart()

```
type(mqc\_scalar) function mqc_algebra::aimag::mqc_scalar_complex_imagpart (  
    type(mqc\_scalar), intent(in) ScalarIn )
```

### 6.3.1.2 mqc\_vector\_complex\_imagpart()

```
type(mqc_vector) function mqc_algebra::aimag::mqc_vector_complex_imagpart (
    class(mqc_vector), intent(in) A )
```

The documentation for this interface was generated from the following file:

- src/mqc\_algebra.F03

## 6.4 mqc\_algebra::asin Interface Reference

### Public Member Functions

- type(mqc\_scalar) function mqc\_scalar\_asin (Scalar)

### 6.4.1 Member Function/Subroutine Documentation

#### 6.4.1.1 mqc\_scalar\_asin()

```
type(mqc_scalar) function mqc_algebra::asin::mqc_scalar_asin (
    type(mqc_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- src/mqc\_algebra.F03

## 6.5 mqc\_algebra::assignment(=) Interface Reference

### Public Member Functions

- subroutine mqc\_input\_integer\_scalar (ScalarOut, ScalarIn)  
***MQC\_Input\_Integer\_Scalar is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar***
- subroutine mqc\_input\_real\_scalar (ScalarOut, ScalarIn)
- subroutine mqc\_input\_complex\_scalar (ScalarOut, ScalarIn)
- subroutine mqc\_output\_mqcscalar\_scalar (ScalarOut, ScalarIn)
- subroutine mqc\_output\_integer\_scalar (ScalarOut, ScalarIn)
- subroutine mqc\_output\_real\_scalar (ScalarOut, ScalarIn)
- subroutine mqc\_output\_complex\_scalar (ScalarOut, ScalarIn)
- subroutine mqc\_set\_vector2vector (VectorOut, VectorIn)
- subroutine mqc\_set\_vector2integerarray (ArrayOut, VectorIn)

- subroutine [mqc\\_set\\_vector2realarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_set\\_vector2complexarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_set\\_array2vector\\_integer](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_array2vector\\_real](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_array2vector\\_complex](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_matrix2matrix](#) (MatrixOut, MatrixIn)
- subroutine [mqc\\_set\\_matrix2integerarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_set\\_matrix2realarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_set\\_matrix2complexarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_set\\_integerarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_set\\_realarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_set\\_complexarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_set\\_array2tensor](#) (TensorOut, ArrayIn)

## 6.5.1 Member Function/Subroutine Documentation

### 6.5.1.1 mqc\_input\_complex\_scalar()

```
subroutine mqc_algebra::assignment(=)::mqc_input_complex_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    complex(kind=real64), intent(in) ScalarIn )
```

### 6.5.1.2 mqc\_input\_integer\_scalar()

```
subroutine mqc_algebra::assignment(=)::mqc_input_integer_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    integer(kind=int64), intent(in) ScalarIn )
```

**MQC\_Input\_Integer\_Scalar** is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar

#### Purpose:

MQC\_Input\_Integer\_Scalar is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar.

#### Parameters

in, out	<i>ScalarOut</i>	ScalarOut is Type(MQC_Scalar) The name of the output variable
in	<i>ScalarIn</i>	ScalarIn is Integer(kind=int64) The value of the input variable

**Author**

L. M. Thompson

**Date**

2016

**6.5.1.3 mqc\_input\_real\_scalar()**

```
subroutine mqc_algebra::assignment(=)::mqc_input_real_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    real(kind=real64), intent(in) ScalarIn )
```

**6.5.1.4 mqc\_output\_complex\_scalar()**

```
subroutine mqc_algebra::assignment(=)::mqc_output_complex_scalar (
    complex(kind=real64), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )
```

**6.5.1.5 mqc\_output\_integer\_scalar()**

```
subroutine mqc_algebra::assignment(=)::mqc_output_integer_scalar (
    integer(kind=int64), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )
```

**6.5.1.6 mqc\_output\_mqcscalar\_scalar()**

```
subroutine mqc_algebra::assignment(=)::mqc_output_mqcscalar_scalar (
    type(mqc_scalar), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )
```

### 6.5.1.7 mqc\_output\_real\_scalar()

```
subroutine mqc_algebra::assignment(=)::mqc_output_real_scalar (
    real(kind=real64), intent(inout) ScalarOut,
    type(mqc_scalar), intent(in) ScalarIn )
```

### 6.5.1.8 mqc\_set\_array2tensor()

```
subroutine mqc_algebra::assignment(=)::mqc_set_array2tensor (
    type(mqc_r4tensor), intent(inout) TensorOut,
    class(*), dimension(:, :, :, :), intent(in) ArrayIn )
```

### 6.5.1.9 mqc\_set\_array2vector\_complex()

```
subroutine mqc_algebra::assignment(=)::mqc_set_array2vector_complex (
    type(mqc_vector), intent(inout) VectorOut,
    complex(kind=real64), dimension(:), intent(in) ArrayIn )
```

### 6.5.1.10 mqc\_set\_array2vector\_integer()

```
subroutine mqc_algebra::assignment(=)::mqc_set_array2vector_integer (
    type(mqc_vector), intent(inout) VectorOut,
    integer(kind=int64), dimension(:), intent(in) ArrayIn )
```

### 6.5.1.11 mqc\_set\_array2vector\_real()

```
subroutine mqc_algebra::assignment(=)::mqc_set_array2vector_real (
    type(mqc_vector), intent(inout) VectorOut,
    real(kind=real64), dimension(:), intent(in) ArrayIn )
```

### 6.5.1.12 mqc\_set\_complexarray2matrix()

```
subroutine mqc_algebra::assignment(=)::mqc_set_complexarray2matrix (
    type(mqc_matrix), intent(inout) MatrixOut,
    complex(kind=real64), dimension(:, :), intent(in) ArrayIn )
```



#### 6.5.1.13 mqc\_set\_integerarray2matrix()

```
subroutine mqc_algebra::assignment(=)::mqc_set_integerarray2matrix (
    type(mqc_matrix), intent(inout) MatrixOut,
    integer(kind=int64), dimension(:,:), intent(in) ArrayIn )
```

#### 6.5.1.14 mqc\_set\_matrix2complexarray()

```
subroutine mqc_algebra::assignment(=)::mqc_set_matrix2complexarray (
    complex(kind=real64), dimension(:,:), intent(inout), allocatable ArrayOut,
    type(mqc_matrix), intent(in) MatrixIn )
```

#### 6.5.1.15 mqc\_set\_matrix2integerarray()

```
subroutine mqc_algebra::assignment(=)::mqc_set_matrix2integerarray (
    integer(kind=int64), dimension(:,:), intent(inout), allocatable ArrayOut,
    type(mqc_matrix), intent(in) MatrixIn )
```

#### 6.5.1.16 mqc\_set\_matrix2matrix()

```
subroutine mqc_algebra::assignment(=)::mqc_set_matrix2matrix (
    class(mqc_matrix), intent(inout) MatrixOut,
    class(mqc_matrix), intent(in) MatrixIn )
```

#### 6.5.1.17 mqc\_set\_matrix2realarray()

```
subroutine mqc_algebra::assignment(=)::mqc_set_matrix2realarray (
    real(kind=real64), dimension(:,:), intent(inout), allocatable ArrayOut,
    type(mqc_matrix), intent(in) MatrixIn )
```

#### 6.5.1.18 mqc\_set\_realarray2matrix()

```
subroutine mqc_algebra::assignment(=)::mqc_set_realarray2matrix (
    type(mqc_matrix), intent(inout) MatrixOut,
    real(kind=real64), dimension(:,:), intent(in) ArrayIn )
```

### 6.5.1.19 mqc\_set\_vector2complexarray()

```
subroutine mqc_algebra::assignment(=)::mqc_set_vector2complexarray (
    complex(kind=real64), dimension(:), intent(inout), allocatable ArrayOut,
    type(mqc_vector), intent(in) VectorIn )
```

### 6.5.1.20 mqc\_set\_vector2integerarray()

```
subroutine mqc_algebra::assignment(=)::mqc_set_vector2integerarray (
    integer(kind=int64), dimension(:), intent(inout), allocatable ArrayOut,
    type(mqc_vector), intent(in) VectorIn )
```

### 6.5.1.21 mqc\_set\_vector2realarray()

```
subroutine mqc_algebra::assignment(=)::mqc_set_vector2realarray (
    real(kind=real64), dimension(:), intent(inout), allocatable ArrayOut,
    type(mqc_vector), intent(in) VectorIn )
```

### 6.5.1.22 mqc\_set\_vector2vector()

```
subroutine mqc_algebra::assignment(=)::mqc_set_vector2vector (
    class(mqc_vector), intent(inout) VectorOut,
    class(mqc_vector), intent(in) VectorIn )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.6 mqc\_est::assignment(=) Interface Reference

### Public Member Functions

- subroutine [mqc\\_integral\\_output\\_array](#) (matrixOut, integralln)
- subroutine [mqc\\_eigenvalues\\_output\\_array](#) (vectorOut, eigenvaluesIn)

### 6.6.1 Member Function/Subroutine Documentation

### 6.6.1.1 mqc\_eigenvalues\_output\_array()

```
subroutine mqc_est::assignment(=)::mqc_eigenvalues_output_array (
    type(mqc_vector), intent(inout) vectorOut,
    class(mqc_scf_eigenvalues), intent(in) eigenvaluesIn )
```

### 6.6.1.2 mqc\_integral\_output\_array()

```
subroutine mqc_est::assignment(=)::mqc_integral_output_array (
    type(mqc_matrix), intent(inout) matrixOut,
    class(mqc_scf_integral), intent(in) integralIn )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.7 mqc\_algebra::atan Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_atan](#) (Scalar)

### 6.7.1 Member Function/Subroutine Documentation

#### 6.7.1.1 mqc\_scalar\_atan()

```
type(mqc_scalar) function mqc_algebra::atan::mqc_scalar_atan (
    type(mqc_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.8 mqc\_algebra::atan2 Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_atan2](#) (Scalar)

## 6.8.1 Member Function/Subroutine Documentation

### 6.8.1.1 `mqc_scalar_atan2()`

```
type(mqc\_scalar) function mqc_algebra::atan2::mqc_scalar_atan2 (  
    type(mqc\_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.9 `mqc_algebra::cmplx` Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_cmplx](#) (*Scalar1*, *Scalar2*)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_cmplx](#) (*Vector1*, *Vector2*)

## 6.9.1 Member Function/Subroutine Documentation

### 6.9.1.1 `mqc_scalar_cmplx()`

```
type(mqc\_scalar) function mqc_algebra::cmplx::mqc_scalar_cmplx (  
    type(mqc\_scalar), intent(in) Scalar1,  
    type(mqc\_scalar), intent(in) Scalar2 )
```

### 6.9.1.2 `mqc_vector_cmplx()`

```
type(mqc\_vector) function mqc_algebra::cmplx::mqc_vector_cmplx (  
    type(mqc\_vector), intent(in) Vector1,  
    type(mqc\_vector), intent(in) Vector2 )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.10 `mqc_algebra::conjg` Interface Reference

### Public Member Functions

- `type(mqc_scalar)` function `mqc_scalar_complex_conjugate` (`ScalarIn`)

#### 6.10.1 Member Function/Subroutine Documentation

##### 6.10.1.1 `mqc_scalar_complex_conjugate()`

```
type(mqc_scalar) function mqc_algebra::conjg::mqc_scalar_complex_conjugate (  
    type(mqc_scalar), intent(in) ScalarIn )
```

The documentation for this interface was generated from the following file:

- `src/mqc_algebra.F03`

## 6.11 `mqc_algebra::contraction` Interface Reference

### Public Member Functions

- `type(mqc_scalar)` function `mqc_matrix_matrix_contraction` (`Matrix1`, `Matrix2`)

#### 6.11.1 Member Function/Subroutine Documentation

##### 6.11.1.1 `mqc_matrix_matrix_contraction()`

```
type(mqc_scalar) function mqc_algebra::contraction::mqc_matrix_matrix_contraction (  
    type(mqc_matrix), intent(in) Matrix1,  
    type(mqc_matrix), intent(in) Matrix2 )
```

The documentation for this interface was generated from the following file:

- `src/mqc_algebra.F03`

## 6.12 mqc\_est::contraction Interface Reference

### Public Member Functions

- type(mqc\_scalar) function [mqc\\_scf\\_integral\\_contraction](#) (integral1, integral2)
- type([mqc\\_scf\\_integral](#)) function [mqc\\_eri\\_integral\\_contraction](#) (eris, integral, label)

### 6.12.1 Member Function/Subroutine Documentation

#### 6.12.1.1 mqc\_eri\_integral\_contraction()

```
type(mqc\_scf\_integral) function mqc_est::contraction::mqc_eri_integral_contraction (
    type(mqc\_twoeris), intent(in) eris,
    type(mqc\_scf\_integral), intent(in) integral,
    character(len=*), optional label )
```

#### 6.12.1.2 mqc\_scf\_integral\_contraction()

```
type(mqc_scalar) function mqc_est::contraction::mqc_scf_integral_contraction (
    type(mqc\_scf\_integral), intent(in) integral1,
    type(mqc\_scf\_integral), intent(in) integral2 )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.13 mqc\_algebra::cos Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_cos](#) (Scalar)

### 6.13.1 Member Function/Subroutine Documentation

### 6.13.1.1 mqc\_scalar\_cos()

```
type(mqc_scalar) function mqc_algebra::cos::mqc_scalar_cos (
    type(mqc_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- src/mqc\_algebra.F03

## 6.14 mqc\_algebra::dagger Interface Reference

### Public Member Functions

- type(mqc\_vector) function mqc\_vector\_conjugate\_transpose (Vector)
- type(mqc\_matrix) function mqc\_matrix\_conjugate\_transpose (Matrix)

### 6.14.1 Member Function/Subroutine Documentation

#### 6.14.1.1 mqc\_matrix\_conjugate\_transpose()

```
type(mqc_matrix) function mqc_algebra::dagger::mqc_matrix_conjugate_transpose (
    class(mqc_matrix), intent(in) Matrix )
```

#### 6.14.1.2 mqc\_vector\_conjugate\_transpose()

```
type(mqc_vector) function mqc_algebra::dagger::mqc_vector_conjugate_transpose (
    class(mqc_vector), intent(in) Vector )
```

The documentation for this interface was generated from the following file:

- src/mqc\_algebra.F03

## 6.15 mqc\_est::dagger Interface Reference

### Public Member Functions

- type(mqc\_scf\_integral) function mqc\_integral\_conjugate\_transpose (integral, label)

## 6.15.1 Member Function/Subroutine Documentation

### 6.15.1.1 `mqc_integral_conjugate_transpose()`

```
type(mqc_scf_integral) function mqc_est::dagger::mqc_integral_conjugate_transpose (
    type(mqc_scf_integral), intent(in) integral,
    character(len=*), intent(in), optional label )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.16 `mqc_algebra::dot_product` Interface Reference

### Public Member Functions

- `type(mqc_scalar) function mqc_vectorvectordotproduct (Vector1, Vector2)`

### 6.16.1 Member Function/Subroutine Documentation

#### 6.16.1.1 `mqc_vectorvectordotproduct()`

```
type(mqc_scalar) function mqc_algebra::dot_product::mqc_vectorvectordotproduct (
    type(mqc_vector), intent(in) Vector1,
    type(mqc_vector), intent(in) Vector2 )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.17 `mqc_est::dot_product` Interface Reference

### Public Member Functions

- `type(mqc_scalar) function mqc_eigenvalue_eigenvalue_dotproduct (eigenvalueA, eigenvalueB)`



## 6.17.1 Member Function/Subroutine Documentation

### 6.17.1.1 mqc\_eigenvalue\_eigenvalue\_dotproduct()

```
type(mqc_scalar) function mqc_est::dot_product::mqc_eigenvalue_eigenvalue_dotproduct (
    type(mqc_scf_eigenvalues), intent(in) eigenvalueA,
    type(mqc_scf_eigenvalues), intent(in) eigenvalueB )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.18 mqc\_algebra::matmul Interface Reference

### Public Member Functions

- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixdotproduct](#) (MA, MB)
- type([mqc\\_vector](#)) function [mqc\\_matrixvectordotproduct](#) (MA, VB)
- type([mqc\\_vector](#)) function [mqc\\_vectormatrixdotproduct](#) (VA, MB)

## 6.18.1 Member Function/Subroutine Documentation

### 6.18.1.1 mqc\_matrixmatrixdotproduct()

```
type(mqc_matrix) function mqc_algebra::matmul::mqc_matrixmatrixdotproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )
```

### 6.18.1.2 mqc\_matrixvectordotproduct()

```
type(mqc_vector) function mqc_algebra::matmul::mqc_matrixvectordotproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_vector), intent(in) VB )
```

### 6.18.1.3 `mqc_vectormatrixdotproduct()`

```
type(mqc_vector) function mqc_algebra::matmul::mqc_vectormatrixdotproduct (
    type(mqc_vector), intent(in) VA,
    type(mqc_matrix), intent(in) MB )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.19 `mqc_est::matmul` Interface Reference

### Public Member Functions

- type([mqc\\_scf\\_integral](#)) function [mqc\\_integral\\_matrix\\_multiply](#) (integralA, matrixB, label)
- type([mqc\\_scf\\_integral](#)) function [mqc\\_matrix\\_integral\\_multiply](#) (matrixA, integralB, label)
- type([mqc\\_scf\\_integral](#)) function [mqc\\_integral\\_integral\\_multiply](#) (integralA, integralB, label)
- type([mqc\\_scf\\_integral](#)) function [mqc\\_integral\\_eigenvalues\\_multiply](#) (integralA, eigenvaluesB, label)
- type([mqc\\_scf\\_integral](#)) function [mqc\\_eigenvalues\\_integral\\_multiply](#) (eigenvaluesA, integralB, label)
- type([mqc\\_scf\\_eigenvalues](#)) function [mqc\\_eigenvalues\\_eigenvalues\\_multiply](#) (eigenvaluesA, eigenvaluesB, label)

### 6.19.1 Member Function/Subroutine Documentation

#### 6.19.1.1 `mqc_eigenvalues_eigenvalues_multiply()`

```
type(mqc_scf_eigenvalues) function mqc_est::matmul::mqc_eigenvalues_eigenvalues_multiply (
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesA,
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesB,
    character(len=*), intent(in), optional label )
```

#### 6.19.1.2 `mqc_eigenvalues_integral_multiply()`

```
type(mqc_scf_integral) function mqc_est::matmul::mqc_eigenvalues_integral_multiply (
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesA,
    type(mqc_scf_integral), intent(in) integralB,
    character(len=*), intent(in), optional label )
```

### 6.19.1.3 mqc\_integral\_eigenvalues\_multiply()

```
type(mqc_scf_integral) function mqc_est::matmul::mqc_integral_eigenvalues_multiply (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesB,
    character(len=*), intent(in), optional label )
```

### 6.19.1.4 mqc\_integral\_integral\_multiply()

```
type(mqc_scf_integral) function mqc_est::matmul::mqc_integral_integral_multiply (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), intent(in) integralB,
    character(len=*), intent(in), optional label )
```

### 6.19.1.5 mqc\_integral\_matrix\_multiply()

```
type(mqc_scf_integral) function mqc_est::matmul::mqc_integral_matrix_multiply (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_matrix), intent(in) matrixB,
    character(len=*), intent(in), optional label )
```

### 6.19.1.6 mqc\_matrix\_integral\_multiply()

```
type(mqc_scf_integral) function mqc_est::matmul::mqc_matrix_integral_multiply (
    type(mqc_matrix), intent(in) matrixA,
    type(mqc_scf_integral), intent(in) integralB,
    character(len=*), intent(in), optional label )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.20 mqc\_algebra::matrix\_symm2sq Interface Reference

### Public Member Functions

- subroutine [matrix\\_symm2sq\\_integer](#) (N, I\_Symm, I\_Sq)
- subroutine [matrix\\_symm2sq\\_real](#) (N, A\_Symm, A\_Sq)
- subroutine [matrix\\_symm2sq\\_complex](#) (N, A\_Symm, A\_Sq)

## 6.20.1 Member Function/Subroutine Documentation

### 6.20.1.1 `matrix_symm2sq_complex()`

```
subroutine mqc_algebra::matrix_symm2sq::matrix_symm2sq_complex (
    integer(kind=int64), intent(in) N,
    complex(kind=real64), dimension(:), intent(in) A_Symm,
    complex(kind=real64), dimension(n,n), intent(out) A_Sq )
```

### 6.20.1.2 `matrix_symm2sq_integer()`

```
subroutine mqc_algebra::matrix_symm2sq::matrix_symm2sq_integer (
    integer(kind=int64), intent(in) N,
    integer(kind=int64), dimension(:), intent(in) I_Symm,
    integer(kind=int64), dimension(n,n), intent(out) I_Sq )
```

### 6.20.1.3 `matrix_symm2sq_real()`

```
subroutine mqc_algebra::matrix_symm2sq::matrix_symm2sq_real (
    integer(kind=int64), intent(in) N,
    real(kind=real64), dimension(:), intent(in) A_Symm,
    real(kind=real64), dimension(n,n), intent(out) A_Sq )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.21 `mqc_algebra::mqc_cast_complex` Interface Reference

### Public Member Functions

- type([mqc\\_vector](#)) function [mqc\\_vector\\_cast\\_complex](#) (VA)
- type([mqc\\_matrix](#)) function [mqc\\_matrix\\_cast\\_complex](#) (MA)

### 6.21.1 Member Function/Subroutine Documentation

### 6.21.1.1 mqc\_matrix\_cast\_complex()

```
type(mqc_matrix) function mqc_algebra::mqc_cast_complex::mqc_matrix_cast_complex (
    type(mqc_matrix), intent(in) MA )
```

### 6.21.1.2 mqc\_vector\_cast\_complex()

```
type(mqc_vector) function mqc_algebra::mqc_cast_complex::mqc_vector_cast_complex (
    type(mqc_vector), intent(in) VA )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.22 mqc\_algebra::mqc\_cast\_real Interface Reference

### Public Member Functions

- type(mqc\_vector) function [mqc\\_vector\\_cast\\_real](#) (VA)
- type(mqc\_matrix) function [mqc\\_matrix\\_cast\\_real](#) (MA)

### 6.22.1 Member Function/Subroutine Documentation

#### 6.22.1.1 mqc\_matrix\_cast\_real()

```
type(mqc_matrix) function mqc_algebra::mqc_cast_real::mqc_matrix_cast_real (
    type(mqc_matrix), intent(in) MA )
```

#### 6.22.1.2 mqc\_vector\_cast\_real()

```
type(mqc_vector) function mqc_algebra::mqc_cast_real::mqc_vector_cast_real (
    type(mqc_vector), intent(in) VA )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.23 mqc\_est::mqc\_determinant Type Reference

### Public Attributes

- type([mqc\\_determinant\\_string](#)) [strings](#)
- character(len=64) [order](#)
- integer(kind=int64) [ndets](#)
- integer(kind=int64) [nalpstr](#)
- integer(kind=int64) [nbetstr](#)

### 6.23.1 Member Data Documentation

#### 6.23.1.1 nalpstr

```
integer(kind=int64) mqc_est::mqc_determinant::nalpstr
```

#### 6.23.1.2 nbetstr

```
integer(kind=int64) mqc_est::mqc_determinant::nbetstr
```

#### 6.23.1.3 ndets

```
integer(kind=int64) mqc_est::mqc_determinant::ndets
```

#### 6.23.1.4 order

```
character(len=64) mqc_est::mqc_determinant::order
```

#### 6.23.1.5 strings

```
type(mqc\_determinant\_string) mqc_est::mqc_determinant::strings
```

The documentation for this type was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.24 mqc\_est::mqc\_determinant\_string Type Reference

### Public Attributes

- type(mqc\_matrix) [alpha](#)
- type(mqc\_matrix) [beta](#)

### 6.24.1 Member Data Documentation

#### 6.24.1.1 alpha

```
type(mqc_matrix) mqc_est::mqc_determinant_string::alpha
```

#### 6.24.1.2 beta

```
type(mqc_matrix) mqc_est::mqc_determinant_string::beta
```

The documentation for this type was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.25 mqc\_algebra::mqc\_have\_complex Interface Reference

### Public Member Functions

- logical function [mqc\\_vector\\_havecomplex](#) (Vector)
- logical function [mqc\\_matrix\\_havecomplex](#) (Matrix)

### 6.25.1 Member Function/Subroutine Documentation

#### 6.25.1.1 mqc\_matrix\_havecomplex()

```
logical function mqc_algebra::mqc_have_complex::mqc_matrix_havecomplex (  
    type(mqc\_matrix), intent(in) Matrix )
```

### 6.25.1.2 `mqc_vector_havecomplex()`

```
logical function mqc_algebra::mqc_have_complex::mqc_vector_havecomplex (
    type(mqc_vector), intent(in) Vector )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.26 `mqc_algebra::mqc_have_int` Interface Reference

### Public Member Functions

- logical function [mqc\\_vector\\_haveinteger](#) (Vector)
- logical function [mqc\\_matrix\\_haveinteger](#) (Matrix)

### 6.26.1 Member Function/Subroutine Documentation

#### 6.26.1.1 `mqc_matrix_haveinteger()`

```
logical function mqc_algebra::mqc_have_int::mqc_matrix_haveinteger (
    type(mqc_matrix), intent(in) Matrix )
```

#### 6.26.1.2 `mqc_vector_haveinteger()`

```
logical function mqc_algebra::mqc_have_int::mqc_vector_haveinteger (
    type(mqc_vector), intent(in) Vector )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.27 `mqc_algebra::mqc_have_real` Interface Reference

### Public Member Functions

- logical function [mqc\\_vector\\_havereal](#) (Vector)
- logical function [mqc\\_matrix\\_havereal](#) (Matrix)



## 6.27.1 Member Function/Subroutine Documentation

### 6.27.1.1 mqc\_matrix\_havereal()

```
logical function mqc_algebra::mqc_have_real::mqc_matrix_havereal (
    type(mqc_matrix), intent(in) Matrix )
```

### 6.27.1.2 mqc\_vector\_havereal()

```
logical function mqc_algebra::mqc_have_real::mqc_vector_havereal (
    type(mqc_vector), intent(in) Vector )
```

The documentation for this interface was generated from the following file:

- src/mqc\_algebra.F03

## 6.28 mqc\_algebra::mqc\_matrix Type Reference

### Public Member Functions

- Procedure, public `print` => `mqc_print_matrix_algebra1`
- Procedure, public `initialize` => `mqc_matrix_initialize`
- Procedure, public `init` => `mqc_matrix_initialize`
- Procedure, public `identity` => `mqc_matrix_identity`
- Procedure, public `set` => `mqc_matrix_set`
- Procedure, public `norm` => `mqc_matrix_norm`
- Procedure, public `transpose` => `mqc_matrix_transpose`
- Procedure, public `dagger` => `mqc_matrix_conjugate_transpose`
- Procedure, public `diag` => `mqc_matrix_diagonalize`
- Procedure, public `svd` => `mqc_matrix_svd`
- Procedure, public `eigensys` => `mqc_matrix_generalized_eigensystem`
- Procedure, public `inv` => `mqc_matrix_inverse`
- Procedure, public `det` => `mqc_matrix_determinant`
- Procedure, public `trace` => `mqc_matrix_trace`
- Procedure, public `rmsmax` => `mqc_matrix_rms_max`
- Procedure, public `sqrt` => `mqc_matrix_sqrt`
- Procedure, public `at` => `mqc_matrix_scalar_at`
- Procedure, public `vat` => `mqc_matrix_vector_at`
- Procedure, public `mat` => `mqc_matrix_matrix_at`
- Procedure, public `put` => `mqc_matrix_scalar_put`
- Procedure, public `vput` => `mqc_matrix_vector_put`
- Procedure, public `mput` => `mqc_matrix_matrix_put`
- Procedure, public `s_type` => `mqc_matrix_storagetype`

## Public Attributes

- [real](#)(kind=real64), dimension(:, :), allocatable [matr](#)
- [integer](#)(kind=int64), dimension(:, :), allocatable [mati](#)
- [complex](#)(kind=real64), dimension(:, :), allocatable [matc](#)

## 6.28.1 Member Function/Subroutine Documentation

### 6.28.1.1 `at()`

Procedure, public `mqc_algebra::mqc_matrix::at ( )`

### 6.28.1.2 `dagger()`

Procedure, public `mqc_algebra::mqc_matrix::dagger ( )`

### 6.28.1.3 `det()`

Procedure, public `mqc_algebra::mqc_matrix::det ( )`

### 6.28.1.4 `diag()`

Procedure, public `mqc_algebra::mqc_matrix::diag ( )`

### 6.28.1.5 `eigensys()`

Procedure, public `mqc_algebra::mqc_matrix::eigensys ( )`

**6.28.1.6 identity()**

Procedure, public mqc\_algebra::mqc\_matrix::identity ( )

**6.28.1.7 init()**

Procedure, public mqc\_algebra::mqc\_matrix::init ( )

**6.28.1.8 initialize()**

Procedure, public mqc\_algebra::mqc\_matrix::initialize ( )

**6.28.1.9 inv()**

Procedure, public mqc\_algebra::mqc\_matrix::inv ( )

**6.28.1.10 mat()**

Procedure, public mqc\_algebra::mqc\_matrix::mat ( )

**6.28.1.11 mput()**

Procedure, public mqc\_algebra::mqc\_matrix::mput ( )

**6.28.1.12 norm()**

Procedure, public mqc\_algebra::mqc\_matrix::norm ( )

**6.28.1.13 print()**

Procedure, public mqc\_algebra::mqc\_matrix::print ( )

**6.28.1.14 put()**

Procedure, public mqc\_algebra::mqc\_matrix::put ( )

**6.28.1.15 rmsmax()**

Procedure, public mqc\_algebra::mqc\_matrix::rmsmax ( )

**6.28.1.16 s\_type()**

Procedure, public mqc\_algebra::mqc\_matrix::s\_type ( )

**6.28.1.17 set()**

Procedure, public mqc\_algebra::mqc\_matrix::set ( )

**6.28.1.18 sqrt()**

Procedure, public mqc\_algebra::mqc\_matrix::sqrt ( )

**6.28.1.19 svd()**

Procedure, public mqc\_algebra::mqc\_matrix::svd ( )

**6.28.1.20 trace()**

```
Procedure, public mqc_algebra::mqc_matrix::trace ( )
```

**6.28.1.21 transpose()**

```
Procedure, public mqc_algebra::mqc_matrix::transpose ( )
```

**6.28.1.22 vat()**

```
Procedure, public mqc_algebra::mqc_matrix::vat ( )
```

**6.28.1.23 vput()**

```
Procedure, public mqc_algebra::mqc_matrix::vput ( )
```

**6.28.2 Member Data Documentation****6.28.2.1 matc**

```
complex(kind=real64), dimension(:,:), allocatable mqc_algebra::mqc_matrix::matc
```

**6.28.2.2 mati**

```
integer(kind=int64), dimension(:,:), allocatable mqc_algebra::mqc_matrix::mati
```

### 6.28.2.3 matr

```
real(kind=real64), dimension(:,:), allocatable mqc_algebra::mqc_matrix::matr
```

The documentation for this type was generated from the following file:

- src/mqc\_algebra.F03

## 6.29 mqc\_algebra::mqc\_matrix\_diagmatrix\_put Interface Reference

### Public Member Functions

- subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_integer](#) (mat, diagMatrixIn)
- subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_real](#) (mat, diagMatrixIn)
- subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_complex](#) (mat, diagMatrixIn)
- subroutine [mqc\\_matrix\\_diagmatrix\\_put\\_vector](#) (diagVectorIn, mat)

### 6.29.1 Member Function/Subroutine Documentation

#### 6.29.1.1 mqc\_matrix\_diagmatrix\_put\_complex()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put::mqc_matrix_diagmatrix_put_complex (
    class(mqc_matrix), intent(inout) mat,
    complex(kind=real64), dimension(:), intent(in) diagMatrixIn )
```

#### 6.29.1.2 mqc\_matrix\_diagmatrix\_put\_integer()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put::mqc_matrix_diagmatrix_put_integer (
    class(mqc_matrix), intent(inout) mat,
    integer(kind=int64), dimension(:), intent(in) diagMatrixIn )
```

#### 6.29.1.3 mqc\_matrix\_diagmatrix\_put\_real()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put::mqc_matrix_diagmatrix_put_real (
    class(mqc_matrix), intent(inout) mat,
    real(kind=real64), dimension(:), intent(in) diagMatrixIn )
```

### 6.29.1.4 mqc\_matrix\_diagmatrix\_put\_vector()

```
subroutine mqc_algebra::mqc_matrix_diagmatrix_put::mqc_matrix_diagmatrix_put_vector (
    class(mqc_vector), intent(in) diagVectorIn,
    class(mqc_matrix), intent(inout) mat )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.30 mqc\_algebra::mqc\_matrix\_symmmatrix\_put Interface Reference

### Public Member Functions

- subroutine [mqc\\_matrix\\_symmmatrix\\_put\\_integer](#) (mat, symmMatrixIn)
- subroutine [mqc\\_matrix\\_symmmatrix\\_put\\_real](#) (mat, symmMatrixIn)
- subroutine [mqc\\_matrix\\_symmmatrix\\_put\\_complex](#) (mat, symmMatrixIn)

### 6.30.1 Member Function/Subroutine Documentation

#### 6.30.1.1 mqc\_matrix\_symmmatrix\_put\_complex()

```
subroutine mqc_algebra::mqc_matrix_symmmatrix_put::mqc_matrix_symmmatrix_put_complex (
    class(mqc_matrix), intent(inout) mat,
    complex(kind=real64), dimension(:), intent(in) symmMatrixIn )
```

#### 6.30.1.2 mqc\_matrix\_symmmatrix\_put\_integer()

```
subroutine mqc_algebra::mqc_matrix_symmmatrix_put::mqc_matrix_symmmatrix_put_integer (
    class(mqc_matrix), intent(inout) mat,
    integer(kind=int64), dimension(:), intent(in) symmMatrixIn )
```

#### 6.30.1.3 mqc\_matrix\_symmmatrix\_put\_real()

```
subroutine mqc_algebra::mqc_matrix_symmmatrix_put::mqc_matrix_symmmatrix_put_real (
    class(mqc_matrix), intent(inout) mat,
    real(kind=real64), dimension(:), intent(in) symmMatrixIn )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.31 mqc\_est::mqc\_matrix\_undospinblockghf Interface Reference

### Public Member Functions

- subroutine [mqc\\_matrix\\_undospinblockghf\\_eigenvalues](#) (eigenvaluesIn, vectorOut)
- subroutine [mqc\\_matrix\\_undospinblockghf\\_integral](#) (integralIn, matrixOut)

### 6.31.1 Member Function/Subroutine Documentation

#### 6.31.1.1 mqc\_matrix\_undospinblockghf\_eigenvalues()

```
subroutine mqc_est::mqc_matrix_undospinblockghf::mqc_matrix_undospinblockghf_eigenvalues (
    type(mqc_scf_eigenvalues), intent(in) eigenvaluesIn,
    type(mqc_vector), intent(out) vectorOut )
```

#### 6.31.1.2 mqc\_matrix\_undospinblockghf\_integral()

```
subroutine mqc_est::mqc_matrix_undospinblockghf::mqc_matrix_undospinblockghf_integral (
    type(mqc_scf_integral), intent(in) integralIn,
    type(mqc_matrix), intent(out) matrixOut )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.32 mqc\_algebra::mqc\_print Interface Reference

### Public Member Functions

- subroutine [mqc\\_print\\_scalar\\_algebra1](#) (Scalar, IOut, Header, Blank\_At\_Top, Blank\_At\_Bottom)
- subroutine [mqc\\_print\\_vector\\_algebra1](#) (Vector, IOut, Header, Verbose, Blank\_At\_Top, Blank\_At\_Bottom)
- subroutine [mqc\\_print\\_matrix\\_algebra1](#) (Matrix, IOut, Header, Blank\_At\_Top, Blank\_At\_Bottom)
- subroutine [mqc\\_print\\_r4tensor\\_algebra1](#) (Tensor, IOut, Header, blank\_at\_top, blank\_at\_bottom)

### 6.32.1 Member Function/Subroutine Documentation



### 6.32.1.1 mqc\_print\_matrix\_algebra1()

```
subroutine mqc_algebra::mqc_print::mqc_print_matrix_algebra1 (
    class(mqc_matrix), intent(in) Matrix,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in) Header,
    logical, intent(in), optional Blank_At_Top,
    logical, intent(in), optional Blank_At_Bottom )
```

### 6.32.1.2 mqc\_print\_r4tensor\_algebra1()

```
subroutine mqc_algebra::mqc_print::mqc_print_r4tensor_algebra1 (
    class(mqc_r4tensor), intent(in) Tensor,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in), optional Header,
    logical, optional blank_at_top,
    logical, optional blank_at_bottom )
```

### 6.32.1.3 mqc\_print\_scalar\_algebra1()

```
subroutine mqc_algebra::mqc_print::mqc_print_scalar_algebra1 (
    class(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in) Header,
    logical, intent(in), optional Blank_At_Top,
    logical, intent(in), optional Blank_At_Bottom )
```

### 6.32.1.4 mqc\_print\_vector\_algebra1()

```
subroutine mqc_algebra::mqc_print::mqc_print_vector_algebra1 (
    class(mqc_vector), intent(in) Vector,
    integer(kind=int64), intent(in) IOut,
    character(len=*), intent(in) Header,
    logical, intent(in), optional Verbose,
    logical, intent(in), optional Blank_At_Top,
    logical, intent(in), optional Blank_At_Bottom )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.33 mqc\_est::mqc\_print Interface Reference

### Public Member Functions

- subroutine [mqc\\_print\\_wavefunction](#) (wavefunction, iOut, label)
- subroutine [mqc\\_print\\_integral](#) (integral, iOut, header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_print\\_eigenvalues](#) (eigenvalues, iOut, header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_print\\_twoeris](#) (twoERIs, iOut, header, blank\_at\_top, blank\_at\_bottom)

### 6.33.1 Member Function/Subroutine Documentation

#### 6.33.1.1 mqc\_print\_eigenvalues()

```
subroutine mqc_est::mqc_print::mqc_print_eigenvalues (
    class(mqc\_scf\_eigenvalues) eigenvalues,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in) header,
    logical, intent(in), optional blank_at_top,
    logical, intent(in), optional blank_at_bottom )
```

#### 6.33.1.2 mqc\_print\_integral()

```
subroutine mqc_est::mqc_print::mqc_print_integral (
    class(mqc\_scf\_integral) integral,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in) header,
    logical, intent(in), optional blank_at_top,
    logical, intent(in), optional blank_at_bottom )
```

#### 6.33.1.3 mqc\_print\_twoeris()

```
subroutine mqc_est::mqc_print::mqc_print_twoeris (
    class(mqc\_twoeris) twoERIs,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in) header,
    logical, intent(in), optional blank_at_top,
    logical, intent(in), optional blank_at_bottom )
```

#### 6.33.1.4 mqc\_print\_wavefunction()

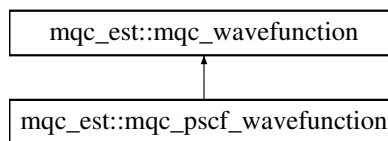
```
subroutine mqc_est::mqc_print::mqc_print_wavefunction (
    class(mqc_wavefunction) wavefunction,
    integer(kind=int64), intent(in) iOut,
    character(len=*), intent(in), optional label )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.34 mqc\_est::mqc\_pscf\_wavefunction Type Reference

Inheritance diagram for mqc\_est::mqc\_pscf\_wavefunction:



### Public Attributes

- integer(kind=int64) [ncore](#)
- integer(kind=int64) [nval](#)
- integer(kind=int64) [nactive](#)
- integer(kind=int64) [nfrz](#)
- type(mqc\_matrix) [pscf\\_amplitudes](#)
- type(mqc\_vector) [pscf\\_energies](#)

### Additional Inherited Members

#### 6.34.1 Member Data Documentation

##### 6.34.1.1 nactive

```
integer(kind=int64) mqc_est::mqc_pscf_wavefunction::nactive
```

#### 6.34.1.2 ncore

```
integer(kind=int64) mqc_est::mqc_pscf_wavefunction::ncore
```

#### 6.34.1.3 nfrz

```
integer(kind=int64) mqc_est::mqc_pscf_wavefunction::nfrz
```

#### 6.34.1.4 nval

```
integer(kind=int64) mqc_est::mqc_pscf_wavefunction::nval
```

#### 6.34.1.5 pscf\_amplitudes

```
type(mqc_matrix) mqc_est::mqc_pscf_wavefunction::pscf_amplitudes
```

#### 6.34.1.6 pscf\_energies

```
type(mqc_vector) mqc_est::mqc_pscf_wavefunction::pscf_energies
```

The documentation for this type was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.35 mqc\_algebra::mqc\_r4tensor Type Reference

### Public Member Functions

- Procedure, public [print](#) => [mqc\\_print\\_r4tensor\\_algebra1](#)
- Procedure, public [at](#) => [mqc\\_r4tensor\\_at](#)
- Procedure, public [put](#) => [mqc\\_r4tensor\\_put](#)
- Procedure, public [initialize](#) => [mqc\\_r4tensor\\_initialize](#)
- Procedure, public [init](#) => [mqc\\_r4tensor\\_initialize](#)

### 6.35.1 Member Function/Subroutine Documentation

#### 6.35.1.1 at()

Procedure, public mqc\_algebra::mqc\_r4tensor::at ( )

#### 6.35.1.2 init()

Procedure, public mqc\_algebra::mqc\_r4tensor::init ( )

#### 6.35.1.3 initialize()

Procedure, public mqc\_algebra::mqc\_r4tensor::initialize ( )

#### 6.35.1.4 print()

Procedure, public mqc\_algebra::mqc\_r4tensor::print ( )

#### 6.35.1.5 put()

Procedure, public mqc\_algebra::mqc\_r4tensor::put ( )

The documentation for this type was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.36 mqc\_algebra::mqc\_scalar Type Reference

### Public Member Functions

- Procedure, public [print](#) => [mqc\\_print\\_scalar\\_algebra1](#)
- Procedure, public [rval](#) => [mqc\\_scalar\\_get\\_intrinsic\\_real](#)
- Procedure, public [ival](#) => [mqc\\_scalar\\_get\\_intrinsic\\_integer](#)
- Procedure, public [cval](#) => [mqc\\_scalar\\_get\\_intrinsic\\_complex](#)
- Procedure, public [abs](#) => [mqc\\_scalar\\_get\\_abs\\_value](#)
- Procedure, public [random](#) => [mqc\\_scalar\\_get\\_random\\_value](#)

### 6.36.1 Member Function/Subroutine Documentation

#### 6.36.1.1 `abs()`

Procedure, public `mqc_algebra::mqc_scalar::abs ( )`

#### 6.36.1.2 `cval()`

Procedure, public `mqc_algebra::mqc_scalar::cval ( )`

#### 6.36.1.3 `ival()`

Procedure, public `mqc_algebra::mqc_scalar::ival ( )`

#### 6.36.1.4 `print()`

Procedure, public `mqc_algebra::mqc_scalar::print ( )`

#### 6.36.1.5 `random()`

Procedure, public `mqc_algebra::mqc_scalar::random ( )`

#### 6.36.1.6 `rval()`

Procedure, public `mqc_algebra::mqc_scalar::rval ( )`

The documentation for this type was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.37 mqc\_est::mqc\_scf\_eigenvalues Type Reference

### Public Member Functions

- Procedure, public [print](#) => [mqc\\_print\\_eigenvalues](#)
- Procedure, public [getlabel](#) => [mqc\\_eigenvalues\\_array\\_name](#)
- Procedure, public [addlabel](#) => [mqc\\_eigenvalues\\_add\\_name](#)
- Procedure, public [getblock](#) => [mqc\\_eigenvalues\\_output\\_block](#)
- Procedure, public [power](#) => [mqc\\_scf\\_eigenvalues\\_power](#)
- Procedure, public [at](#) => [mqc\\_eigenvalues\\_at](#)

### 6.37.1 Member Function/Subroutine Documentation

#### 6.37.1.1 addlabel()

Procedure, public mqc\_est::mqc\_scf\_eigenvalues::addlabel ( )

#### 6.37.1.2 at()

Procedure, public mqc\_est::mqc\_scf\_eigenvalues::at ( )

#### 6.37.1.3 getblock()

Procedure, public mqc\_est::mqc\_scf\_eigenvalues::getblock ( )

#### 6.37.1.4 getlabel()

Procedure, public mqc\_est::mqc\_scf\_eigenvalues::getlabel ( )

#### 6.37.1.5 power()

Procedure, public mqc\_est::mqc\_scf\_eigenvalues::power ( )

### 6.37.1.6 print()

```
Procedure, public mqc_est::mqc_scf_eigenvalues::print ( )
```

The documentation for this type was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.38 mqc\_est::mqc\_scf\_integral Type Reference

### Public Member Functions

- Procedure, public [print](#) => [mqc\\_print\\_integral](#)
- Procedure, public [getlabel](#) => [mqc\\_integral\\_array\\_name](#)
- Procedure, public [addlabel](#) => [mqc\\_integral\\_add\\_name](#)
- Procedure, public [getblock](#) => [mqc\\_integral\\_output\\_block](#)
- Procedure, public [identity](#) => [mqc\\_integral\\_identity](#)
- Procedure, public [init](#) => [mqc\\_integral\\_initialize](#)
- Procedure, public [diag](#) => [mqc\\_scf\\_integral\\_diagonalize](#)
- Procedure, public [eigensys](#) => [mqc\\_scf\\_integral\\_generalized\\_eigensystem](#)
- Procedure, public [inv](#) => [mqc\\_scf\\_integral\\_inverse](#)
- Procedure, public [trace](#) => [mqc\\_scf\\_integral\\_trace](#)
- Procedure, public [det](#) => [mqc\\_scf\\_integral\\_determinant](#)
- Procedure, public [norm](#) => [mqc\\_integral\\_norm](#)
- Procedure, public [setelist](#) => [mqc\\_integral\\_set\\_energy\\_list](#)
- Procedure, public [getelist](#) => [mqc\\_integral\\_get\\_energy\\_list](#)
- Procedure, public [deleteelist](#) => [mqc\\_integral\\_delete\\_energy\\_list](#)
- Procedure, public [orbitals](#) => [mqc\\_integral\\_output\\_orbitals](#)
- Procedure, public [swap](#) => [mqc\\_integral\\_swap\\_orbitals](#)

### 6.38.1 Member Function/Subroutine Documentation

#### 6.38.1.1 addlabel()

```
Procedure, public mqc_est::mqc_scf_integral::addlabel ( )
```

#### 6.38.1.2 deleteelist()

```
Procedure, public mqc_est::mqc_scf_integral::deleteelist ( )
```



**6.38.1.3 det()**

Procedure, public mqc\_est::mqc\_scf\_integral::det ( )

**6.38.1.4 diag()**

Procedure, public mqc\_est::mqc\_scf\_integral::diag ( )

**6.38.1.5 eigensys()**

Procedure, public mqc\_est::mqc\_scf\_integral::eigensys ( )

**6.38.1.6 getblock()**

Procedure, public mqc\_est::mqc\_scf\_integral::getblock ( )

**6.38.1.7 getelist()**

Procedure, public mqc\_est::mqc\_scf\_integral::getelist ( )

**6.38.1.8 getlabel()**

Procedure, public mqc\_est::mqc\_scf\_integral::getlabel ( )

**6.38.1.9 identity()**

Procedure, public mqc\_est::mqc\_scf\_integral::identity ( )

**6.38.1.10 init()**

Procedure, public mqc\_est::mqc\_scf\_integral::init ( )

**6.38.1.11 inv()**

Procedure, public mqc\_est::mqc\_scf\_integral::inv ( )

**6.38.1.12 norm()**

Procedure, public mqc\_est::mqc\_scf\_integral::norm ( )

**6.38.1.13 orbitals()**

Procedure, public mqc\_est::mqc\_scf\_integral::orbitals ( )

**6.38.1.14 print()**

Procedure, public mqc\_est::mqc\_scf\_integral::print ( )

**6.38.1.15 setelist()**

Procedure, public mqc\_est::mqc\_scf\_integral::setelist ( )

**6.38.1.16 swap()**

Procedure, public mqc\_est::mqc\_scf\_integral::swap ( )

### 6.38.1.17 trace()

Procedure, public mqc\_est::mqc\_scf\_integral::trace ( )

The documentation for this type was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.39 mqc\_algebra::mqc\_set\_array2vector Interface Reference

### Public Member Functions

- subroutine [mqc\\_set\\_array2vector\\_integer](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_array2vector\\_real](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_set\\_array2vector\\_complex](#) (VectorOut, ArrayIn)

### 6.39.1 Member Function/Subroutine Documentation

#### 6.39.1.1 mqc\_set\_array2vector\_complex()

```
subroutine mqc_algebra::mqc_set_array2vector::mqc_set_array2vector_complex (
    type(mqc_vector), intent(inout) VectorOut,
    complex(kind=real64), dimension(:), intent(in) ArrayIn )
```

#### 6.39.1.2 mqc\_set\_array2vector\_integer()

```
subroutine mqc_algebra::mqc_set_array2vector::mqc_set_array2vector_integer (
    type(mqc_vector), intent(inout) VectorOut,
    integer(kind=int64), dimension(:), intent(in) ArrayIn )
```

#### 6.39.1.3 mqc\_set\_array2vector\_real()

```
subroutine mqc_algebra::mqc_set_array2vector::mqc_set_array2vector_real (
    type(mqc_vector), intent(inout) VectorOut,
    real(kind=real64), dimension(:), intent(in) ArrayIn )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.40 mqc\_est::mqc\_twoeris Type Reference

### Public Member Functions

- procedure, public [print](#) => [mqc\\_print\\_twoeris](#)

### 6.40.1 Member Function/Subroutine Documentation

#### 6.40.1.1 print()

```
procedure, public mqc_est::mqc_twoeris::print ( )
```

The documentation for this type was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.41 mqc\_algebra::mqc\_vector Type Reference

### Public Member Functions

- Procedure, public [print](#) => [mqc\\_print\\_vector\\_algebra1](#)
- Procedure, public [initialize](#) => [mqc\\_vector\\_initialize](#)
- Procedure, public [size](#) => [mqc\\_length\\_vector](#)
- Procedure, public [init](#) => [mqc\\_vector\\_initialize](#)
- Procedure, public [norm](#) => [mqc\\_vector\\_norm](#)
- Procedure, public [transpose](#) => [mqc\\_vector\\_transpose](#)
- Procedure, public [dagger](#) => [mqc\\_vector\\_conjugate\\_transpose](#)
- Procedure, public [at](#) => [mqc\\_vector\\_scalar\\_at](#)
- Procedure, public [vat](#) => [mqc\\_vector\\_vector\\_at](#)
- Procedure, public [put](#) => [mqc\\_vector\\_scalar\\_put](#)
- Procedure, public [vput](#) => [mqc\\_vector\\_vector\\_put](#)
- Procedure, public [push](#) => [mqc\\_vector\\_push](#)
- Procedure, public [unshift](#) => [mqc\\_vector\\_unshift](#)
- Procedure, public [pop](#) => [mqc\\_vector\\_pop](#)
- Procedure, public [shift](#) => [mqc\\_vector\\_shift](#)
- Procedure, public [maxval](#) => [mqc\\_vector\\_maxval](#)
- Procedure, public [minval](#) => [mqc\\_vector\\_minloc](#)
- Procedure, public [maxloc](#) => [mqc\\_vector\\_maxval](#)
- Procedure, public [minloc](#) => [mqc\\_vector\\_minloc](#)
- Procedure, public [argsort](#) => [mqc\\_vector\\_argsort](#)
- Procedure, public [sort](#) => [mqc\\_vector\\_sort](#)
- Procedure, public [sqrt](#) => [mqc\\_vector\\_sqrt](#)
- Procedure, public [abs](#) => [mqc\\_vector\\_abs](#)
- Procedure, public [power](#) => [mqc\\_vector\\_power](#)
- Procedure, public [diag](#) => [mqc\\_matrix\\_diagmatrix\\_put\\_vector](#)

## Public Attributes

- integer(kind=int64) [length](#) =0
- character(len=64) [data\\_type](#)
- [real](#)(kind=real64), dimension(:), allocatable [vecr](#)
- integer(kind=int64), dimension(:), allocatable [veci](#)
- complex(kind=real64), dimension(:), allocatable [vecc](#)

## 6.41.1 Member Function/Subroutine Documentation

### 6.41.1.1 abs()

Procedure, public mqc\_algebra::mqc\_vector::abs ( )

### 6.41.1.2 argsort()

Procedure, public mqc\_algebra::mqc\_vector::argsort ( )

### 6.41.1.3 at()

Procedure, public mqc\_algebra::mqc\_vector::at ( )

### 6.41.1.4 dagger()

Procedure, public mqc\_algebra::mqc\_vector::dagger ( )

### 6.41.1.5 diag()

Procedure, public mqc\_algebra::mqc\_vector::diag ( )

**6.41.1.6 init()**

Procedure, public mqc\_algebra::mqc\_vector::init ( )

**6.41.1.7 initialize()**

Procedure, public mqc\_algebra::mqc\_vector::initialize ( )

**6.41.1.8 maxloc()**

Procedure, public mqc\_algebra::mqc\_vector::maxloc ( )

**6.41.1.9 maxval()**

Procedure, public mqc\_algebra::mqc\_vector::maxval ( )

**6.41.1.10 minloc()**

Procedure, public mqc\_algebra::mqc\_vector::minloc ( )

**6.41.1.11 minval()**

Procedure, public mqc\_algebra::mqc\_vector::minval ( )

**6.41.1.12 norm()**

Procedure, public mqc\_algebra::mqc\_vector::norm ( )

**6.41.1.13 pop()**

Procedure, public mqc\_algebra::mqc\_vector::pop ( )

**6.41.1.14 power()**

Procedure, public mqc\_algebra::mqc\_vector::power ( )

**6.41.1.15 print()**

Procedure, public mqc\_algebra::mqc\_vector::print ( )

**6.41.1.16 push()**

Procedure, public mqc\_algebra::mqc\_vector::push ( )

**6.41.1.17 put()**

Procedure, public mqc\_algebra::mqc\_vector::put ( )

**6.41.1.18 shift()**

Procedure, public mqc\_algebra::mqc\_vector::shift ( )

**6.41.1.19 size()**

Procedure, public mqc\_algebra::mqc\_vector::size ( )

**6.41.1.20 sort()**

```
Procedure, public mqc_algebra::mqc_vector::sort ( )
```

**6.41.1.21 sqrt()**

```
Procedure, public mqc_algebra::mqc_vector::sqrt ( )
```

**6.41.1.22 transpose()**

```
Procedure, public mqc_algebra::mqc_vector::transpose ( )
```

**6.41.1.23 unshift()**

```
Procedure, public mqc_algebra::mqc_vector::unshift ( )
```

**6.41.1.24 vat()**

```
Procedure, public mqc_algebra::mqc_vector::vat ( )
```

**6.41.1.25 vput()**

```
Procedure, public mqc_algebra::mqc_vector::vput ( )
```

**6.41.2 Member Data Documentation****6.41.2.1 data\_type**

```
character(len=64) mqc_algebra::mqc_vector::data_type
```



### 6.41.2.2 length

```
integer(kind=int64) mqc_algebra::mqc_vector::length =0
```

### 6.41.2.3 vecc

```
complex(kind=real64), dimension(:), allocatable mqc_algebra::mqc_vector::vecc
```

### 6.41.2.4 veci

```
integer(kind=int64), dimension(:), allocatable mqc_algebra::mqc_vector::veci
```

### 6.41.2.5 vecr

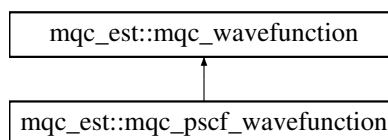
```
real(kind=real64), dimension(:), allocatable mqc_algebra::mqc_vector::vecr
```

The documentation for this type was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.42 mqc\_est::mqc\_wavefunction Type Reference

Inheritance diagram for mqc\_est::mqc\_wavefunction:



### Public Member Functions

- Procedure, public [print](#) => [mqc\\_print\\_wavefunction](#)

## Public Attributes

- type(mqc\_scf\_integral) [mo\\_coefficients](#)
- type(mqc\_scf\_eigenvalues) [mo\\_energies](#)
- type(mqc\_scf\_eigenvalues) [mo\\_symmetries](#)
- type(mqc\_scf\_integral) [core\\_hamiltonian](#)
- type(mqc\_scf\_integral) [fock\\_matrix](#)
- type(mqc\_scf\_integral) [density\\_matrix](#)
- type(mqc\_scf\_integral) [scf\\_density\\_matrix](#)
- type(mqc\_scf\_integral) [overlap\\_matrix](#)
- type(mqc\_scalar) [nalpha](#)
- type(mqc\_scalar) [nbeta](#)
- type(mqc\_scalar) [nelectrons](#)
- type(mqc\_scalar) [nbasis](#)
- type(mqc\_scalar) [charge](#)
- type(mqc\_scalar) [multiplicity](#)
- character(len=256) [basis](#)
- character(len=256) [symmetry](#)
- character(len=256) [wf\\_type](#)
- logical [wf\\_complex](#)

## 6.42.1 Member Function/Subroutine Documentation

### 6.42.1.1 `print()`

Procedure, public mqc\_est::mqc\_wavefunction::print ( )

## 6.42.2 Member Data Documentation

### 6.42.2.1 `basis`

character(len=256) mqc\_est::mqc\_wavefunction::basis

### 6.42.2.2 `charge`

type(mqc\_scalar) mqc\_est::mqc\_wavefunction::charge

### 6.42.2.3 core\_hamiltonian

```
type(mqc_scf_integral) mqc_est::mqc_wavefunction::core_hamiltonian
```

### 6.42.2.4 density\_matrix

```
type(mqc_scf_integral) mqc_est::mqc_wavefunction::density_matrix
```

### 6.42.2.5 fock\_matrix

```
type(mqc_scf_integral) mqc_est::mqc_wavefunction::fock_matrix
```

### 6.42.2.6 mo\_coefficients

```
type(mqc_scf_integral) mqc_est::mqc_wavefunction::mo_coefficients
```

### 6.42.2.7 mo\_energies

```
type(mqc_scf_eigenvalues) mqc_est::mqc_wavefunction::mo_energies
```

### 6.42.2.8 mo\_symmetries

```
type(mqc_scf_eigenvalues) mqc_est::mqc_wavefunction::mo_symmetries
```

### 6.42.2.9 multiplicity

```
type(mqc_scalar) mqc_est::mqc_wavefunction::multiplicity
```

**6.42.2.10 nalpha**

```
type(mqc_scalar) mqc_est::mqc_wavefunction::nalpha
```

**6.42.2.11 nbasis**

```
type(mqc_scalar) mqc_est::mqc_wavefunction::nbasis
```

**6.42.2.12 nbeta**

```
type(mqc_scalar) mqc_est::mqc_wavefunction::nbeta
```

**6.42.2.13 nelectrons**

```
type(mqc_scalar) mqc_est::mqc_wavefunction::nelectrons
```

**6.42.2.14 overlap\_matrix**

```
type(mqc_scf_integral) mqc_est::mqc_wavefunction::overlap_matrix
```

**6.42.2.15 scf\_density\_matrix**

```
type(mqc_scf_integral) mqc_est::mqc_wavefunction::scf_density_matrix
```

**6.42.2.16 symmetry**

```
character(len=256) mqc_est::mqc_wavefunction::symmetry
```

## 6.42.2.17 wf\_complex

```
logical mqc_est::mqc_wavefunction::wf_complex
```

## 6.42.2.18 wf\_type

```
character(len=256) mqc_est::mqc_wavefunction::wf_type
```

The documentation for this type was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.43 mqc\_algebra::operator(\*) Interface Reference

## Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalarmultiply](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalarmultiply](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegermultiply](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalarmultiply](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealmultiply](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalarmultiply](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexmultiply](#) (Scalar, ComplexIn)
- type([mqc\\_vector](#)) function [mqc\\_scalarvectorproduct](#) (Scalar, Vector)
- type([mqc\\_vector](#)) function [mqc\\_vectorscalarproduct](#) (vector, scalar)
- type([mqc\\_matrix](#)) function [mqc\\_scalarmatrixproduct](#) (Scalar, Matrix)
- type([mqc\\_matrix](#)) function [mqc\\_matrixscalarproduct](#) (Matrix, Scalar)
- type([mqc\\_vector](#)) function [mqc\\_realvectorproduct](#) (RealIn, Vector)
- type([mqc\\_vector](#)) function [mqc\\_vectorrealproduct](#) (vector, realIn)
- type([mqc\\_vector](#)) function [mqc\\_integervectorproduct](#) (intIn, Vector)
- type([mqc\\_vector](#)) function [mqc\\_vectorintegerproduct](#) (vector, intIn)
- type([mqc\\_vector](#)) function [mqc\\_complexvectorproduct](#) (CompIn, Vector)
- type([mqc\\_vector](#)) function [mqc\\_vectorcomplexproduct](#) (vector, compIn)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixproduct](#) (MA, MB)

## 6.43.1 Member Function/Subroutine Documentation

#### 6.43.1.1 mqc\_complexscalarmultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_complexscalarmultiply (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 6.43.1.2 mqc\_complexvectorproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_complexvectorproduct (
    complex(kind=real64), intent(in) CompIn,
    type(mqc_vector), intent(in) Vector )
```

#### 6.43.1.3 mqc\_integerscalarmultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_integerscalarmultiply (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 6.43.1.4 mqc\_integervectorproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_integervectorproduct (
    integer(kind=int64), intent(in) intIn,
    type(mqc_vector), intent(in) Vector )
```

#### 6.43.1.5 mqc\_matrixmatrixproduct()

```
type(mqc_matrix) function mqc_algebra::operator(*)::mqc_matrixmatrixproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )
```

#### 6.43.1.6 mqc\_matrixscalarproduct()

```
type(mqc_matrix) function mqc_algebra::operator(*)::mqc_matrixscalarproduct (
    type(mqc_matrix), intent(in) Matrix,
    type(mqc_scalar), intent(in) Scalar )
```

#### 6.43.1.7 mqc\_realscalarmultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_realscalarmultiply (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 6.43.1.8 mqc\_realvectorproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_realvectorproduct (
    real(kind=real64), intent(in) RealIn,
    type(mqc_vector), intent(in) Vector )
```

#### 6.43.1.9 mqc\_scalarcomplexmultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_scalarcomplexmultiply (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

#### 6.43.1.10 mqc\_scalarintegermultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_scalarintegermultiply (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

#### 6.43.1.11 mqc\_scalarmatrixproduct()

```
type(mqc_matrix) function mqc_algebra::operator(*)::mqc_scalarmatrixproduct (
    type(mqc_scalar), intent(in) Scalar,
    type(mqc_matrix), intent(in) Matrix )
```

#### 6.43.1.12 mqc\_scalarmultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_scalarmultiply (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 6.43.1.13 mqc\_scalarrealmultiply()

```
type(mqc_scalar) function mqc_algebra::operator(*)::mqc_scalarrealmultiply (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 6.43.1.14 mqc\_scalarvectorproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_scalarvectorproduct (
    type(mqc_scalar), intent(in) Scalar,
    type(mqc_vector), intent(in) Vector )
```

#### 6.43.1.15 mqc\_vectorcomplexproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_vectorcomplexproduct (
    type(mqc_vector), intent(in) vector,
    complex(kind=real64), intent(in) compIn )
```

#### 6.43.1.16 mqc\_vectorintegerproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_vectorintegerproduct (
    type(mqc_vector), intent(in) vector,
    integer(kind=int64), intent(in) intIn )
```

#### 6.43.1.17 mqc\_vectorrealproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_vectorrealproduct (
    type(mqc_vector), intent(in) vector,
    real(kind=real64), intent(in) realIn )
```

#### 6.43.1.18 mqc\_vectorscalarproduct()

```
type(mqc_vector) function mqc_algebra::operator(*)::mqc_vectorscalarproduct (
    type(mqc_vector), intent(in) vector,
    type(mqc_scalar), intent(in) scalar )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)



## 6.44 `mqc_est::operator(*)` Interface Reference

### Public Member Functions

- `type(mqc_scf_integral)` function `mqc_scalar_integral_multiply` (scalar, integral)
- `type(mqc_scf_integral)` function `mqc_integral_scalar_multiply` (integral, scalar)

### 6.44.1 Member Function/Subroutine Documentation

#### 6.44.1.1 `mqc_integral_scalar_multiply()`

```
type(mqc_scf_integral) function mqc_est::operator(*)::mqc_integral_scalar_multiply (
    type(mqc_scf_integral), intent(in) integral,
    type(mqc_scalar), intent(in) scalar )
```

#### 6.44.1.2 `mqc_scalar_integral_multiply()`

```
type(mqc_scf_integral) function mqc_est::operator(*)::mqc_scalar_integral_multiply (
    type(mqc_scalar), intent(in) scalar,
    type(mqc_scf_integral), intent(in) integral )
```

The documentation for this interface was generated from the following file:

- `src/mqc_est.F03`

## 6.45 `mqc_algebra::operator(**)` Interface Reference

### Public Member Functions

- `type(mqc_scalar)` function `mqc_scalarexponent` (Scalar1, Scalar2)
- `type(mqc_scalar)` function `mqc_scalarintegerexponent` (Scalar, IntIn)
- `type(mqc_scalar)` function `mqc_scalarrealexponent` (Scalar, Realln)
- `type(mqc_scalar)` function `mqc_scalarcomplexexponent` (Scalar, Compln)

### 6.45.1 Member Function/Subroutine Documentation

### 6.45.1.1 `mqc_scalarcomplexexponent()`

```
type(mqc_scalar) function mqc_algebra::operator(**)::mqc_scalarcomplexexponent (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) CompIn )
```

### 6.45.1.2 `mqc_scalarexponent()`

```
type(mqc_scalar) function mqc_algebra::operator(**)::mqc_scalarexponent (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

### 6.45.1.3 `mqc_scalarintegerexponent()`

```
type(mqc_scalar) function mqc_algebra::operator(**)::mqc_scalarintegerexponent (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntIn )
```

### 6.45.1.4 `mqc_scalarrealexponent()`

```
type(mqc_scalar) function mqc_algebra::operator(**)::mqc_scalarrealexponent (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.46 `mqc_est::operator(+)` Interface Reference

### Public Member Functions

- `type(mqc_scf_integral)` function `mqc_integral_sum` (integralA, integralB)

### 6.46.1 Member Function/Subroutine Documentation

6.46.1.1 `mqc_integral_sum()`

```
type(mqc_scf_integral) function mqc_est::operator(+)::mqc_integral_sum (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), intent(in) integralB )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

6.47 `mqc_algebra::operator(+)` Interface Reference

## Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalaradd](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalaradd](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegeradd](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalaradd](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealadd](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalaradd](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexadd](#) (Scalar, ComplexIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorvectorsum](#) (Vector1In, Vector2In)
- type([mqc\\_vector](#)) function [mqc\\_scalarvectorsum](#) (ScalarIn, VectorIn)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixsum](#) (MA, MB)

## 6.47.1 Member Function/Subroutine Documentation

6.47.1.1 `mqc_complexscalaradd()`

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_complexscalaradd (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```

6.47.1.2 `mqc_integerscalaradd()`

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_integerscalaradd (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

### 6.47.1.3 mqc\_matrixmatrixsum()

```
type(mqc_matrix) function mqc_algebra::operator(+)::mqc_matrixmatrixsum (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )
```

### 6.47.1.4 mqc\_realscalaradd()

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_realscalaradd (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )
```

### 6.47.1.5 mqc\_scalaradd()

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_scalaradd (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

### 6.47.1.6 mqc\_scalarcomplexadd()

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_scalarcomplexadd (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

### 6.47.1.7 mqc\_scalarintegeradd()

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_scalarintegeradd (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

### 6.47.1.8 mqc\_scalarrealadd()

```
type(mqc_scalar) function mqc_algebra::operator(+)::mqc_scalarrealadd (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

### 6.47.1.9 mqc\_scalarvectorsum()

```
type(mqc_vector) function mqc_algebra::operator(+)::mqc_scalarvectorsum (
    type(mqc_scalar), intent(in) ScalarIn,
    type(mqc_vector), intent(in) VectorIn )
```

### 6.47.1.10 mqc\_vectorvectorsum()

```
type(mqc_vector) function mqc_algebra::operator(+)::mqc_vectorvectorsum (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.48 mqc\_est::operator(-) Interface Reference

### Public Member Functions

- type([mqc\\_scf\\_integral](#)) function [mqc\\_integral\\_difference](#) (integralA, integralB)

### 6.48.1 Member Function/Subroutine Documentation

#### 6.48.1.1 mqc\_integral\_difference()

```
type(mqc_scf_integral) function mqc_est::operator(-)::mqc_integral_difference (
    type(mqc_scf_integral), intent(in) integralA,
    type(mqc_scf_integral), intent(in) integralB )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_est.F03](#)

## 6.49 mqc\_algebra::operator(-) Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalarsubtract](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalarsubtract](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegersubtract](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalarsubtract](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealsubtract](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalarsubtract](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexsubtract](#) (Scalar, ComplexIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorvectordifference](#) (Vector1In, Vector2In)
- type([mqc\\_vector](#)) function [mqc\\_scalarvectordifference](#) (ScalarIn, VectorIn)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixsubtract](#) (MA, MB)

### 6.49.1 Member Function/Subroutine Documentation

#### 6.49.1.1 mqc\_complexscalarsubtract()

```
type(mqc\_scalar) function mqc_algebra::operator(-)::mqc_complexscalarsubtract (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc\_scalar), intent(in) Scalar )
```

#### 6.49.1.2 mqc\_integerscalarsubtract()

```
type(mqc\_scalar) function mqc_algebra::operator(-)::mqc_integerscalarsubtract (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc\_scalar), intent(in) Scalar )
```

#### 6.49.1.3 mqc\_matrixmatrixsubtract()

```
type(mqc\_matrix) function mqc_algebra::operator(-)::mqc_matrixmatrixsubtract (
    type(mqc\_matrix), intent(in) MA,
    type(mqc\_matrix), intent(in) MB )
```

#### 6.49.1.4 mqc\_realscalarsubtract()

```
type(mqc_scalar) function mqc_algebra::operator(-)::mqc_realscalarsubtract (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 6.49.1.5 mqc\_scalarcomplexsubtract()

```
type(mqc_scalar) function mqc_algebra::operator(-)::mqc_scalarcomplexsubtract (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

#### 6.49.1.6 mqc\_scalarintegersubtract()

```
type(mqc_scalar) function mqc_algebra::operator(-)::mqc_scalarintegersubtract (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

#### 6.49.1.7 mqc\_scalarrealsubtract()

```
type(mqc_scalar) function mqc_algebra::operator(-)::mqc_scalarrealsubtract (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 6.49.1.8 mqc\_scalarsubtract()

```
type(mqc_scalar) function mqc_algebra::operator(-)::mqc_scalarsubtract (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 6.49.1.9 mqc\_scalarvectordifference()

```
type(mqc_vector) function mqc_algebra::operator(-)::mqc_scalarvectordifference (
    type(mqc_scalar), intent(in) ScalarIn,
    type(mqc_vector), intent(in) VectorIn )
```

### 6.49.1.10 `mqc_vectorvectordifference()`

```
type(mqc_vector) function mqc_algebra::operator(-)::mqc_vectorvectordifference (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.50 `mqc_algebra::operator(.dot.)` Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_vectorvectordotproduct](#) (Vector1, Vector2)
- type([mqc\\_vector](#)) function [mqc\\_vectormatrixdotproduct](#) (VA, MB)
- type([mqc\\_vector](#)) function [mqc\\_matrixvectordotproduct](#) (MA, VB)
- type([mqc\\_matrix](#)) function [mqc\\_matrixmatrixdotproduct](#) (MA, MB)

### 6.50.1 Member Function/Subroutine Documentation

#### 6.50.1.1 `mqc_matrixmatrixdotproduct()`

```
type(mqc_matrix) function mqc_algebra::operator(.dot.)::mqc_matrixmatrixdotproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_matrix), intent(in) MB )
```

#### 6.50.1.2 `mqc_matrixvectordotproduct()`

```
type(mqc_vector) function mqc_algebra::operator(.dot.)::mqc_matrixvectordotproduct (
    type(mqc_matrix), intent(in) MA,
    type(mqc_vector), intent(in) VB )
```

#### 6.50.1.3 `mqc_vectormatrixdotproduct()`

```
type(mqc_vector) function mqc_algebra::operator(.dot.)::mqc_vectormatrixdotproduct (
    type(mqc_vector), intent(in) VA,
    type(mqc_matrix), intent(in) MB )
```



#### 6.50.1.4 `mqc_vectorvectordotproduct()`

```
type(mqc_scalar) function mqc_algebra::operator(.dot.)::mqc_vectorvectordotproduct (
    type(mqc_vector), intent(in) Vector1,
    type(mqc_vector), intent(in) Vector2 )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.51 `mqc_algebra::operator(.eq.)` Interface Reference

### Public Member Functions

- logical function [mqc\\_scalareq](#) (Scalar1, Scalar2)

#### 6.51.1 Member Function/Subroutine Documentation

##### 6.51.1.1 `mqc_scalareq()`

```
logical function mqc_algebra::operator(.eq.)::mqc_scalareq (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.52 `mqc_algebra::operator(.ewd.)` Interface Reference

### Public Member Functions

- type([mqc\\_matrix](#)) function [mqc\\_elementmatrixdivide](#) (A, B)

#### 6.52.1 Member Function/Subroutine Documentation

### 6.52.1.1 `mqc_elementmatrixdivide()`

```
type(mqc_matrix) function mqc_algebra::operator(.ewd.)::mqc_elementmatrixdivide (
    type(mqc_matrix), intent(in) A,
    type(mqc_matrix), intent(in) B )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.53 `mqc_algebra::operator(.ewp.)` Interface Reference

### Public Member Functions

- `type(mqc_vector)` function `mqc_elementvectorproduct` (Vector1In, Vector2In)
- `type(mqc_matrix)` function `mqc_elementmatrixproduct` (A, B)

### 6.53.1 Member Function/Subroutine Documentation

#### 6.53.1.1 `mqc_elementmatrixproduct()`

```
type(mqc_matrix) function mqc_algebra::operator(.ewp.)::mqc_elementmatrixproduct (
    type(mqc_matrix), intent(in) A,
    type(mqc_matrix), intent(in) B )
```

#### 6.53.1.2 `mqc_elementvectorproduct()`

```
type(mqc_vector) function mqc_algebra::operator(.ewp.)::mqc_elementvectorproduct (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.54 `mqc_algebra::operator(.ge.)` Interface Reference

### Public Member Functions

- logical function `mqc_scalarge` (Scalar1, Scalar2)

## 6.54.1 Member Function/Subroutine Documentation

### 6.54.1.1 `mqc_scalarge()`

```
logical function mqc_algebra::operator(.ge.)::mqc_scalarge (  
    type(mqc_scalar), intent(in) Scalar1,  
    type(mqc_scalar), intent(in) Scalar2 )
```

The documentation for this interface was generated from the following file:

- `src/mqc_algebra.F03`

## 6.55 `mqc_algebra::operator(.gt.)` Interface Reference

### Public Member Functions

- logical function `mqc_scalargt` (Scalar1, Scalar2)
- logical function `mqc_scalargtinteger` (Scalar, IntIn)
- logical function `mqc_integertscalar` (IntIn, Scalar)
- logical function `mqc_scalargtreal` (Scalar, RealIn)
- logical function `mqc_realgtscalar` (RealIn, Scalar)

## 6.55.1 Member Function/Subroutine Documentation

### 6.55.1.1 `mqc_integertscalar()`

```
logical function mqc_algebra::operator(.gt.)::mqc_integertscalar (  
    integer(kind=int64), intent(in) IntIn,  
    type(mqc_scalar), intent(in) Scalar )
```

### 6.55.1.2 `mqc_realgtscalar()`

```
logical function mqc_algebra::operator(.gt.)::mqc_realgtscalar (  
    real(kind=real64), intent(in) RealIn,  
    type(mqc_scalar), intent(in) Scalar )
```

### 6.55.1.3 `mqc_scalargt()`

```
logical function mqc_algebra::operator(.gt.)::mqc_scalargt (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

### 6.55.1.4 `mqc_scalargtinteger()`

```
logical function mqc_algebra::operator(.gt.)::mqc_scalargtinteger (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntIn )
```

### 6.55.1.5 `mqc_scalargtreal()`

```
logical function mqc_algebra::operator(.gt.)::mqc_scalargtreal (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.56 `mqc_algebra::operator(.le.)` Interface Reference

### Public Member Functions

- logical function [mqc\\_scalarle](#) (Scalar1, Scalar2)
- logical function [mqc\\_scalarlereal](#) (Scalar, RealIn)
- logical function [mqc\\_reallscalar](#) (RealIn, Scalar)
- logical function [mqc\\_scalarleinteger](#) (Scalar, IntIn)
- logical function [mqc\\_integerlescalar](#) (IntIn, Scalar)

### 6.56.1 Member Function/Subroutine Documentation

### 6.56.1.1 mqc\_integerlescalar()

```
logical function mqc_algebra::operator(.le.)::mqc_integerlescalar (
    integer(kind=int64), intent(in) IntIn,
    type(mqc_scalar), intent(in) Scalar )
```

### 6.56.1.2 mqc\_reallescalar()

```
logical function mqc_algebra::operator(.le.)::mqc_reallescalar (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )
```

### 6.56.1.3 mqc\_scalarle()

```
logical function mqc_algebra::operator(.le.)::mqc_scalarle (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

### 6.56.1.4 mqc\_scalarleinteger()

```
logical function mqc_algebra::operator(.le.)::mqc_scalarleinteger (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntIn )
```

### 6.56.1.5 mqc\_scalarlereal()

```
logical function mqc_algebra::operator(.le.)::mqc_scalarlereal (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.57 `mqc_algebra::operator(.lt.)` Interface Reference

### Public Member Functions

- logical function [mqc\\_scalarlt](#) (Scalar1, Scalar2)
- logical function [mqc\\_scalarltreal](#) (Scalar, RealIn)
- logical function [mqc\\_realltscalar](#) (RealIn, Scalar)

### 6.57.1 Member Function/Subroutine Documentation

#### 6.57.1.1 `mqc_realltscalar()`

```
logical function mqc_algebra::operator(.lt.)::mqc_realltscalar (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )
```

#### 6.57.1.2 `mqc_scalarlt()`

```
logical function mqc_algebra::operator(.lt.)::mqc_scalarlt (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

#### 6.57.1.3 `mqc_scalarltreal()`

```
logical function mqc_algebra::operator(.lt.)::mqc_scalarltreal (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.58 `mqc_algebra::operator(.ne.)` Interface Reference

### Public Member Functions

- logical function [mqc\\_scalarne](#) (Scalar1, Scalar2)

## 6.58.1 Member Function/Subroutine Documentation

### 6.58.1.1 `mqc_scalarne()`

```
logical function mqc_algebra::operator(.ne.)::mqc_scalarne (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

The documentation for this interface was generated from the following file:

- `src/mqc_algebra.F03`

## 6.59 `mqc_algebra::operator(.outer.)` Interface Reference

### Public Member Functions

- `type(mqc_matrix)` function `mqc_outer` (VA, VB)

## 6.59.1 Member Function/Subroutine Documentation

### 6.59.1.1 `mqc_outer()`

```
type(mqc_matrix) function mqc_algebra::operator(.outer.)::mqc_outer (
    type(mqc_vector), intent(in) VA,
    type(mqc_vector), intent(in) VB )
```

The documentation for this interface was generated from the following file:

- `src/mqc_algebra.F03`

## 6.60 `mqc_algebra::operator(.x.)` Interface Reference

### Public Member Functions

- `type(mqc_vector)` function `mqc_crossproduct` (Vector1In, Vector2In)

## 6.60.1 Member Function/Subroutine Documentation

### 6.60.1.1 `mqc_crossproduct()`

```
type(mqc_vector) function mqc_algebra::operator(.x.)::mqc_crossproduct (
    type(mqc_vector), intent(in) Vector1In,
    type(mqc_vector), intent(in) Vector2In )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.61 `mqc_algebra::operator(/)` Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalardivide](#) (Scalar1, Scalar2)
- type([mqc\\_scalar](#)) function [mqc\\_integerscalardivide](#) (IntegerIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarintegerdivide](#) (Scalar, IntegerIn)
- type([mqc\\_scalar](#)) function [mqc\\_realscalardivide](#) (RealIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarrealddivide](#) (Scalar, RealIn)
- type([mqc\\_scalar](#)) function [mqc\\_complexscalardivide](#) (ComplexIn, Scalar)
- type([mqc\\_scalar](#)) function [mqc\\_scalarcomplexdivide](#) (Scalar, ComplexIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorscalardivide](#) (vector, scalar)
- type([mqc\\_vector](#)) function [mqc\\_vectorrealddivide](#) (vector, realIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorintegerdivide](#) (vector, intIn)
- type([mqc\\_vector](#)) function [mqc\\_vectorcomplexdivide](#) (vector, compIn)

## 6.61.1 Member Function/Subroutine Documentation

### 6.61.1.1 `mqc_complexscalardivide()`

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_complexscalardivide (
    complex(kind=real64), intent(in) ComplexIn,
    type(mqc_scalar), intent(in) Scalar )
```



### 6.61.1.2 mqc\_integerscalardivide()

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_integerscalardivide (
    integer(kind=int64), intent(in) IntegerIn,
    type(mqc_scalar), intent(in) Scalar )
```

### 6.61.1.3 mqc\_realscalardivide()

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_realscalardivide (
    real(kind=real64), intent(in) RealIn,
    type(mqc_scalar), intent(in) Scalar )
```

### 6.61.1.4 mqc\_scalarcomplexdivide()

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_scalarcomplexdivide (
    type(mqc_scalar), intent(in) Scalar,
    complex(kind=real64), intent(in) ComplexIn )
```

### 6.61.1.5 mqc\_scalardivide()

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_scalardivide (
    type(mqc_scalar), intent(in) Scalar1,
    type(mqc_scalar), intent(in) Scalar2 )
```

### 6.61.1.6 mqc\_scalarintegerdivide()

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_scalarintegerdivide (
    type(mqc_scalar), intent(in) Scalar,
    integer(kind=int64), intent(in) IntegerIn )
```

### 6.61.1.7 mqc\_scalarrealddivide()

```
type(mqc_scalar) function mqc_algebra::operator(/)::mqc_scalarrealddivide (
    type(mqc_scalar), intent(in) Scalar,
    real(kind=real64), intent(in) RealIn )
```

#### 6.61.1.8 mqc\_vectorcomplexdivide()

```
type(mqc_vector) function mqc_algebra::operator(/)::mqc_vectorcomplexdivide (
    type(mqc_vector), intent(in) vector,
    complex(kind=real64), intent(in) compIn )
```

#### 6.61.1.9 mqc\_vectorintegerdivide()

```
type(mqc_vector) function mqc_algebra::operator(/)::mqc_vectorintegerdivide (
    type(mqc_vector), intent(in) vector,
    integer(kind=int64), intent(in) intIn )
```

#### 6.61.1.10 mqc\_vectorrealddivide()

```
type(mqc_vector) function mqc_algebra::operator(/)::mqc_vectorrealddivide (
    type(mqc_vector), intent(in) vector,
    real(kind=real64), intent(in) realIn )
```

#### 6.61.1.11 mqc\_vectorscalarddivide()

```
type(mqc_vector) function mqc_algebra::operator(/)::mqc_vectorscalarddivide (
    type(mqc_vector), intent(in) vector,
    type(mqc_scalar), intent(in) scalar )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.62 mqc\_algebra::real Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_complex\\_realpart](#) (ScalarIn)
- type([mqc\\_vector](#)) function [mqc\\_vector\\_complex\\_realpart](#) (A)

### 6.62.1 Member Function/Subroutine Documentation

### 6.62.1.1 mqc\_scalar\_complex\_realpart()

```
type(mqc_scalar) function mqc_algebra::real::mqc_scalar_complex_realpart (
    type(mqc_scalar), intent(in) ScalarIn )
```

### 6.62.1.2 mqc\_vector\_complex\_realpart()

```
type(mqc_vector) function mqc_algebra::real::mqc_vector_complex_realpart (
    class(mqc_vector), intent(in) A )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.63 mqc\_algebra::sin Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_sin](#) (Scalar)

### 6.63.1 Member Function/Subroutine Documentation

#### 6.63.1.1 mqc\_scalar\_sin()

```
type(mqc_scalar) function mqc_algebra::sin::mqc_scalar_sin (
    type(mqc_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- src/[mqc\\_algebra.F03](#)

## 6.64 mqc\_algebra::sqrt Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_sqrt](#) (Scalar)

## 6.64.1 Member Function/Subroutine Documentation

### 6.64.1.1 `mqc_scalar_sqrt()`

```
type(mqc\_scalar) function mqc_algebra::sqrt::mqc_scalar_sqrt (
    type(mqc\_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.65 `mqc_algebra::tan` Interface Reference

### Public Member Functions

- type([mqc\\_scalar](#)) function [mqc\\_scalar\\_tan](#) (*Scalar*)

## 6.65.1 Member Function/Subroutine Documentation

### 6.65.1.1 `mqc_scalar_tan()`

```
type(mqc\_scalar) function mqc_algebra::tan::mqc_scalar_tan (
    type(mqc\_scalar), intent(in) Scalar )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)

## 6.66 `mqc_est::transpose` Interface Reference

### Public Member Functions

- type([mqc\\_scf\\_integral](#)) function [mqc\\_integral\\_transpose](#) (*integral*, *label*)

## 6.66.1 Member Function/Subroutine Documentation

### 6.66.1.1 `mqc_integral_transpose()`

```
type(mqc_scf_integral) function mqc_est::transpose::mqc_integral_transpose (
    type(mqc_scf_integral), intent(in) integral,
    character(len=*), intent(in), optional label )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_est.F03](#)

## 6.67 `mqc_algebra::transpose` Interface Reference

### Public Member Functions

- type(`mqc_vector`) function [mqc\\_vector\\_transpose](#) (Vector)
- type(`mqc_matrix`) function [mqc\\_matrix\\_transpose](#) (Matrix)

### 6.67.1 Member Function/Subroutine Documentation

#### 6.67.1.1 `mqc_matrix_transpose()`

```
type(mqc_matrix) function mqc_algebra::transpose::mqc_matrix_transpose (
    class(mqc_matrix), intent(in) Matrix )
```

#### 6.67.1.2 `mqc_vector_transpose()`

```
type(mqc_vector) function mqc_algebra::transpose::mqc_vector_transpose (
    class(mqc_vector), intent(in) Vector )
```

The documentation for this interface was generated from the following file:

- [src/mqc\\_algebra.F03](#)



## Chapter 7

# File Documentation

### 7.1 src/mqc\_algebra.F03 File Reference

#### Data Types

- type [mqc\\_algebra::mqc\\_scalar](#)
- type [mqc\\_algebra::mqc\\_vector](#)
- type [mqc\\_algebra::mqc\\_matrix](#)
- type [mqc\\_algebra::mqc\\_r4tensor](#)
- interface [mqc\\_algebra::mqc\\_print](#)
- interface [mqc\\_algebra::contraction](#)
- interface [mqc\\_algebra::conjg](#)
- interface [mqc\\_algebra::mqc\\_have\\_real](#)
- interface [mqc\\_algebra::mqc\\_have\\_int](#)
- interface [mqc\\_algebra::mqc\\_have\\_complex](#)
- interface [mqc\\_algebra::mqc\\_cast\\_real](#)
- interface [mqc\\_algebra::mqc\\_cast\\_complex](#)
- interface [mqc\\_algebra::matmul](#)
- interface [mqc\\_algebra::transpose](#)
- interface [mqc\\_algebra::dagger](#)
- interface [mqc\\_algebra::cmplx](#)
- interface [mqc\\_algebra::sqrt](#)
- interface [mqc\\_algebra::abs](#)
- interface [mqc\\_algebra::real](#)
- interface [mqc\\_algebra::aimag](#)
- interface [mqc\\_algebra::sin](#)
- interface [mqc\\_algebra::cos](#)
- interface [mqc\\_algebra::tan](#)
- interface [mqc\\_algebra::asin](#)
- interface [mqc\\_algebra::acos](#)
- interface [mqc\\_algebra::atan](#)
- interface [mqc\\_algebra::atan2](#)
- interface [mqc\\_algebra::mqc\\_set\\_array2vector](#)
- interface [mqc\\_algebra::mqc\\_matrix\\_symmmatrix\\_put](#)

- interface [mqc\\_algebra::mqc\\_matrix\\_diagmatrix\\_put](#)
- interface [mqc\\_algebra::matrix\\_symm2sq](#)
- interface [mqc\\_algebra::dot\\_product](#)
- interface [mqc\\_algebra::assignment\(=\)](#)
- interface [mqc\\_algebra::operator\(+\)](#)
- interface [mqc\\_algebra::operator\(-\)](#)
- interface [mqc\\_algebra::operator\(\\*\)](#)
- interface [mqc\\_algebra::operator\(/\)](#)
- interface [mqc\\_algebra::operator\(\\*\\*\)](#)
- interface [mqc\\_algebra::operator\(.ne.\)](#)
- interface [mqc\\_algebra::operator\(.eq.\)](#)
- interface [mqc\\_algebra::operator\(.lt.\)](#)
- interface [mqc\\_algebra::operator\(.gt.\)](#)
- interface [mqc\\_algebra::operator\(.le.\)](#)
- interface [mqc\\_algebra::operator\(.ge.\)](#)
- interface [mqc\\_algebra::assignment\(=\)](#)
- interface [mqc\\_algebra::operator\(.dot.\)](#)
- interface [mqc\\_algebra::operator\(\\*\)](#)
- interface [mqc\\_algebra::operator\(/\)](#)
- interface [mqc\\_algebra::operator\(+\)](#)
- interface [mqc\\_algebra::operator\(-\)](#)
- interface [mqc\\_algebra::operator\(.ewp.\)](#)
- interface [mqc\\_algebra::operator\(.ewd.\)](#)
- interface [mqc\\_algebra::operator\(.x.\)](#)
- interface [mqc\\_algebra::operator\(.outer.\)](#)
- interface [mqc\\_algebra::assignment\(=\)](#)
- interface [mqc\\_algebra::operator\(+\)](#)
- interface [mqc\\_algebra::operator\(-\)](#)
- interface [mqc\\_algebra::operator\(\\*\)](#)
- interface [mqc\\_algebra::operator\(.dot.\)](#)
- interface [mqc\\_algebra::assignment\(=\)](#)

## Modules

- module [mqc\\_algebra](#)

## Functions/Subroutines

- integer(kind=int64) function [mqc\\_algebra::factorial](#) (n)
- integer(kind=int64) function [mqc\\_algebra::bin\\_coeff](#) (N, K)
- subroutine [mqc\\_algebra::mqc\\_allocate\\_scalar](#) (Scalar, Data\_type)  
***MQC\_Allocate\_Scalar is used to allocate a scalar type variable of the MQC\_Scalar class***
- subroutine [mqc\\_algebra::mqc\\_deallocate\\_scalar](#) (Scalar)
- logical function [mqc\\_algebra::mqc\\_scalar\\_isallocated](#) (Scalar)
- subroutine [mqc\\_algebra::mqc\\_input\\_integer\\_scalar](#) (ScalarOut, ScalarIn)  
***MQC\_Input\_Integer\_Scalar is a subroutine is used to set an intrinsic scalar to an MQC\_Scalar***
- subroutine [mqc\\_algebra::mqc\\_input\\_real\\_scalar](#) (ScalarOut, ScalarIn)
- subroutine [mqc\\_algebra::mqc\\_input\\_complex\\_scalar](#) (ScalarOut, ScalarIn)



- subroutine [mqc\\_algebra::mqc\\_output\\_mqcscalar\\_scalar](#) (ScalarOut, ScalarIn)
- subroutine [mqc\\_algebra::mqc\\_output\\_integer\\_scalar](#) (ScalarOut, ScalarIn)
- subroutine [mqc\\_algebra::mqc\\_output\\_real\\_scalar](#) (ScalarOut, ScalarIn)
- subroutine [mqc\\_algebra::mqc\\_output\\_complex\\_scalar](#) (ScalarOut, ScalarIn)
- subroutine [mqc\\_algebra::mqc\\_print\\_scalar\\_algebra1](#) (Scalar, IOut, Header, Blank\_At\_Top, Blank\_At\_Bottom)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_cmplx](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_sqrt](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_sin](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_cos](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_tan](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_asin](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_acos](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_atan](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_atan2](#) (Scalar)
- logical function [mqc\\_algebra::mqc\\_scalar\\_havereal](#) (Scalar)
- logical function [mqc\\_algebra::mqc\\_scalar\\_haveinteger](#) (Scalar)
- logical function [mqc\\_algebra::mqc\\_scalar\\_havecomplex](#) (Scalar)
- real(kind=real64) function [mqc\\_algebra::mqc\\_scalar\\_get\\_intrinsic\\_real](#) (Scalar)
- integer(kind=int64) function [mqc\\_algebra::mqc\\_scalar\\_get\\_intrinsic\\_integer](#) (Scalar)
- complex(kind=real64) function [mqc\\_algebra::mqc\\_scalar\\_get\\_intrinsic\\_complex](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_get\\_abs\\_value](#) (Scalar)
- subroutine [mqc\\_algebra::mqc\\_scalar\\_get\\_random\\_value](#) (Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalaradd](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarsubtract](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarmultiply](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalardivide](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarexponent](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarintegerexponent](#) (Scalar, IntIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarrealexponent](#) (Scalar, RealIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarcomplexexponent](#) (Scalar, ComplIn)
- logical function [mqc\\_algebra::mqc\\_scalarne](#) (Scalar1, Scalar2)
- logical function [mqc\\_algebra::mqc\\_scalareq](#) (Scalar1, Scalar2)
- logical function [mqc\\_algebra::mqc\\_scalarlt](#) (Scalar1, Scalar2)
- logical function [mqc\\_algebra::mqc\\_realltscalar](#) (RealIn, Scalar)
- logical function [mqc\\_algebra::mqc\\_scalarltreal](#) (Scalar, RealIn)
- logical function [mqc\\_algebra::mqc\\_scalargt](#) (Scalar1, Scalar2)
- logical function [mqc\\_algebra::mqc\\_integertgtscalar](#) (IntIn, Scalar)
- logical function [mqc\\_algebra::mqc\\_scalargtgtinteger](#) (Scalar, IntIn)
- logical function [mqc\\_algebra::mqc\\_realgtscalar](#) (RealIn, Scalar)
- logical function [mqc\\_algebra::mqc\\_scalargtgtreal](#) (Scalar, RealIn)
- logical function [mqc\\_algebra::mqc\\_scalarle](#) (Scalar1, Scalar2)
- logical function [mqc\\_algebra::mqc\\_realleltscalar](#) (RealIn, Scalar)
- logical function [mqc\\_algebra::mqc\\_scalarlereal](#) (Scalar, RealIn)
- logical function [mqc\\_algebra::mqc\\_integerltscalar](#) (IntIn, Scalar)
- logical function [mqc\\_algebra::mqc\\_scalarleinteger](#) (Scalar, IntIn)
- logical function [mqc\\_algebra::mqc\\_scalarge](#) (Scalar1, Scalar2)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_complex\\_conjugate](#) (ScalarIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_complex\\_realpart](#) (ScalarIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalar\\_complex\\_imagpart](#) (ScalarIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_integerscalarmultiply](#) (IntegerIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarintegermultiply](#) (Scalar, IntegerIn)

- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_realscalarmultiply](#) (RealIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarrealmultiply](#) (Scalar, RealIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_complexscalarmultiply](#) (ComplexIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarcomplexmultiply](#) (Scalar, ComplexIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_integerscalardivide](#) (IntegerIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarintegerdivide](#) (Scalar, IntegerIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_realscalardivide](#) (RealIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarrealdivide](#) (Scalar, RealIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_complexscalardivide](#) (ComplexIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarcomplexdivide](#) (Scalar, ComplexIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_integerscalaradd](#) (IntegerIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarintegeradd](#) (Scalar, IntegerIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_realscalaradd](#) (RealIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarrealadd](#) (Scalar, RealIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_complexscalaradd](#) (ComplexIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarcomplexadd](#) (Scalar, ComplexIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_integerscalarsubtract](#) (IntegerIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarintegersubtract](#) (Scalar, IntegerIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_realscalarsubtract](#) (RealIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarrealsubtract](#) (Scalar, RealIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_complexscalarsubtract](#) (ComplexIn, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_scalarcomplexsubtract](#) (Scalar, ComplexIn)
- subroutine [mqc\\_algebra::mqc\\_allocate\\_vector](#) (N, Vector, Data\_Type)
- subroutine [mqc\\_algebra::mqc\\_deallocate\\_vector](#) (Vector)
- integer(kind=int64) function [mqc\\_algebra::mqc\\_length\\_vector](#) (Vector)
- logical function [mqc\\_algebra::mqc\\_vector\\_havereal](#) (Vector)
- logical function [mqc\\_algebra::mqc\\_vector\\_haveinteger](#) (Vector)
- logical function [mqc\\_algebra::mqc\\_vector\\_havecomplex](#) (Vector)
- logical function [mqc\\_algebra::mqc\\_vector\\_iscolumn](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_copy\\_int2real](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_copy\\_int2complex](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_copy\\_real2int](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_copy\\_real2complex](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_copy\\_complex2int](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_copy\\_complex2real](#) (Vector)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vector\\_scalar\\_at](#) (Vec, I)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_vector\\_at](#) (Vec, I, J)
- subroutine [mqc\\_algebra::mqc\\_set\\_vector2integerarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_vector2realarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_vector2complexarray](#) (ArrayOut, VectorIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_array2vector\\_integer](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_array2vector\\_real](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_array2vector\\_complex](#) (VectorOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_vector2vector](#) (VectorOut, VectorIn)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorvectorsum](#) (Vector1In, Vector2In)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorvectordifference](#) (Vector1In, Vector2In)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_scalarvectorsum](#) (ScalarIn, VectorIn)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_scalarvectordifference](#) (ScalarIn, VectorIn)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_elementvectorproduct](#) (Vector1In, Vector2In)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_transpose](#) (Vector)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_conjugate\\_transpose](#) (Vector)

- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vectorvectordotproduct](#) (Vector1, Vector2)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_outer](#) (VA, VB)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_crossproduct](#) (Vector1In, Vector2In)
- subroutine [mqc\\_algebra::mqc\\_print\\_vector\\_algebra1](#) (Vector, IOut, Header, Verbose, Blank\_At\_Top, Blank\_At←\_Bottom)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_cast\\_real](#) (VA)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_cast\\_complex](#) (VA)
- subroutine [mqc\\_algebra::mqc\\_vector\\_scalar\\_put](#) (Vector, Scalar, I)
- subroutine [mqc\\_algebra::mqc\\_vector\\_scalar\\_increment](#) (Vector, Scalar, I)
- subroutine [mqc\\_algebra::mqc\\_vector\\_vector\\_put](#) (Vector, VectorIn, I)
- subroutine [mqc\\_algebra::mqc\\_vector\\_initialize](#) (Vector, Length, Scalar)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_scalarvectorproduct](#) (Scalar, Vector)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorscalarproduct](#) (vector, scalar)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorscalardivide](#) (vector, scalar)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_realvectorproduct](#) (Realln, Vector)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorrealproduct](#) (vector, realln)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorrealddivide](#) (vector, realln)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_integervectorproduct](#) (intln, Vector)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorintegerproduct](#) (vector, intln)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorintegerdivide](#) (vector, intln)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_complexvectorproduct](#) (Compln, Vector)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorcomplexproduct](#) (vector, compln)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectorcomplexdivide](#) (vector, compln)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vector\\_norm](#) (vector, methodIn)
- logical function [mqc\\_algebra::mqc\\_vector\\_isallocated](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_push](#) (Vector, Scalar)
- subroutine [mqc\\_algebra::mqc\\_vector\\_unshift](#) (Vector, Scalar)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vector\\_pop](#) (Vector)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vector\\_shift](#) (Vector)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vector\\_maxval](#) (Vector)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_vector\\_minval](#) (Vector)
- integer function [mqc\\_algebra::mqc\\_vector\\_maxloc](#) (Vector)
- integer function [mqc\\_algebra::mqc\\_vector\\_minloc](#) (Vector)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_argsort](#) (Vector)
- subroutine [mqc\\_algebra::mqc\\_vector\\_sort](#) (Vector, idx)
- subroutine [mqc\\_algebra::mqc\\_vector\\_sqrt](#) (A)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_abs](#) (A)
- subroutine [mqc\\_algebra::mqc\\_vector\\_power](#) (A, P)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_complex\\_realpart](#) (A)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_complex\\_imagpart](#) (A)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vector\\_cmplx](#) (Vector1, Vector2)
- character(len=64) function [mqc\\_algebra::mqc\\_matrix\\_storage\\_type](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_diagonalize](#) (A, EVals, EVecs)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_cast\\_real](#) (MA)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_cast\\_complex](#) (MA)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_matrix\\_scalar\\_at](#) (Mat, I, J)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_matrix\\_vector\\_at](#) (Mat, Rows, Cols)
- recursive subroutine [mqc\\_algebra::mqc\\_matrix\\_vector\\_put](#) (Mat, VectorIn, Rows, Cols)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_matrix\\_at](#) (Mat, Rows, Cols)

***MQC\_Matrix\_Matrix\_At is a function that returns a submatrix of the matrix***

- subroutine [mqc\\_algebra::mqc\\_matrix\\_diagmatrix\\_put\\_vector](#) (diagVectorIn, mat)

- subroutine [mqc\\_algebra::mqc\\_matrix\\_diagmatrix\\_put\\_integer](#) (mat, diagMatrixIn)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_diagmatrix\\_put\\_real](#) (mat, diagMatrixIn)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_diagmatrix\\_put\\_complex](#) (mat, diagMatrixIn)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symmmatrix\\_put\\_integer](#) (mat, symmMatrixIn)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symmmatrix\\_put\\_real](#) (mat, symmMatrixIn)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symmmatrix\\_put\\_complex](#) (mat, symmMatrixIn)
- recursive subroutine [mqc\\_algebra::mqc\\_matrix\\_matrix\\_put](#) (Mat, MatrixIn, Rows, Cols)
- integer(kind=int64) function [mqc\\_algebra::symindexhash](#) (i, j, k, l)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_elementmatrixproduct](#) (A, B)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_elementmatrixdivide](#) (A, B)
- logical function [mqc\\_algebra::mqc\\_matrix\\_test\\_symmetric](#) (Matrix, Option)
- logical function [mqc\\_algebra::mqc\\_matrix\\_test\\_diagonal](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_allocate\\_matrix](#) (M, N, Matrix, Data\_Type, Storage)
- subroutine [mqc\\_algebra::mqc\\_deallocate\\_matrix](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_isallocated](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_set\\_integerarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_realarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_complexarray2matrix](#) (MatrixOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_matrix2integerarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_matrix2realarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_matrix2complexarray](#) (ArrayOut, MatrixIn)
- subroutine [mqc\\_algebra::mqc\\_set\\_matrix2matrix](#) (MatrixOut, MatrixIn)
- subroutine [mqc\\_algebra::mqc\\_print\\_matrix\\_algebra1](#) (Matrix, IOut, Header, Blank\_At\_Top, Blank\_At\_Bottom)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_copy\\_int2real](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_copy\\_int2complex](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_copy\\_real2int](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_copy\\_real2complex](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_copy\\_complex2int](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_copy\\_complex2real](#) (Matrix)
- integer(kind=int64) function [mqc\\_algebra::mqc\\_matrix\\_rows](#) (Matrix)
- integer(kind=int64) function [mqc\\_algebra::mqc\\_matrix\\_columns](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_havereal](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_haveinteger](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_havecomplex](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_havefull](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_havesymmetric](#) (Matrix)
- logical function [mqc\\_algebra::mqc\\_matrix\\_havediagonal](#) (Matrix)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_transpose](#) (Matrix)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_conjugate\\_transpose](#) (Matrix)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_symmetrize](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_full2symm](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symm2full](#) (Matrix, Option)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_full2diag](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_diag2full](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symm2diag](#) (Matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_diag2symm](#) (Matrix)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_symm2full\\_func](#) (Matrix)
- subroutine [mqc\\_algebra::matrix\\_symm2sq\\_integer](#) (N, I\_Symm, I\_Sq)
- subroutine [mqc\\_algebra::matrix\\_symm2sq\\_real](#) (N, A\_Symm, A\_Sq)
- subroutine [mqc\\_algebra::matrix\\_symm2sq\\_complex](#) (N, A\_Symm, A\_Sq)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_vector2diagmatrix](#) (vector)

- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrixmatrixsum](#) (MA, MB)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrixmatrixsubtract](#) (MA, MB)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrixmatrixproduct](#) (MA, MB)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrixmatrixdotproduct](#) (MA, MB)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_matrixvectordotproduct](#) (MA, VB)
- type(mqc\_vector) function [mqc\\_algebra::mqc\\_vectormatrixdotproduct](#) (VA, MB)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrixscalarproduct](#) (Matrix, Scalar)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_scalarmatrixproduct](#) (Scalar, Matrix)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_matrix\\_matrix\\_contraction](#) (Matrix1, Matrix2)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_scalar\\_put](#) (Matrix, Scalar, I, J)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_initialize](#) (Matrix, Rows, Columns, Scalar, Storage)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_identity](#) (matrix, n, m)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_set](#) (matrix, scalar, storage)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_matrix\\_norm](#) (matrix, methodIn)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_matrix\\_determinant](#) (a)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_matrix\\_inverse](#) (a)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_matrix\\_trace](#) (matrix)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_generalized\\_eigensystem](#) (a, bIn, eigenvals, reigenvecs, leigenvecs)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_svd](#) (A, EVals, EUVecs, EVVecs)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_rms\\_max](#) (A, rms\_A, max\_A)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_sqrt](#) (A, eVals, eVecs)
- subroutine [mqc\\_algebra::mqc\\_allocate\\_r4tensor](#) (I, J, K, L, Tensor, Data\_Type, Storage)
- subroutine [mqc\\_algebra::mqc\\_deallocate\\_r4tensor](#) (Tensor)
- type(mqc\_scalar) function [mqc\\_algebra::mqc\\_r4tensor\\_at](#) (Tensor, I, J, K, L)
- subroutine [mqc\\_algebra::mqc\\_r4tensor\\_put](#) (Tensor, Element, I, J, K, L)
- subroutine [mqc\\_algebra::mqc\\_print\\_r4tensor\\_algebra1](#) (Tensor, IOut, Header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_algebra::mqc\\_set\\_array2tensor](#) (TensorOut, ArrayIn)
- subroutine [mqc\\_algebra::mqc\\_r4tensor\\_initialize](#) (R4Tensor, I, J, K, L, Scalar)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symmsymmr4tensor\\_put\\_real](#) (r4Tensor, symmSymmMatrixIn)
- subroutine [mqc\\_algebra::mqc\\_matrix\\_symmsymmr4tensor\\_put\\_complex](#) (r4Tensor, symmSymmMatrixIn)
- logical function [mqc\\_algebra::mqc\\_r4tensor\\_haveinteger](#) (R4Tensor)
- logical function [mqc\\_algebra::mqc\\_r4tensor\\_havereal](#) (R4Tensor)
- logical function [mqc\\_algebra::mqc\\_r4tensor\\_havecomplex](#) (R4Tensor)
- type(mqc\_matrix) function [mqc\\_algebra::mqc\\_givens\\_matrix](#) (m\_size, angle, p, q)

## 7.2 src/mqc\_est.F03 File Reference

### Data Types

- type [mqc\\_est::mqc\\_scf\\_integral](#)
- type [mqc\\_est::mqc\\_scf\\_eigenvalues](#)
- type [mqc\\_est::mqc\\_wavefunction](#)
- type [mqc\\_est::mqc\\_pscf\\_wavefunction](#)
- type [mqc\\_est::mqc\\_determinant\\_string](#)
- type [mqc\\_est::mqc\\_determinant](#)
- type [mqc\\_est::mqc\\_twoeris](#)
- interface [mqc\\_est::mqc\\_print](#)
- interface [mqc\\_est::matmul](#)

- interface [mqc\\_est::dot\\_product](#)
- interface [mqc\\_est::transpose](#)
- interface [mqc\\_est::dagger](#)
- interface [mqc\\_est::contraction](#)
- interface [mqc\\_est::mqc\\_matrix\\_undospinblockghf](#)
- interface [mqc\\_est::assignment\(=\)](#)
- interface [mqc\\_est::operator\(+\)](#)
- interface [mqc\\_est::operator\(-\)](#)
- interface [mqc\\_est::operator\(\\*\)](#)

## Modules

- module [mqc\\_est](#)

## Functions/Subroutines

- subroutine [mqc\\_est::mqc\\_print\\_wavefunction](#) (wavefunction, iOut, label)
- subroutine [mqc\\_est::mqc\\_print\\_integral](#) (integral, iOut, header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_est::mqc\\_print\\_eigenvalues](#) (eigenvalues, iOut, header, blank\_at\_top, blank\_at\_bottom)
- subroutine [mqc\\_est::mqc\\_print\\_twoeris](#) (twoERIs, iOut, header, blank\_at\_top, blank\_at\_bottom)
- logical function [mqc\\_est::mqc\\_integral\\_isallocated](#) (Integral)
- logical function [mqc\\_est::mqc\\_eigenvalues\\_isallocated](#) (Eigenvalues)
- logical function [mqc\\_est::mqc\\_integral\\_has\\_alpha](#) (integral)
- logical function [mqc\\_est::mqc\\_integral\\_has\\_beta](#) (integral)
- logical function [mqc\\_est::mqc\\_integral\\_has\\_alphabeta](#) (integral)
- logical function [mqc\\_est::mqc\\_integral\\_has\\_betaalpha](#) (integral)
- logical function [mqc\\_est::mqc\\_eigenvalues\\_has\\_alpha](#) (eigenvalues)
- logical function [mqc\\_est::mqc\\_eigenvalues\\_has\\_beta](#) (eigenvalues)
- character(len=64) function [mqc\\_est::mqc\\_integral\\_array\\_type](#) (integral)
- character(len=64) function [mqc\\_est::mqc\\_eigenvalues\\_array\\_type](#) (eigenvalues)
- character(len=64) function [mqc\\_est::mqc\\_integral\\_array\\_name](#) (integral)
- character(len=64) function [mqc\\_est::mqc\\_eigenvalues\\_array\\_name](#) (eigenvalues)
- subroutine [mqc\\_est::mqc\\_integral\\_add\\_name](#) (integral, arrayName)
- subroutine [mqc\\_est::mqc\\_eigenvalues\\_add\\_name](#) (eigenvalues, arrayName)
- integer(kind=int64) function [mqc\\_est::mqc\\_integral\\_dimension](#) (integral, label, axis)
- integer(kind=int64) function [mqc\\_est::mqc\\_eigenvalues\\_dimension](#) (eigenvalues, label)
- subroutine [mqc\\_est::mqc\\_twoeris\\_allocate](#) (twoERIs, storageType, integralType, alpha, beta, alphaBeta, beta↔Alpha)
- subroutine [mqc\\_est::mqc\\_integral\\_allocate](#) (integral, arrayName, arrayType, alpha, beta, alphaBeta, betaAlpha)
- subroutine [mqc\\_est::mqc\\_eigenvalues\\_allocate](#) (eigenvalues, arrayName, arrayType, alpha, beta)
- subroutine [mqc\\_est::mqc\\_integral\\_identity](#) (integral, nAlpha, nBeta, label, nAlpha2, nBeta2)
- subroutine [mqc\\_est::mqc\\_integral\\_initialize](#) (integral, nAlpha, nBeta, scalar, label, nAlpha2, nBeta2)
- type(mqc\_matrix) function [mqc\\_est::mqc\\_integral\\_output\\_block](#) (integral, blockName)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_output\\_orbitals](#) (integral, orbString, alphaOrbsIn, beta↔OrbsIn, axis)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_swap\\_orbitals](#) (integral, alphaOrbsIn, betaOrbsIn, axis)
- type(mqc\_vector) function [mqc\\_est::mqc\\_eigenvalues\\_output\\_block](#) (eigenvalues, blockName)
- subroutine [mqc\\_est::mqc\\_integral\\_output\\_array](#) (matrixOut, integralln)
- subroutine [mqc\\_est::mqc\\_eigenvalues\\_output\\_array](#) (vectorOut, eigenvaluesIn)



- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_matrix\\_multiply](#) (integralA, matrixB, label)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_matrix\\_integral\\_multiply](#) (matrixA, integralB, label)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_sum](#) (integralA, integralB)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_difference](#) (integralA, integralB)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_integral\\_multiply](#) (integralA, integralB, label)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_scalar\\_integral\\_multiply](#) (scalar, integral)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_scalar\\_multiply](#) (integral, scalar)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_eigenvalues\\_multiply](#) (integralA, eigenvaluesB, label)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_eigenvalues\\_integral\\_multiply](#) (eigenvaluesA, integralB, label)
- type(mqc\_scf\_eigenvalues) function [mqc\\_est::mqc\\_eigenvalues\\_eigenvalues\\_multiply](#) (eigenvaluesA, eigenvaluesB, label)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_eigenvalue\\_eigenvalue\\_dotproduct](#) (eigenvalueA, eigenvalueB)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_transpose](#) (integral, label)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_integral\\_conjugate\\_transpose](#) (integral, label)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_integral\\_norm](#) (integral, methodIn)
- subroutine [mqc\\_est::mqc\\_matrix\\_spinblockghf](#) (array, nelec, multi, elist)
- subroutine [mqc\\_est::mqc\\_matrix\\_undospinblockghf\\_eigenvalues](#) (eigenvaluesIn, vectorOut)
- subroutine [mqc\\_est::mqc\\_matrix\\_undospinblockghf\\_integral](#) (integralIn, matrixOut)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_scf\\_integral\\_contraction](#) (integral1, integral2)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_eri\\_integral\\_contraction](#) (eris, integral, label)
- subroutine [mqc\\_est::mqc\\_scf\\_integral\\_generalized\\_eigensystem](#) (integralA, integralB, eVals, rEVecs, lEVecs)
- subroutine [mqc\\_est::mqc\\_scf\\_integral\\_diagonalize](#) (integral, eVals, eVecs)
- type(mqc\_scf\_integral) function [mqc\\_est::mqc\\_scf\\_integral\\_inverse](#) (integral)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_scf\\_integral\\_trace](#) (integral)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_scf\\_integral\\_determinant](#) (integral)
- subroutine [mqc\\_est::mqc\\_integral\\_set\\_energy\\_list](#) (integral, elist)
- integer(kind=int64) function, dimension(:), allocatable [mqc\\_est::mqc\\_integral\\_get\\_energy\\_list](#) (integral)
- subroutine [mqc\\_est::mqc\\_integral\\_delete\\_energy\\_list](#) (integral)
- subroutine [mqc\\_est::mqc\\_scf\\_eigenvalues\\_power](#) (eigenvalues, power)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_twoeris\\_at](#) (twoERIs, i, j, k, l, spinBlock)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_integral\\_at](#) (integral, i, j, spinBlock)
- type(mqc\_scalar) function [mqc\\_est::mqc\\_eigenvalues\\_at](#) (eigenvalues, i, spinBlock)
- subroutine [mqc\\_est::mqc\\_scf\\_transformation\\_matrix](#) (overlap, transform\_matrix, nBasUse)
- subroutine [mqc\\_est::gen\\_det\\_str](#) (IOut, IPrint, NBasisIn, NAlphaIn, NBetaIn, Determinants, NCoreIn)
- type(mqc\_scalar) function [mqc\\_est::slater\\_condon](#) (IOut, IPrint, NBasisIn, Determinants, L\_A\_String, L\_B\_String, R\_A\_String, R\_B\_String, Core\_Hamiltonian, ERIs, UHF)
- subroutine [mqc\\_est::twoeri\\_trans](#) (IOut, IPrint, MO\_Coeff, ERIs, MO\_ERIs, UHF)
- subroutine [mqc\\_est::mqc\\_build\\_ci\\_hamiltonian](#) (IOut, IPrint, NBasis, Determinants, MO\_Core\_Ham, MO\_ERIs, UHF, CI\_Hamiltonian)
- type(mqc\_matrix) function [mqc\\_est::get\\_one\\_gamma\\_matrix](#) (iOut, iPrint, nBasisIn, nState, determinants, ci\_← amplitudes, nCoreIn, nOrbsIn)





# Index

- abs
  - mqc\_algebra::mqc\_scalar, 112
  - mqc\_algebra::mqc\_vector, 119
- addlabel
  - mqc\_est::mqc\_scf\_eigenvalues, 113
  - mqc\_est::mqc\_scf\_integral, 114
- alpha
  - mqc\_est::mqc\_determinant\_string, 97
- argsort
  - mqc\_algebra::mqc\_vector, 119
- at
  - mqc\_algebra::mqc\_matrix, 100
  - mqc\_algebra::mqc\_r4tensor, 111
  - mqc\_algebra::mqc\_vector, 119
  - mqc\_est::mqc\_scf\_eigenvalues, 113
- basis
  - mqc\_est::mqc\_wavefunction, 124
- beta
  - mqc\_est::mqc\_determinant\_string, 97
- bin\_coeff
  - mqc\_algebra, 15
- charge
  - mqc\_est::mqc\_wavefunction, 124
- core\_hamiltonian
  - mqc\_est::mqc\_wavefunction, 124
- cval
  - mqc\_algebra::mqc\_scalar, 112
- dagger
  - mqc\_algebra::mqc\_matrix, 100
  - mqc\_algebra::mqc\_vector, 119
- data\_type
  - mqc\_algebra::mqc\_vector, 122
- deleteelist
  - mqc\_est::mqc\_scf\_integral, 114
- density\_matrix
  - mqc\_est::mqc\_wavefunction, 125
- det
  - mqc\_algebra::mqc\_matrix, 100
  - mqc\_est::mqc\_scf\_integral, 114
- diag
  - mqc\_algebra::mqc\_matrix, 100
  - mqc\_algebra::mqc\_vector, 119
  - mqc\_est::mqc\_scf\_integral, 115
- eigensys
  - mqc\_algebra::mqc\_matrix, 100
  - mqc\_est::mqc\_scf\_integral, 115
- factorial
  - mqc\_algebra, 15
- fock\_matrix
  - mqc\_est::mqc\_wavefunction, 125
- gen\_det\_str
  - mqc\_est, 62
- get\_one\_gamma\_matrix
  - mqc\_est, 62
- getblock
  - mqc\_est::mqc\_scf\_eigenvalues, 113
  - mqc\_est::mqc\_scf\_integral, 115
- getelist
  - mqc\_est::mqc\_scf\_integral, 115
- getlabel
  - mqc\_est::mqc\_scf\_eigenvalues, 113
  - mqc\_est::mqc\_scf\_integral, 115
- identity
  - mqc\_algebra::mqc\_matrix, 100
  - mqc\_est::mqc\_scf\_integral, 115
- init
  - mqc\_algebra::mqc\_matrix, 101
  - mqc\_algebra::mqc\_r4tensor, 111
  - mqc\_algebra::mqc\_vector, 119
  - mqc\_est::mqc\_scf\_integral, 115
- initialize
  - mqc\_algebra::mqc\_matrix, 101
  - mqc\_algebra::mqc\_r4tensor, 111
  - mqc\_algebra::mqc\_vector, 120
- inv
  - mqc\_algebra::mqc\_matrix, 101
  - mqc\_est::mqc\_scf\_integral, 116
- ival
  - mqc\_algebra::mqc\_scalar, 112
- length
  - mqc\_algebra::mqc\_vector, 122
- mat
  - mqc\_algebra::mqc\_matrix, 101
- matc

- mqc\_algebra::mqc\_matrix, 103
- mati
  - mqc\_algebra::mqc\_matrix, 103
- matr
  - mqc\_algebra::mqc\_matrix, 103
- matrix\_symm2sq\_complex
  - mqc\_algebra, 15
  - mqc\_algebra::matrix\_symm2sq, 94
- matrix\_symm2sq\_integer
  - mqc\_algebra, 15
  - mqc\_algebra::matrix\_symm2sq, 94
- matrix\_symm2sq\_real
  - mqc\_algebra, 16
  - mqc\_algebra::matrix\_symm2sq, 94
- maxloc
  - mqc\_algebra::mqc\_vector, 120
- maxval
  - mqc\_algebra::mqc\_vector, 120
- minloc
  - mqc\_algebra::mqc\_vector, 120
- minval
  - mqc\_algebra::mqc\_vector, 120
- mo\_coefficients
  - mqc\_est::mqc\_wavefunction, 125
- mo\_energies
  - mqc\_est::mqc\_wavefunction, 125
- mo\_symmetries
  - mqc\_est::mqc\_wavefunction, 125
- mput
  - mqc\_algebra::mqc\_matrix, 101
- mqc\_algebra, 9
  - bin\_coeff, 15
  - factorial, 15
  - matrix\_symm2sq\_complex, 15
  - matrix\_symm2sq\_integer, 15
  - matrix\_symm2sq\_real, 16
  - mqc\_allocate\_matrix, 16
  - mqc\_allocate\_r4tensor, 16
  - mqc\_allocate\_scalar, 16
  - mqc\_allocate\_vector, 17
  - mqc\_complexscalaradd, 17
  - mqc\_complexscalardivide, 18
  - mqc\_complexscalarmultiply, 18
  - mqc\_complexscalarsubtract, 18
  - mqc\_complexvectorproduct, 18
  - mqc\_crossproduct, 18
  - mqc\_deallocate\_matrix, 18
  - mqc\_deallocate\_r4tensor, 19
  - mqc\_deallocate\_scalar, 19
  - mqc\_deallocate\_vector, 19
  - mqc\_elementmatrixdivide, 19
  - mqc\_elementmatrixproduct, 19
  - mqc\_elementvectorproduct, 19
  - mqc\_givens\_matrix, 20
  - mqc\_input\_complex\_scalar, 20
  - mqc\_input\_integer\_scalar, 20
  - mqc\_input\_real\_scalar, 21
  - mqc\_integertscalar, 21
  - mqc\_integerlescalar, 21
  - mqc\_integerscalaradd, 21
  - mqc\_integerscalardivide, 21
  - mqc\_integerscalarmultiply, 22
  - mqc\_integerscalarsubtract, 22
  - mqc\_integervectorproduct, 22
  - mqc\_length\_vector, 22
  - mqc\_matrix\_cast\_complex, 22
  - mqc\_matrix\_cast\_real, 22
  - mqc\_matrix\_columns, 23
  - mqc\_matrix\_conjugate\_transpose, 23
  - mqc\_matrix\_copy\_complex2int, 23
  - mqc\_matrix\_copy\_complex2real, 23
  - mqc\_matrix\_copy\_int2complex, 23
  - mqc\_matrix\_copy\_int2real, 23
  - mqc\_matrix\_copy\_real2complex, 24
  - mqc\_matrix\_copy\_real2int, 24
  - mqc\_matrix\_determinant, 24
  - mqc\_matrix\_diag2full, 24
  - mqc\_matrix\_diag2symm, 24
  - mqc\_matrix\_diagmatrix\_put\_complex, 24
  - mqc\_matrix\_diagmatrix\_put\_integer, 25
  - mqc\_matrix\_diagmatrix\_put\_real, 25
  - mqc\_matrix\_diagmatrix\_put\_vector, 25
  - mqc\_matrix\_diagonalize, 25
  - mqc\_matrix\_full2diag, 25
  - mqc\_matrix\_full2symm, 25
  - mqc\_matrix\_generalized\_eigensystem, 26
  - mqc\_matrix\_havecomplex, 26
  - mqc\_matrix\_havediagonal, 26
  - mqc\_matrix\_havetfull, 26
  - mqc\_matrix\_haveinteger, 26
  - mqc\_matrix\_havereal, 26
  - mqc\_matrix\_havesymmetric, 27
  - mqc\_matrix\_identity, 27
  - mqc\_matrix\_initialize, 27
  - mqc\_matrix\_inverse, 27
  - mqc\_matrix\_isallocated, 27
  - mqc\_matrix\_matrix\_at, 27
  - mqc\_matrix\_matrix\_contraction, 29
  - mqc\_matrix\_matrix\_put, 29
  - mqc\_matrix\_norm, 29
  - mqc\_matrix\_rms\_max, 30
  - mqc\_matrix\_rows, 30
  - mqc\_matrix\_scalar\_at, 30
  - mqc\_matrix\_scalar\_put, 30
  - mqc\_matrix\_set, 30
  - mqc\_matrix\_sqrt, 31
  - mqc\_matrix\_storagetype, 31
  - mqc\_matrix\_svd, 31

`mqc_matrix_symm2diag`, 31  
`mqc_matrix_symm2full`, 31  
`mqc_matrix_symm2full_func`, 32  
`mqc_matrix_symmetrize`, 32  
`mqc_matrix_symmmatrix_put_complex`, 32  
`mqc_matrix_symmmatrix_put_integer`, 32  
`mqc_matrix_symmmatrix_put_real`, 32  
`mqc_matrix_symmsymmr4tensor_put_complex`, 32  
`mqc_matrix_symmsymmr4tensor_put_real`, 33  
`mqc_matrix_test_diagonal`, 33  
`mqc_matrix_test_symmetric`, 33  
`mqc_matrix_trace`, 33  
`mqc_matrix_transpose`, 33  
`mqc_matrix_vector_at`, 33  
`mqc_matrix_vector_put`, 34  
`mqc_matrixmatrixdotproduct`, 34  
`mqc_matrixmatrixproduct`, 34  
`mqc_matrixmatrixsubtract`, 34  
`mqc_matrixmatrixsum`, 34  
`mqc_matrixscalarproduct`, 35  
`mqc_matrixvectordotproduct`, 35  
`mqc_outer`, 35  
`mqc_output_complex_scalar`, 35  
`mqc_output_integer_scalar`, 35  
`mqc_output_mqcscalar_scalar`, 35  
`mqc_output_real_scalar`, 36  
`mqc_print_matrix_algebra1`, 36  
`mqc_print_r4tensor_algebra1`, 36  
`mqc_print_scalar_algebra1`, 36  
`mqc_print_vector_algebra1`, 36  
`mqc_r4tensor_at`, 37  
`mqc_r4tensor_havecomplex`, 37  
`mqc_r4tensor_haveinteger`, 37  
`mqc_r4tensor_havereal`, 37  
`mqc_r4tensor_initialize`, 37  
`mqc_r4tensor_put`, 38  
`mqc_realgtscalar`, 38  
`mqc_realltscalar`, 38  
`mqc_realscalaradd`, 38  
`mqc_realscalardivide`, 39  
`mqc_realscalarmultiply`, 39  
`mqc_realscalarsubtract`, 39  
`mqc_realvectorproduct`, 39  
`mqc_scalar_acos`, 39  
`mqc_scalar_asin`, 39  
`mqc_scalar_atan`, 40  
`mqc_scalar_atan2`, 40  
`mqc_scalar_cmplx`, 40  
`mqc_scalar_complex_conjugate`, 40  
`mqc_scalar_complex_imagpart`, 40  
`mqc_scalar_complex_realpart`, 40  
`mqc_scalar_cos`, 41  
`mqc_scalar_get_abs_value`, 41  
`mqc_scalar_get_intrinsic_complex`, 41  
`mqc_scalar_get_intrinsic_integer`, 41  
`mqc_scalar_get_intrinsic_real`, 41  
`mqc_scalar_get_random_value`, 41  
`mqc_scalar_havecomplex`, 42  
`mqc_scalar_haveinteger`, 42  
`mqc_scalar_havereal`, 42  
`mqc_scalar_isallocated`, 42  
`mqc_scalar_sin`, 42  
`mqc_scalar_sqrt`, 42  
`mqc_scalar_tan`, 43  
`mqc_scalaradd`, 43  
`mqc_scalarcomplexadd`, 43  
`mqc_scalarcomplexdivide`, 43  
`mqc_scalarcomplexexponent`, 43  
`mqc_scalarcomplexmultiply`, 43  
`mqc_scalarcomplexsubtract`, 44  
`mqc_scalardivide`, 44  
`mqc_scalareq`, 44  
`mqc_scalarexponent`, 44  
`mqc_scalarge`, 44  
`mqc_scalargt`, 44  
`mqc_scalargtinteger`, 45  
`mqc_scalargtreal`, 45  
`mqc_scalarintegeradd`, 45  
`mqc_scalarintegerdivide`, 45  
`mqc_scalarintegerexponent`, 45  
`mqc_scalarintegermultiply`, 45  
`mqc_scalarintegersubtract`, 46  
`mqc_scalarle`, 46  
`mqc_scalarleinteger`, 46  
`mqc_scalarlereal`, 46  
`mqc_scalarlt`, 46  
`mqc_scalarltreal`, 46  
`mqc_scalarmatrixproduct`, 47  
`mqc_scalarmultiply`, 47  
`mqc_scalarne`, 47  
`mqc_scalarrealadd`, 47  
`mqc_scalarrealdivide`, 47  
`mqc_scalarrealexponent`, 47  
`mqc_scalarrealmultiply`, 48  
`mqc_scalarrealsubtract`, 48  
`mqc_scalarsubtract`, 48  
`mqc_scalarvectordifference`, 48  
`mqc_scalarvectorproduct`, 48  
`mqc_scalarvectorsum`, 48  
`mqc_set_array2tensor`, 49  
`mqc_set_array2vector_complex`, 49  
`mqc_set_array2vector_integer`, 49  
`mqc_set_array2vector_real`, 49  
`mqc_set_complexarray2matrix`, 49  
`mqc_set_integerarray2matrix`, 49  
`mqc_set_matrix2complexarray`, 50  
`mqc_set_matrix2integerarray`, 50

- mqc\_set\_matrix2matrix, 50
- mqc\_set\_matrix2realarray, 50
- mqc\_set\_realarray2matrix, 50
- mqc\_set\_vector2complexarray, 50
- mqc\_set\_vector2integerarray, 51
- mqc\_set\_vector2realarray, 51
- mqc\_set\_vector2vector, 51
- mqc\_vector2diagmatrix, 51
- mqc\_vector\_abs, 51
- mqc\_vector\_argsort, 51
- mqc\_vector\_cast\_complex, 52
- mqc\_vector\_cast\_real, 52
- mqc\_vector\_cmplx, 52
- mqc\_vector\_complex\_imagpart, 52
- mqc\_vector\_complex\_realpart, 52
- mqc\_vector\_conjugate\_transpose, 52
- mqc\_vector\_copy\_complex2int, 53
- mqc\_vector\_copy\_complex2real, 53
- mqc\_vector\_copy\_int2complex, 53
- mqc\_vector\_copy\_int2real, 53
- mqc\_vector\_copy\_real2complex, 53
- mqc\_vector\_copy\_real2int, 53
- mqc\_vector\_havecomplex, 54
- mqc\_vector\_haveinteger, 54
- mqc\_vector\_havereal, 54
- mqc\_vector\_initialize, 54
- mqc\_vector\_isallocated, 54
- mqc\_vector\_iscolumn, 54
- mqc\_vector\_maxloc, 55
- mqc\_vector\_maxval, 55
- mqc\_vector\_minloc, 55
- mqc\_vector\_minval, 55
- mqc\_vector\_norm, 55
- mqc\_vector\_pop, 55
- mqc\_vector\_power, 56
- mqc\_vector\_push, 56
- mqc\_vector\_scalar\_at, 56
- mqc\_vector\_scalar\_increment, 56
- mqc\_vector\_scalar\_put, 56
- mqc\_vector\_shift, 56
- mqc\_vector\_sort, 57
- mqc\_vector\_sqrt, 57
- mqc\_vector\_transpose, 57
- mqc\_vector\_unshift, 57
- mqc\_vector\_vector\_at, 57
- mqc\_vector\_vector\_put, 57
- mqc\_vectorcomplexdivide, 58
- mqc\_vectorcomplexproduct, 58
- mqc\_vectorintegerdivide, 58
- mqc\_vectorintegerproduct, 58
- mqc\_vectormatrixdotproduct, 58
- mqc\_vectorrealdivide, 58
- mqc\_vectorrealproduct, 59
- mqc\_vectorscalardivide, 59
- mqc\_vectorscalarproduct, 59
- mqc\_vectorvectordifference, 59
- mqc\_vectorvectordotproduct, 59
- mqc\_vectorvectorsum, 59
- symindexhash, 60
- mqc\_algebra::abs, 77
  - mqc\_scalar\_get\_abs\_value, 77
  - mqc\_vector\_abs, 77
- mqc\_algebra::acos, 78
  - mqc\_scalar\_acos, 78
- mqc\_algebra::aimag, 78
  - mqc\_scalar\_complex\_imagpart, 78
  - mqc\_vector\_complex\_imagpart, 78
- mqc\_algebra::asin, 79
  - mqc\_scalar\_asin, 79
- mqc\_algebra::assignment(=), 79
  - mqc\_input\_complex\_scalar, 80
  - mqc\_input\_integer\_scalar, 80
  - mqc\_input\_real\_scalar, 81
  - mqc\_output\_complex\_scalar, 81
  - mqc\_output\_integer\_scalar, 81
  - mqc\_output\_mqcscalar\_scalar, 81
  - mqc\_output\_real\_scalar, 81
  - mqc\_set\_array2tensor, 82
  - mqc\_set\_array2vector\_complex, 82
  - mqc\_set\_array2vector\_integer, 82
  - mqc\_set\_array2vector\_real, 82
  - mqc\_set\_complexarray2matrix, 82
  - mqc\_set\_integerarray2matrix, 82
  - mqc\_set\_matrix2complexarray, 83
  - mqc\_set\_matrix2integerarray, 83
  - mqc\_set\_matrix2matrix, 83
  - mqc\_set\_matrix2realarray, 83
  - mqc\_set\_realarray2matrix, 83
  - mqc\_set\_vector2complexarray, 83
  - mqc\_set\_vector2integerarray, 84
  - mqc\_set\_vector2realarray, 84
  - mqc\_set\_vector2vector, 84
- mqc\_algebra::atan, 85
  - mqc\_scalar\_atan, 85
- mqc\_algebra::atan2, 85
  - mqc\_scalar\_atan2, 86
- mqc\_algebra::cmplx, 86
  - mqc\_scalar\_cmplx, 86
  - mqc\_vector\_cmplx, 86
- mqc\_algebra::conjg, 87
  - mqc\_scalar\_complex\_conjugate, 87
- mqc\_algebra::contraction, 87
  - mqc\_matrix\_matrix\_contraction, 87
- mqc\_algebra::cos, 88
  - mqc\_scalar\_cos, 88
- mqc\_algebra::dagger, 89
  - mqc\_matrix\_conjugate\_transpose, 89
  - mqc\_vector\_conjugate\_transpose, 89

- mqc\_algebra::dot\_product, 90
  - mqc\_vectorvectordotproduct, 90
- mqc\_algebra::matmul, 91
  - mqc\_matrixmatrixdotproduct, 91
  - mqc\_matrixvectordotproduct, 91
  - mqc\_vectormatrixdotproduct, 91
- mqc\_algebra::matrix\_symm2sq, 93
  - matrix\_symm2sq\_complex, 94
  - matrix\_symm2sq\_integer, 94
  - matrix\_symm2sq\_real, 94
- mqc\_algebra::mqc\_cast\_complex, 94
  - mqc\_matrix\_cast\_complex, 94
  - mqc\_vector\_cast\_complex, 95
- mqc\_algebra::mqc\_cast\_real, 95
  - mqc\_matrix\_cast\_real, 95
  - mqc\_vector\_cast\_real, 95
- mqc\_algebra::mqc\_have\_complex, 97
  - mqc\_matrix\_havecomplex, 97
  - mqc\_vector\_havecomplex, 97
- mqc\_algebra::mqc\_have\_int, 98
  - mqc\_matrix\_haveinteger, 98
  - mqc\_vector\_haveinteger, 98
- mqc\_algebra::mqc\_have\_real, 98
  - mqc\_matrix\_havereal, 99
  - mqc\_vector\_havereal, 99
- mqc\_algebra::mqc\_matrix, 99
  - at, 100
  - dagger, 100
  - det, 100
  - diag, 100
  - eigensys, 100
  - identity, 100
  - init, 101
  - initialize, 101
  - inv, 101
  - mat, 101
  - matc, 103
  - mati, 103
  - matr, 103
  - mput, 101
  - norm, 101
  - print, 101
  - put, 102
  - rmsmax, 102
  - s\_type, 102
  - set, 102
  - sqrt, 102
  - svd, 102
  - trace, 102
  - transpose, 103
  - vat, 103
  - vput, 103
- mqc\_algebra::mqc\_matrix\_diagmatrix\_put, 104
  - mqc\_matrix\_diagmatrix\_put\_complex, 104
  - mqc\_matrix\_diagmatrix\_put\_integer, 104
  - mqc\_matrix\_diagmatrix\_put\_real, 104
  - mqc\_matrix\_diagmatrix\_put\_vector, 104
- mqc\_algebra::mqc\_matrix\_symmmatrix\_put, 105
  - mqc\_matrix\_symmmatrix\_put\_complex, 105
  - mqc\_matrix\_symmmatrix\_put\_integer, 105
  - mqc\_matrix\_symmmatrix\_put\_real, 105
- mqc\_algebra::mqc\_print, 106
  - mqc\_print\_matrix\_algebra1, 106
  - mqc\_print\_r4tensor\_algebra1, 107
  - mqc\_print\_scalar\_algebra1, 107
  - mqc\_print\_vector\_algebra1, 107
- mqc\_algebra::mqc\_r4tensor, 110
  - at, 111
  - init, 111
  - initialize, 111
  - print, 111
  - put, 111
- mqc\_algebra::mqc\_scalar, 111
  - abs, 112
  - cval, 112
  - ival, 112
  - print, 112
  - random, 112
  - rval, 112
- mqc\_algebra::mqc\_set\_array2vector, 117
  - mqc\_set\_array2vector\_complex, 117
  - mqc\_set\_array2vector\_integer, 117
  - mqc\_set\_array2vector\_real, 117
- mqc\_algebra::mqc\_vector, 118
  - abs, 119
  - argsort, 119
  - at, 119
  - dagger, 119
  - data\_type, 122
  - diag, 119
  - init, 119
  - initialize, 120
  - length, 122
  - maxloc, 120
  - maxval, 120
  - minloc, 120
  - minval, 120
  - norm, 120
  - pop, 120
  - power, 121
  - print, 121
  - push, 121
  - put, 121
  - shift, 121
  - size, 121
  - sort, 121
  - sqrt, 122
  - transpose, 122

- unshift, 122
- vat, 122
- vecc, 123
- veci, 123
- vecr, 123
- vput, 122
- mqc\_algebra::operator(\*\*), 131
  - mqc\_scalarcomplexexponent, 131
  - mqc\_scalarexponent, 132
  - mqc\_scalarintegerexponent, 132
  - mqc\_scalarrealexponent, 132
- mqc\_algebra::operator(\*), 127
  - mqc\_complexscalarmultiply, 127
  - mqc\_complexvectorproduct, 128
  - mqc\_integerscalarmultiply, 128
  - mqc\_integervectorproduct, 128
  - mqc\_matrixmatrixproduct, 128
  - mqc\_matrixscalarproduct, 128
  - mqc\_realscalarmultiply, 128
  - mqc\_realvectorproduct, 129
  - mqc\_scalarcomplexmultiply, 129
  - mqc\_scalarintegermultiply, 129
  - mqc\_scalarmatrixproduct, 129
  - mqc\_scalarmultiply, 129
  - mqc\_scalarrealmultiply, 129
  - mqc\_scalarvectorproduct, 130
  - mqc\_vectorcomplexproduct, 130
  - mqc\_vectorintegerproduct, 130
  - mqc\_vectorrealproduct, 130
  - mqc\_vectorscalarproduct, 130
- mqc\_algebra::operator(+), 133
  - mqc\_complexscalaradd, 133
  - mqc\_integerscalaradd, 133
  - mqc\_matrixmatrixsum, 133
  - mqc\_realscalaradd, 134
  - mqc\_scalaradd, 134
  - mqc\_scalarcomplexadd, 134
  - mqc\_scalarintegeradd, 134
  - mqc\_scalarrealadd, 134
  - mqc\_scalarvectorsum, 134
  - mqc\_vectorvectorsum, 135
- mqc\_algebra::operator(-), 136
  - mqc\_complexscalarsubtract, 136
  - mqc\_integerscalarsubtract, 136
  - mqc\_matrixmatrixsubtract, 136
  - mqc\_realscalarsubtract, 136
  - mqc\_scalarcomplexsubtract, 137
  - mqc\_scalarintegersubtract, 137
  - mqc\_scalarrealsubtract, 137
  - mqc\_scalarsubtract, 137
  - mqc\_scalarvectordifference, 137
  - mqc\_vectorvectordifference, 137
- mqc\_algebra::operator(.dot.), 138
  - mqc\_matrixmatrixdotproduct, 138
  - mqc\_matrixvectordotproduct, 138
  - mqc\_vectormatrixdotproduct, 138
  - mqc\_vectorvectordotproduct, 138
- mqc\_algebra::operator(.eq.), 139
  - mqc\_scalareq, 139
- mqc\_algebra::operator(.ewd.), 139
  - mqc\_elementmatrixdivide, 139
- mqc\_algebra::operator(.ewp.), 140
  - mqc\_elementmatrixproduct, 140
  - mqc\_elementvectorproduct, 140
- mqc\_algebra::operator(.ge.), 140
  - mqc\_scalarge, 141
- mqc\_algebra::operator(.gt.), 141
  - mqc\_integergtscalar, 141
  - mqc\_realgtscalar, 141
  - mqc\_scalargt, 141
  - mqc\_scalargtinteger, 142
  - mqc\_scalargtreal, 142
- mqc\_algebra::operator(.le.), 142
  - mqc\_integerlescalar, 142
  - mqc\_reallescalar, 143
  - mqc\_scalarle, 143
  - mqc\_scalarleinteger, 143
  - mqc\_scalarlereal, 143
- mqc\_algebra::operator(.lt.), 144
  - mqc\_realltscalar, 144
  - mqc\_scalarlt, 144
  - mqc\_scalarltreal, 144
- mqc\_algebra::operator(.ne.), 144
  - mqc\_scalarne, 145
- mqc\_algebra::operator(.outer.), 145
  - mqc\_outer, 145
- mqc\_algebra::operator(.x.), 145
  - mqc\_crossproduct, 146
- mqc\_algebra::operator(/), 146
  - mqc\_complexscalardivide, 146
  - mqc\_integerscalardivide, 146
  - mqc\_realscalardivide, 147
  - mqc\_scalarcomplexdivide, 147
  - mqc\_scalardivide, 147
  - mqc\_scalarintegerdivide, 147
  - mqc\_scalarrealddivide, 147
  - mqc\_vectorcomplexdivide, 147
  - mqc\_vectorintegerdivide, 148
  - mqc\_vectorrealddivide, 148
  - mqc\_vectorscalardivide, 148
- mqc\_algebra::real, 148
  - mqc\_scalar\_complex\_realpart, 148
  - mqc\_vector\_complex\_realpart, 149
- mqc\_algebra::sin, 149
  - mqc\_scalar\_sin, 149
- mqc\_algebra::sqrt, 149
  - mqc\_scalar\_sqrt, 150
- mqc\_algebra::tan, 150

- mqc\_scalar\_tan, [150](#)
- mqc\_algebra::transpose, [151](#)
  - mqc\_matrix\_transpose, [151](#)
  - mqc\_vector\_transpose, [151](#)
- mqc\_allocate\_matrix
  - mqc\_algebra, [16](#)
- mqc\_allocate\_r4tensor
  - mqc\_algebra, [16](#)
- mqc\_allocate\_scalar
  - mqc\_algebra, [16](#)
- mqc\_allocate\_vector
  - mqc\_algebra, [17](#)
- mqc\_build\_ci\_hamiltonian
  - mqc\_est, [62](#)
- mqc\_complexscalaradd
  - mqc\_algebra, [17](#)
  - mqc\_algebra::operator(+), [133](#)
- mqc\_complexscalardivide
  - mqc\_algebra, [18](#)
  - mqc\_algebra::operator(/), [146](#)
- mqc\_complexscalarmultiply
  - mqc\_algebra, [18](#)
  - mqc\_algebra::operator(\*), [127](#)
- mqc\_complexscalarsubtract
  - mqc\_algebra, [18](#)
  - mqc\_algebra::operator(-), [136](#)
- mqc\_complexvectorproduct
  - mqc\_algebra, [18](#)
  - mqc\_algebra::operator(\*), [128](#)
- mqc\_crossproduct
  - mqc\_algebra, [18](#)
  - mqc\_algebra::operator(.x.), [146](#)
- mqc\_deallocate\_matrix
  - mqc\_algebra, [18](#)
- mqc\_deallocate\_r4tensor
  - mqc\_algebra, [19](#)
- mqc\_deallocate\_scalar
  - mqc\_algebra, [19](#)
- mqc\_deallocate\_vector
  - mqc\_algebra, [19](#)
- mqc\_eigenvalue\_eigenvalue\_dotproduct
  - mqc\_est, [63](#)
  - mqc\_est::dot\_product, [91](#)
- mqc\_eigenvalues\_add\_name
  - mqc\_est, [63](#)
- mqc\_eigenvalues\_allocate
  - mqc\_est, [63](#)
- mqc\_eigenvalues\_array\_name
  - mqc\_est, [63](#)
- mqc\_eigenvalues\_array\_type
  - mqc\_est, [63](#)
- mqc\_eigenvalues\_at
  - mqc\_est, [64](#)
- mqc\_eigenvalues\_dimension
  - mqc\_est, [64](#)
- mqc\_eigenvalues\_eigenvalues\_multiply
  - mqc\_est, [64](#)
- mqc\_eigenvalues\_has\_alpha
  - mqc\_est, [64](#)
- mqc\_eigenvalues\_has\_beta
  - mqc\_est, [64](#)
- mqc\_eigenvalues\_integral\_multiply
  - mqc\_est, [64](#)
  - mqc\_est::matmul, [92](#)
- mqc\_eigenvalues\_isallocated
  - mqc\_est, [65](#)
- mqc\_eigenvalues\_output\_array
  - mqc\_est, [65](#)
  - mqc\_est::assignment(=), [84](#)
- mqc\_eigenvalues\_output\_block
  - mqc\_est, [65](#)
- mqc\_elementmatrixdivide
  - mqc\_algebra, [19](#)
  - mqc\_algebra::operator(.ewd.), [139](#)
- mqc\_elementmatrixproduct
  - mqc\_algebra, [19](#)
  - mqc\_algebra::operator(.ewp.), [140](#)
- mqc\_elementvectorproduct
  - mqc\_algebra, [19](#)
  - mqc\_algebra::operator(.ewp.), [140](#)
- mqc\_eri\_integral\_contraction
  - mqc\_est, [65](#)
  - mqc\_est::contraction, [88](#)
- mqc\_est, [60](#)
  - gen\_det\_str, [62](#)
  - get\_one\_gamma\_matrix, [62](#)
  - mqc\_build\_ci\_hamiltonian, [62](#)
  - mqc\_eigenvalue\_eigenvalue\_dotproduct, [63](#)
  - mqc\_eigenvalues\_add\_name, [63](#)
  - mqc\_eigenvalues\_allocate, [63](#)
  - mqc\_eigenvalues\_array\_name, [63](#)
  - mqc\_eigenvalues\_array\_type, [63](#)
  - mqc\_eigenvalues\_at, [64](#)
  - mqc\_eigenvalues\_dimension, [64](#)
  - mqc\_eigenvalues\_eigenvalues\_multiply, [64](#)
  - mqc\_eigenvalues\_has\_alpha, [64](#)
  - mqc\_eigenvalues\_has\_beta, [64](#)
  - mqc\_eigenvalues\_integral\_multiply, [64](#)
  - mqc\_eigenvalues\_isallocated, [65](#)
  - mqc\_eigenvalues\_output\_array, [65](#)
  - mqc\_eigenvalues\_output\_block, [65](#)
  - mqc\_eri\_integral\_contraction, [65](#)
  - mqc\_integral\_add\_name, [65](#)
  - mqc\_integral\_allocate, [65](#)
  - mqc\_integral\_array\_name, [66](#)
  - mqc\_integral\_array\_type, [66](#)
  - mqc\_integral\_at, [66](#)



- mqc\_integral\_conjugate\_transpose, 66
- mqc\_integral\_delete\_energy\_list, 66
- mqc\_integral\_difference, 67
- mqc\_integral\_dimension, 67
- mqc\_integral\_eigenvalues\_multiply, 67
- mqc\_integral\_get\_energy\_list, 67
- mqc\_integral\_has\_alpha, 67
- mqc\_integral\_has\_alphabeta, 67
- mqc\_integral\_has\_beta, 68
- mqc\_integral\_has\_betaalpha, 68
- mqc\_integral\_identity, 68
- mqc\_integral\_initialize, 68
- mqc\_integral\_integral\_multiply, 68
- mqc\_integral\_isallocated, 69
- mqc\_integral\_matrix\_multiply, 69
- mqc\_integral\_norm, 69
- mqc\_integral\_output\_array, 69
- mqc\_integral\_output\_block, 69
- mqc\_integral\_output\_orbitals, 69
- mqc\_integral\_scalar\_multiply, 70
- mqc\_integral\_set\_energy\_list, 70
- mqc\_integral\_sum, 70
- mqc\_integral\_swap\_orbitals, 70
- mqc\_integral\_transpose, 70
- mqc\_matrix\_integral\_multiply, 71
- mqc\_matrix\_spinblockghf, 71
- mqc\_matrix\_undospinblockghf\_eigenvalues, 71
- mqc\_matrix\_undospinblockghf\_integral, 71
- mqc\_print\_eigenvalues, 71
- mqc\_print\_integral, 72
- mqc\_print\_twoeris, 72
- mqc\_print\_wavefunction, 72
- mqc\_scalar\_integral\_multiply, 72
- mqc\_scf\_eigenvalues\_power, 72
- mqc\_scf\_integral\_contraction, 73
- mqc\_scf\_integral\_determinant, 73
- mqc\_scf\_integral\_diagonalize, 73
- mqc\_scf\_integral\_generalized\_eigensystem, 73
- mqc\_scf\_integral\_inverse, 73
- mqc\_scf\_integral\_trace, 74
- mqc\_scf\_transformation\_matrix, 74
- mqc\_twoeris\_allocate, 74
- mqc\_twoeris\_at, 74
- slater\_condon, 74
- twoeri\_trans, 75
- mqc\_est::assignment(=), 84
  - mqc\_eigenvalues\_output\_array, 84
  - mqc\_integral\_output\_array, 85
- mqc\_est::contraction, 88
  - mqc\_eri\_integral\_contraction, 88
  - mqc\_scf\_integral\_contraction, 88
- mqc\_est::dagger, 89
  - mqc\_integral\_conjugate\_transpose, 90
- mqc\_est::dot\_product, 90
  - mqc\_eigenvalue\_eigenvalue\_dotproduct, 91
- mqc\_est::matmul, 92
  - mqc\_eigenvalues\_eigenvalues\_multiply, 92
  - mqc\_eigenvalues\_integral\_multiply, 92
  - mqc\_integral\_eigenvalues\_multiply, 92
  - mqc\_integral\_integral\_multiply, 93
  - mqc\_integral\_matrix\_multiply, 93
  - mqc\_matrix\_integral\_multiply, 93
- mqc\_est::mqc\_determinant, 96
  - nalpstr, 96
  - nbetstr, 96
  - ndets, 96
  - order, 96
  - strings, 96
- mqc\_est::mqc\_determinant\_string, 97
  - alpha, 97
  - beta, 97
- mqc\_est::mqc\_matrix\_undospinblockghf, 106
  - mqc\_matrix\_undospinblockghf\_eigenvalues, 106
  - mqc\_matrix\_undospinblockghf\_integral, 106
- mqc\_est::mqc\_print, 108
  - mqc\_print\_eigenvalues, 108
  - mqc\_print\_integral, 108
  - mqc\_print\_twoeris, 108
  - mqc\_print\_wavefunction, 108
- mqc\_est::mqc\_pscf\_wavefunction, 109
  - nactive, 109
  - ncore, 109
  - nfrz, 110
  - nval, 110
  - pscf\_amplitudes, 110
  - pscf\_energies, 110
- mqc\_est::mqc\_scf\_eigenvalues, 113
  - addlabel, 113
  - at, 113
  - getblock, 113
  - getlabel, 113
  - power, 113
  - print, 113
- mqc\_est::mqc\_scf\_integral, 114
  - addlabel, 114
  - deleteelist, 114
  - det, 114
  - diag, 115
  - eigensys, 115
  - getblock, 115
  - getelist, 115
  - getlabel, 115
  - identity, 115
  - init, 115
  - inv, 116
  - norm, 116
  - orbitals, 116
  - print, 116



- setelist, 116
- swap, 116
- trace, 116
- mqc\_est::mqc\_twoeris, 118
- print, 118
- mqc\_est::mqc\_wavefunction, 123
  - basis, 124
  - charge, 124
  - core\_hamiltonian, 124
  - density\_matrix, 125
  - fock\_matrix, 125
  - mo\_coefficients, 125
  - mo\_energies, 125
  - mo\_symmetries, 125
  - multiplicity, 125
  - nalpha, 125
  - nbasis, 126
  - nbeta, 126
  - nelectrons, 126
  - overlap\_matrix, 126
  - print, 124
  - scf\_density\_matrix, 126
  - symmetry, 126
  - wf\_complex, 126
  - wf\_type, 127
- mqc\_est::operator(\*), 131
  - mqc\_integral\_scalar\_multiply, 131
  - mqc\_scalar\_integral\_multiply, 131
- mqc\_est::operator(+), 132
  - mqc\_integral\_sum, 132
- mqc\_est::operator(-), 135
  - mqc\_integral\_difference, 135
- mqc\_est::transpose, 150
  - mqc\_integral\_transpose, 150
- mqc\_givens\_matrix
  - mqc\_algebra, 20
- mqc\_input\_complex\_scalar
  - mqc\_algebra, 20
  - mqc\_algebra::assignment(=), 80
- mqc\_input\_integer\_scalar
  - mqc\_algebra, 20
  - mqc\_algebra::assignment(=), 80
- mqc\_input\_real\_scalar
  - mqc\_algebra, 21
  - mqc\_algebra::assignment(=), 81
- mqc\_integer\_tscalar
  - mqc\_algebra, 21
  - mqc\_algebra::operator(.gt.), 141
- mqc\_integerlescalar
  - mqc\_algebra, 21
  - mqc\_algebra::operator(.le.), 142
- mqc\_integerscalaradd
  - mqc\_algebra, 21
  - mqc\_algebra::operator(+), 133
- mqc\_integerscalardivide
  - mqc\_algebra, 21
  - mqc\_algebra::operator(/), 146
- mqc\_integerscalarmultiply
  - mqc\_algebra, 22
  - mqc\_algebra::operator(\*), 128
- mqc\_integerscalarsubtract
  - mqc\_algebra, 22
  - mqc\_algebra::operator(-), 136
- mqc\_integervectorproduct
  - mqc\_algebra, 22
  - mqc\_algebra::operator(\*), 128
- mqc\_integral\_add\_name
  - mqc\_est, 65
- mqc\_integral\_allocate
  - mqc\_est, 65
- mqc\_integral\_array\_name
  - mqc\_est, 66
- mqc\_integral\_array\_type
  - mqc\_est, 66
- mqc\_integral\_at
  - mqc\_est, 66
- mqc\_integral\_conjugate\_transpose
  - mqc\_est, 66
  - mqc\_est::dagger, 90
- mqc\_integral\_delete\_energy\_list
  - mqc\_est, 66
- mqc\_integral\_difference
  - mqc\_est, 67
  - mqc\_est::operator(-), 135
- mqc\_integral\_dimension
  - mqc\_est, 67
- mqc\_integral\_eigenvalues\_multiply
  - mqc\_est, 67
  - mqc\_est::matmul, 92
- mqc\_integral\_get\_energy\_list
  - mqc\_est, 67
- mqc\_integral\_has\_alpha
  - mqc\_est, 67
- mqc\_integral\_has\_alphabeta
  - mqc\_est, 67
- mqc\_integral\_has\_beta
  - mqc\_est, 68
- mqc\_integral\_has\_betaalpha
  - mqc\_est, 68
- mqc\_integral\_identity
  - mqc\_est, 68
- mqc\_integral\_initialize
  - mqc\_est, 68
- mqc\_integral\_integral\_multiply
  - mqc\_est, 68
  - mqc\_est::matmul, 93
- mqc\_integral\_isallocated
  - mqc\_est, 69

mqc\_integral\_matrix\_multiply  
     mqc\_est, 69  
     mqc\_est::matmul, 93  
 mqc\_integral\_norm  
     mqc\_est, 69  
 mqc\_integral\_output\_array  
     mqc\_est, 69  
     mqc\_est::assignment(=), 85  
 mqc\_integral\_output\_block  
     mqc\_est, 69  
 mqc\_integral\_output\_orbitals  
     mqc\_est, 69  
 mqc\_integral\_scalar\_multiply  
     mqc\_est, 70  
     mqc\_est::operator(\*), 131  
 mqc\_integral\_set\_energy\_list  
     mqc\_est, 70  
 mqc\_integral\_sum  
     mqc\_est, 70  
     mqc\_est::operator(+), 132  
 mqc\_integral\_swap\_orbitals  
     mqc\_est, 70  
 mqc\_integral\_transpose  
     mqc\_est, 70  
     mqc\_est::transpose, 150  
 mqc\_length\_vector  
     mqc\_algebra, 22  
 mqc\_matrix\_cast\_complex  
     mqc\_algebra, 22  
     mqc\_algebra::mqc\_cast\_complex, 94  
 mqc\_matrix\_cast\_real  
     mqc\_algebra, 22  
     mqc\_algebra::mqc\_cast\_real, 95  
 mqc\_matrix\_columns  
     mqc\_algebra, 23  
 mqc\_matrix\_conjugate\_transpose  
     mqc\_algebra, 23  
     mqc\_algebra::dagger, 89  
 mqc\_matrix\_copy\_complex2int  
     mqc\_algebra, 23  
 mqc\_matrix\_copy\_complex2real  
     mqc\_algebra, 23  
 mqc\_matrix\_copy\_int2complex  
     mqc\_algebra, 23  
 mqc\_matrix\_copy\_int2real  
     mqc\_algebra, 23  
 mqc\_matrix\_copy\_real2complex  
     mqc\_algebra, 24  
 mqc\_matrix\_copy\_real2int  
     mqc\_algebra, 24  
 mqc\_matrix\_determinant  
     mqc\_algebra, 24  
 mqc\_matrix\_diag2full  
     mqc\_algebra, 24  
 mqc\_matrix\_diag2symm  
     mqc\_algebra, 24  
 mqc\_matrix\_diagmatrix\_put\_complex  
     mqc\_algebra, 24  
     mqc\_algebra::mqc\_matrix\_diagmatrix\_put, 104  
 mqc\_matrix\_diagmatrix\_put\_integer  
     mqc\_algebra, 25  
     mqc\_algebra::mqc\_matrix\_diagmatrix\_put, 104  
 mqc\_matrix\_diagmatrix\_put\_real  
     mqc\_algebra, 25  
     mqc\_algebra::mqc\_matrix\_diagmatrix\_put, 104  
 mqc\_matrix\_diagmatrix\_put\_vector  
     mqc\_algebra, 25  
     mqc\_algebra::mqc\_matrix\_diagmatrix\_put, 104  
 mqc\_matrix\_diagonalize  
     mqc\_algebra, 25  
 mqc\_matrix\_full2diag  
     mqc\_algebra, 25  
 mqc\_matrix\_full2symm  
     mqc\_algebra, 25  
 mqc\_matrix\_generalized\_eigensystem  
     mqc\_algebra, 26  
 mqc\_matrix\_havecomplex  
     mqc\_algebra, 26  
     mqc\_algebra::mqc\_have\_complex, 97  
 mqc\_matrix\_havediagonal  
     mqc\_algebra, 26  
 mqc\_matrix\_havefull  
     mqc\_algebra, 26  
 mqc\_matrix\_haveinteger  
     mqc\_algebra, 26  
     mqc\_algebra::mqc\_have\_int, 98  
 mqc\_matrix\_havereal  
     mqc\_algebra, 26  
     mqc\_algebra::mqc\_have\_real, 99  
 mqc\_matrix\_havesymmetric  
     mqc\_algebra, 27  
 mqc\_matrix\_identity  
     mqc\_algebra, 27  
 mqc\_matrix\_initialize  
     mqc\_algebra, 27  
 mqc\_matrix\_integral\_multiply  
     mqc\_est, 71  
     mqc\_est::matmul, 93  
 mqc\_matrix\_inverse  
     mqc\_algebra, 27  
 mqc\_matrix\_isallocated  
     mqc\_algebra, 27  
 mqc\_matrix\_matrix\_at  
     mqc\_algebra, 27  
 mqc\_matrix\_matrix\_contraction  
     mqc\_algebra, 29  
     mqc\_algebra::contraction, 87  
 mqc\_matrix\_matrix\_put

- mqc\_algebra, 29
- mqc\_matrix\_norm
  - mqc\_algebra, 29
- mqc\_matrix\_rms\_max
  - mqc\_algebra, 30
- mqc\_matrix\_rows
  - mqc\_algebra, 30
- mqc\_matrix\_scalar\_at
  - mqc\_algebra, 30
- mqc\_matrix\_scalar\_put
  - mqc\_algebra, 30
- mqc\_matrix\_set
  - mqc\_algebra, 30
- mqc\_matrix\_spinblockghf
  - mqc\_est, 71
- mqc\_matrix\_sqrt
  - mqc\_algebra, 31
- mqc\_matrix\_storage\_type
  - mqc\_algebra, 31
- mqc\_matrix\_svd
  - mqc\_algebra, 31
- mqc\_matrix\_symm2diag
  - mqc\_algebra, 31
- mqc\_matrix\_symm2full
  - mqc\_algebra, 31
- mqc\_matrix\_symm2full\_func
  - mqc\_algebra, 32
- mqc\_matrix\_symmetrize
  - mqc\_algebra, 32
- mqc\_matrix\_symmmatrix\_put\_complex
  - mqc\_algebra, 32
  - mqc\_algebra::mqc\_matrix\_symmmatrix\_put, 105
- mqc\_matrix\_symmmatrix\_put\_integer
  - mqc\_algebra, 32
  - mqc\_algebra::mqc\_matrix\_symmmatrix\_put, 105
- mqc\_matrix\_symmmatrix\_put\_real
  - mqc\_algebra, 32
  - mqc\_algebra::mqc\_matrix\_symmmatrix\_put, 105
- mqc\_matrix\_symmsymmr4tensor\_put\_complex
  - mqc\_algebra, 32
- mqc\_matrix\_symmsymmr4tensor\_put\_real
  - mqc\_algebra, 33
- mqc\_matrix\_test\_diagonal
  - mqc\_algebra, 33
- mqc\_matrix\_test\_symmetric
  - mqc\_algebra, 33
- mqc\_matrix\_trace
  - mqc\_algebra, 33
- mqc\_matrix\_transpose
  - mqc\_algebra, 33
  - mqc\_algebra::transpose, 151
- mqc\_matrix\_undospinblockghf\_eigenvalues
  - mqc\_est, 71
  - mqc\_est::mqc\_matrix\_undospinblockghf, 106
- mqc\_matrix\_undospinblockghf\_integral
  - mqc\_est, 71
  - mqc\_est::mqc\_matrix\_undospinblockghf, 106
- mqc\_matrix\_vector\_at
  - mqc\_algebra, 33
- mqc\_matrix\_vector\_put
  - mqc\_algebra, 34
- mqc\_matrixmatrixdotproduct
  - mqc\_algebra, 34
  - mqc\_algebra::matmul, 91
  - mqc\_algebra::operator(.dot.), 138
- mqc\_matrixmatrixproduct
  - mqc\_algebra, 34
  - mqc\_algebra::operator(\*), 128
- mqc\_matrixmatrixsubtract
  - mqc\_algebra, 34
  - mqc\_algebra::operator(-), 136
- mqc\_matrixmatrixsum
  - mqc\_algebra, 34
  - mqc\_algebra::operator(+), 133
- mqc\_matrixscalarproduct
  - mqc\_algebra, 35
  - mqc\_algebra::operator(\*), 128
- mqc\_matrixvectordotproduct
  - mqc\_algebra, 35
  - mqc\_algebra::matmul, 91
  - mqc\_algebra::operator(.dot.), 138
- mqc\_outer
  - mqc\_algebra, 35
  - mqc\_algebra::operator(.outer.), 145
- mqc\_output\_complex\_scalar
  - mqc\_algebra, 35
  - mqc\_algebra::assignment(=), 81
- mqc\_output\_integer\_scalar
  - mqc\_algebra, 35
  - mqc\_algebra::assignment(=), 81
- mqc\_output\_mqcscalar\_scalar
  - mqc\_algebra, 35
  - mqc\_algebra::assignment(=), 81
- mqc\_output\_real\_scalar
  - mqc\_algebra, 36
  - mqc\_algebra::assignment(=), 81
- mqc\_print\_eigenvalues
  - mqc\_est, 71
  - mqc\_est::mqc\_print, 108
- mqc\_print\_integral
  - mqc\_est, 72
  - mqc\_est::mqc\_print, 108
- mqc\_print\_matrix\_algebra1
  - mqc\_algebra, 36
  - mqc\_algebra::mqc\_print, 106
- mqc\_print\_r4tensor\_algebra1
  - mqc\_algebra, 36
  - mqc\_algebra::mqc\_print, 107

mqc\_print\_scalar\_algebra1  
     mqc\_algebra, 36  
     mqc\_algebra::mqc\_print, 107  
 mqc\_print\_twoeris  
     mqc\_est, 72  
     mqc\_est::mqc\_print, 108  
 mqc\_print\_vector\_algebra1  
     mqc\_algebra, 36  
     mqc\_algebra::mqc\_print, 107  
 mqc\_print\_wavefunction  
     mqc\_est, 72  
     mqc\_est::mqc\_print, 108  
 mqc\_r4tensor\_at  
     mqc\_algebra, 37  
 mqc\_r4tensor\_havecomplex  
     mqc\_algebra, 37  
 mqc\_r4tensor\_haveinteger  
     mqc\_algebra, 37  
 mqc\_r4tensor\_havereal  
     mqc\_algebra, 37  
 mqc\_r4tensor\_initialize  
     mqc\_algebra, 37  
 mqc\_r4tensor\_put  
     mqc\_algebra, 38  
 mqc\_realgtscalar  
     mqc\_algebra, 38  
     mqc\_algebra::operator(.gt.), 141  
 mqc\_reallscalar  
     mqc\_algebra, 38  
     mqc\_algebra::operator(.le.), 143  
 mqc\_realltscalar  
     mqc\_algebra, 38  
     mqc\_algebra::operator(.lt.), 144  
 mqc\_realscalaradd  
     mqc\_algebra, 38  
     mqc\_algebra::operator(+), 134  
 mqc\_realscalardivide  
     mqc\_algebra, 39  
     mqc\_algebra::operator(/), 147  
 mqc\_realscalarmultiply  
     mqc\_algebra, 39  
     mqc\_algebra::operator(\*), 128  
 mqc\_realscalarsubtract  
     mqc\_algebra, 39  
     mqc\_algebra::operator(-), 136  
 mqc\_realvectorproduct  
     mqc\_algebra, 39  
     mqc\_algebra::operator(\*), 129  
 mqc\_scalar\_acos  
     mqc\_algebra, 39  
     mqc\_algebra::acos, 78  
 mqc\_scalar\_asin  
     mqc\_algebra, 39  
     mqc\_algebra::asin, 79  
 mqc\_scalar\_atan  
     mqc\_algebra, 40  
     mqc\_algebra::atan, 85  
 mqc\_scalar\_atan2  
     mqc\_algebra, 40  
     mqc\_algebra::atan2, 86  
 mqc\_scalar\_cmplx  
     mqc\_algebra, 40  
     mqc\_algebra::cmplx, 86  
 mqc\_scalar\_complex\_conjugate  
     mqc\_algebra, 40  
     mqc\_algebra::conjg, 87  
 mqc\_scalar\_complex\_imagpart  
     mqc\_algebra, 40  
     mqc\_algebra::aimag, 78  
 mqc\_scalar\_complex\_realpart  
     mqc\_algebra, 40  
     mqc\_algebra::real, 148  
 mqc\_scalar\_cos  
     mqc\_algebra, 41  
     mqc\_algebra::cos, 88  
 mqc\_scalar\_get\_abs\_value  
     mqc\_algebra, 41  
     mqc\_algebra::abs, 77  
 mqc\_scalar\_get\_intrinsic\_complex  
     mqc\_algebra, 41  
 mqc\_scalar\_get\_intrinsic\_integer  
     mqc\_algebra, 41  
 mqc\_scalar\_get\_intrinsic\_real  
     mqc\_algebra, 41  
 mqc\_scalar\_get\_random\_value  
     mqc\_algebra, 41  
 mqc\_scalar\_havecomplex  
     mqc\_algebra, 42  
 mqc\_scalar\_haveinteger  
     mqc\_algebra, 42  
 mqc\_scalar\_havereal  
     mqc\_algebra, 42  
 mqc\_scalar\_integral\_multiply  
     mqc\_est, 72  
     mqc\_est::operator(\*), 131  
 mqc\_scalar\_isallocated  
     mqc\_algebra, 42  
 mqc\_scalar\_sin  
     mqc\_algebra, 42  
     mqc\_algebra::sin, 149  
 mqc\_scalar\_sqrt  
     mqc\_algebra, 42  
     mqc\_algebra::sqrt, 150  
 mqc\_scalar\_tan  
     mqc\_algebra, 43  
     mqc\_algebra::tan, 150  
 mqc\_scalaradd  
     mqc\_algebra, 43

mqc\_algebra::operator(+), 134  
 mqc\_scalarcomplexadd  
   mqc\_algebra, 43  
   mqc\_algebra::operator(+), 134  
 mqc\_scalarcomplexdivide  
   mqc\_algebra, 43  
   mqc\_algebra::operator(/), 147  
 mqc\_scalarcomplexexponent  
   mqc\_algebra, 43  
   mqc\_algebra::operator(\*\*), 131  
 mqc\_scalarcomplexmultiply  
   mqc\_algebra, 43  
   mqc\_algebra::operator(\*), 129  
 mqc\_scalarcomplexsubtract  
   mqc\_algebra, 44  
   mqc\_algebra::operator(-), 137  
 mqc\_scalardivide  
   mqc\_algebra, 44  
   mqc\_algebra::operator(/), 147  
 mqc\_scalareq  
   mqc\_algebra, 44  
   mqc\_algebra::operator(.eq.), 139  
 mqc\_scalarexponent  
   mqc\_algebra, 44  
   mqc\_algebra::operator(\*\*), 132  
 mqc\_scalarge  
   mqc\_algebra, 44  
   mqc\_algebra::operator(.ge.), 141  
 mqc\_scalargt  
   mqc\_algebra, 44  
   mqc\_algebra::operator(.gt.), 141  
 mqc\_scalargtinteger  
   mqc\_algebra, 45  
   mqc\_algebra::operator(.gt.), 142  
 mqc\_scalargtreal  
   mqc\_algebra, 45  
   mqc\_algebra::operator(.gt.), 142  
 mqc\_scalarintegeradd  
   mqc\_algebra, 45  
   mqc\_algebra::operator(+), 134  
 mqc\_scalarintegerdivide  
   mqc\_algebra, 45  
   mqc\_algebra::operator(/), 147  
 mqc\_scalarintegerexponent  
   mqc\_algebra, 45  
   mqc\_algebra::operator(\*\*), 132  
 mqc\_scalarintegermultiply  
   mqc\_algebra, 45  
   mqc\_algebra::operator(\*), 129  
 mqc\_scalarintegersubtract  
   mqc\_algebra, 46  
   mqc\_algebra::operator(-), 137  
 mqc\_scalarle  
   mqc\_algebra, 46  
   mqc\_algebra::operator(.le.), 143  
 mqc\_scalarleinteger  
   mqc\_algebra, 46  
   mqc\_algebra::operator(.le.), 143  
 mqc\_scalarlereal  
   mqc\_algebra, 46  
   mqc\_algebra::operator(.le.), 143  
 mqc\_scalarlt  
   mqc\_algebra, 46  
   mqc\_algebra::operator(.lt.), 144  
 mqc\_scalarltreal  
   mqc\_algebra, 46  
   mqc\_algebra::operator(.lt.), 144  
 mqc\_scalarmatrixproduct  
   mqc\_algebra, 47  
   mqc\_algebra::operator(\*), 129  
 mqc\_scalarmultiply  
   mqc\_algebra, 47  
   mqc\_algebra::operator(\*), 129  
 mqc\_scalarne  
   mqc\_algebra, 47  
   mqc\_algebra::operator(.ne.), 145  
 mqc\_scalarrealadd  
   mqc\_algebra, 47  
   mqc\_algebra::operator(+), 134  
 mqc\_scalarrealddivide  
   mqc\_algebra, 47  
   mqc\_algebra::operator(/), 147  
 mqc\_scalarrealexponent  
   mqc\_algebra, 47  
   mqc\_algebra::operator(\*\*), 132  
 mqc\_scalarrealmultiply  
   mqc\_algebra, 48  
   mqc\_algebra::operator(\*), 129  
 mqc\_scalarrealsubtract  
   mqc\_algebra, 48  
   mqc\_algebra::operator(-), 137  
 mqc\_scalarsubtract  
   mqc\_algebra, 48  
   mqc\_algebra::operator(-), 137  
 mqc\_scalarvectordifference  
   mqc\_algebra, 48  
   mqc\_algebra::operator(-), 137  
 mqc\_scalarvectorproduct  
   mqc\_algebra, 48  
   mqc\_algebra::operator(\*), 130  
 mqc\_scalarvectorsum  
   mqc\_algebra, 48  
   mqc\_algebra::operator(+), 134  
 mqc\_scf\_eigenvalues\_power  
   mqc\_est, 72  
 mqc\_scf\_integral\_contraction  
   mqc\_est, 73  
   mqc\_est::contraction, 88

mqc\_scf\_integral\_determinant  
     mqc\_est, 73  
 mqc\_scf\_integral\_diagonalize  
     mqc\_est, 73  
 mqc\_scf\_integral\_generalized\_eigensystem  
     mqc\_est, 73  
 mqc\_scf\_integral\_inverse  
     mqc\_est, 73  
 mqc\_scf\_integral\_trace  
     mqc\_est, 74  
 mqc\_scf\_transformation\_matrix  
     mqc\_est, 74  
 mqc\_set\_array2tensor  
     mqc\_algebra, 49  
     mqc\_algebra::assignment(=), 82  
 mqc\_set\_array2vector\_complex  
     mqc\_algebra, 49  
     mqc\_algebra::assignment(=), 82  
     mqc\_algebra::mqc\_set\_array2vector, 117  
 mqc\_set\_array2vector\_integer  
     mqc\_algebra, 49  
     mqc\_algebra::assignment(=), 82  
     mqc\_algebra::mqc\_set\_array2vector, 117  
 mqc\_set\_array2vector\_real  
     mqc\_algebra, 49  
     mqc\_algebra::assignment(=), 82  
     mqc\_algebra::mqc\_set\_array2vector, 117  
 mqc\_set\_complexarray2matrix  
     mqc\_algebra, 49  
     mqc\_algebra::assignment(=), 82  
 mqc\_set\_integerarray2matrix  
     mqc\_algebra, 49  
     mqc\_algebra::assignment(=), 82  
 mqc\_set\_matrix2complexarray  
     mqc\_algebra, 50  
     mqc\_algebra::assignment(=), 83  
 mqc\_set\_matrix2integerarray  
     mqc\_algebra, 50  
     mqc\_algebra::assignment(=), 83  
 mqc\_set\_matrix2matrix  
     mqc\_algebra, 50  
     mqc\_algebra::assignment(=), 83  
 mqc\_set\_matrix2realarray  
     mqc\_algebra, 50  
     mqc\_algebra::assignment(=), 83  
 mqc\_set\_realarray2matrix  
     mqc\_algebra, 50  
     mqc\_algebra::assignment(=), 83  
 mqc\_set\_vector2complexarray  
     mqc\_algebra, 50  
     mqc\_algebra::assignment(=), 83  
 mqc\_set\_vector2integerarray  
     mqc\_algebra, 51  
     mqc\_algebra::assignment(=), 84  
 mqc\_set\_vector2realarray  
     mqc\_algebra, 51  
     mqc\_algebra::assignment(=), 84  
 mqc\_set\_vector2vector  
     mqc\_algebra, 51  
     mqc\_algebra::assignment(=), 84  
 mqc\_twoeris\_allocate  
     mqc\_est, 74  
 mqc\_twoeris\_at  
     mqc\_est, 74  
 mqc\_vector2diagmatrix  
     mqc\_algebra, 51  
 mqc\_vector\_abs  
     mqc\_algebra, 51  
     mqc\_algebra::abs, 77  
 mqc\_vector\_arg sort  
     mqc\_algebra, 51  
 mqc\_vector\_cast\_complex  
     mqc\_algebra, 52  
     mqc\_algebra::mqc\_cast\_complex, 95  
 mqc\_vector\_cast\_real  
     mqc\_algebra, 52  
     mqc\_algebra::mqc\_cast\_real, 95  
 mqc\_vector\_cmplx  
     mqc\_algebra, 52  
     mqc\_algebra::cmplx, 86  
 mqc\_vector\_complex\_imagpart  
     mqc\_algebra, 52  
     mqc\_algebra::aimag, 78  
 mqc\_vector\_complex\_realpart  
     mqc\_algebra, 52  
     mqc\_algebra::real, 149  
 mqc\_vector\_conjugate\_transpose  
     mqc\_algebra, 52  
     mqc\_algebra::dagger, 89  
 mqc\_vector\_copy\_complex2int  
     mqc\_algebra, 53  
 mqc\_vector\_copy\_complex2real  
     mqc\_algebra, 53  
 mqc\_vector\_copy\_int2complex  
     mqc\_algebra, 53  
 mqc\_vector\_copy\_int2real  
     mqc\_algebra, 53  
 mqc\_vector\_copy\_real2complex  
     mqc\_algebra, 53  
 mqc\_vector\_copy\_real2int  
     mqc\_algebra, 53  
 mqc\_vector\_havecomplex  
     mqc\_algebra, 54  
     mqc\_algebra::mqc\_have\_complex, 97  
 mqc\_vector\_haveinteger  
     mqc\_algebra, 54  
     mqc\_algebra::mqc\_have\_int, 98  
 mqc\_vector\_havereal

- mqc\_algebra, 54
- mqc\_algebra::mqc\_have\_real, 99
- mqc\_vector\_initialize
  - mqc\_algebra, 54
- mqc\_vector\_isallocated
  - mqc\_algebra, 54
- mqc\_vector\_iscolumn
  - mqc\_algebra, 54
- mqc\_vector\_maxloc
  - mqc\_algebra, 55
- mqc\_vector\_maxval
  - mqc\_algebra, 55
- mqc\_vector\_minloc
  - mqc\_algebra, 55
- mqc\_vector\_minval
  - mqc\_algebra, 55
- mqc\_vector\_norm
  - mqc\_algebra, 55
- mqc\_vector\_pop
  - mqc\_algebra, 55
- mqc\_vector\_power
  - mqc\_algebra, 56
- mqc\_vector\_push
  - mqc\_algebra, 56
- mqc\_vector\_scalar\_at
  - mqc\_algebra, 56
- mqc\_vector\_scalar\_increment
  - mqc\_algebra, 56
- mqc\_vector\_scalar\_put
  - mqc\_algebra, 56
- mqc\_vector\_shift
  - mqc\_algebra, 56
- mqc\_vector\_sort
  - mqc\_algebra, 57
- mqc\_vector\_sqrt
  - mqc\_algebra, 57
- mqc\_vector\_transpose
  - mqc\_algebra, 57
  - mqc\_algebra::transpose, 151
- mqc\_vector\_unshift
  - mqc\_algebra, 57
- mqc\_vector\_vector\_at
  - mqc\_algebra, 57
- mqc\_vector\_vector\_put
  - mqc\_algebra, 57
- mqc\_vectorcomplexdivide
  - mqc\_algebra, 58
  - mqc\_algebra::operator(/), 147
- mqc\_vectorcomplexproduct
  - mqc\_algebra, 58
  - mqc\_algebra::operator(\*), 130
- mqc\_vectorintegerdivide
  - mqc\_algebra, 58
  - mqc\_algebra::operator(/), 148
- mqc\_vectorintegerproduct
  - mqc\_algebra, 58
  - mqc\_algebra::operator(\*), 130
- mqc\_vectormatrixdotproduct
  - mqc\_algebra, 58
  - mqc\_algebra::matmul, 91
  - mqc\_algebra::operator(.dot.), 138
- mqc\_vectorrealdive
  - mqc\_algebra, 58
  - mqc\_algebra::operator(/), 148
- mqc\_vectorrealproduct
  - mqc\_algebra, 59
  - mqc\_algebra::operator(\*), 130
- mqc\_vectorscalardivide
  - mqc\_algebra, 59
  - mqc\_algebra::operator(/), 148
- mqc\_vectorscalarproduct
  - mqc\_algebra, 59
  - mqc\_algebra::operator(\*), 130
- mqc\_vectorvectordifference
  - mqc\_algebra, 59
  - mqc\_algebra::operator(-), 137
- mqc\_vectorvectordotproduct
  - mqc\_algebra, 59
  - mqc\_algebra::dot\_product, 90
  - mqc\_algebra::operator(.dot.), 138
- mqc\_vectorvectorsum
  - mqc\_algebra, 59
  - mqc\_algebra::operator(+), 135
- multiplicity
  - mqc\_est::mqc\_wavefunction, 125
- nactive
  - mqc\_est::mqc\_pscf\_wavefunction, 109
- nalpha
  - mqc\_est::mqc\_wavefunction, 125
- nalpstr
  - mqc\_est::mqc\_determinant, 96
- nbasis
  - mqc\_est::mqc\_wavefunction, 126
- nbeta
  - mqc\_est::mqc\_wavefunction, 126
- nbetstr
  - mqc\_est::mqc\_determinant, 96
- ncore
  - mqc\_est::mqc\_pscf\_wavefunction, 109
- ndets
  - mqc\_est::mqc\_determinant, 96
- nelectrons
  - mqc\_est::mqc\_wavefunction, 126
- nfrz
  - mqc\_est::mqc\_pscf\_wavefunction, 110
- norm
  - mqc\_algebra::mqc\_matrix, 101

- mqc\_algebra::mqc\_vector, 120
  - mqc\_est::mqc\_scf\_integral, 116
- nval
  - mqc\_est::mqc\_pscf\_wavefunction, 110
- orbitals
  - mqc\_est::mqc\_scf\_integral, 116
- order
  - mqc\_est::mqc\_determinant, 96
- overlap\_matrix
  - mqc\_est::mqc\_wavefunction, 126
- pop
  - mqc\_algebra::mqc\_vector, 120
- power
  - mqc\_algebra::mqc\_vector, 121
  - mqc\_est::mqc\_scf\_eigenvalues, 113
- print
  - mqc\_algebra::mqc\_matrix, 101
  - mqc\_algebra::mqc\_r4tensor, 111
  - mqc\_algebra::mqc\_scalar, 112
  - mqc\_algebra::mqc\_vector, 121
  - mqc\_est::mqc\_scf\_eigenvalues, 113
  - mqc\_est::mqc\_scf\_integral, 116
  - mqc\_est::mqc\_twoeris, 118
  - mqc\_est::mqc\_wavefunction, 124
- pscf\_amplitudes
  - mqc\_est::mqc\_pscf\_wavefunction, 110
- pscf\_energies
  - mqc\_est::mqc\_pscf\_wavefunction, 110
- push
  - mqc\_algebra::mqc\_vector, 121
- put
  - mqc\_algebra::mqc\_matrix, 102
  - mqc\_algebra::mqc\_r4tensor, 111
  - mqc\_algebra::mqc\_vector, 121
- random
  - mqc\_algebra::mqc\_scalar, 112
- rmsmax
  - mqc\_algebra::mqc\_matrix, 102
- rval
  - mqc\_algebra::mqc\_scalar, 112
- s\_type
  - mqc\_algebra::mqc\_matrix, 102
- scf\_density\_matrix
  - mqc\_est::mqc\_wavefunction, 126
- set
  - mqc\_algebra::mqc\_matrix, 102
- setelist
  - mqc\_est::mqc\_scf\_integral, 116
- shift
  - mqc\_algebra::mqc\_vector, 121
- size
  - mqc\_algebra::mqc\_vector, 121
- slater\_condon
  - mqc\_est, 74
- sort
  - mqc\_algebra::mqc\_vector, 121
- sqrt
  - mqc\_algebra::mqc\_matrix, 102
  - mqc\_algebra::mqc\_vector, 122
- src/mqc\_algebra.F03, 153
- src/mqc\_est.F03, 159
- strings
  - mqc\_est::mqc\_determinant, 96
- svd
  - mqc\_algebra::mqc\_matrix, 102
- swap
  - mqc\_est::mqc\_scf\_integral, 116
- symindexhash
  - mqc\_algebra, 60
- symmetry
  - mqc\_est::mqc\_wavefunction, 126
- trace
  - mqc\_algebra::mqc\_matrix, 102
  - mqc\_est::mqc\_scf\_integral, 116
- transpose
  - mqc\_algebra::mqc\_matrix, 103
  - mqc\_algebra::mqc\_vector, 122
- twoeri\_trans
  - mqc\_est, 75
- unshift
  - mqc\_algebra::mqc\_vector, 122
- vat
  - mqc\_algebra::mqc\_matrix, 103
  - mqc\_algebra::mqc\_vector, 122
- vecc
  - mqc\_algebra::mqc\_vector, 123
- veci
  - mqc\_algebra::mqc\_vector, 123
- vecr
  - mqc\_algebra::mqc\_vector, 123
- vput
  - mqc\_algebra::mqc\_matrix, 103
  - mqc\_algebra::mqc\_vector, 122
- wf\_complex
  - mqc\_est::mqc\_wavefunction, 126
- wf\_type
  - mqc\_est::mqc\_wavefunction, 127