MathProject

1.2

Generated by Doxygen 1.8.17

15

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 include/SqEq.hh File Reference	3
2.1.1 Macro Definition Documentation	4
2.1.1.1 SQEQ_ERROR	4
2.1.1.2 MAX_ROOT_NUM	4
2.1.1.3 COEFF_NUM	4
2.1.1.4 EPS	4
2.1.2 Enumeration Type Documentation	4
2.1.2.1 ResType	4
2.1.3 Function Documentation	5
2.1.3.1 read_coeffs()	5
2.1.3.2 print_res()	5
2.1.3.3 solve_sqeq()	6
2.1.3.4 solve_linear()	6
2.1.3.5 is_equal()	7
2.1.3.6 ret_code()	7
2.1.3.7 unit_testing()	8
2.2 SqEq.hh	8
2.3 lib/SqEq.cc File Reference	9
2.3.1 Detailed Description	10
2.3.2 Function Documentation	10
2.3.2.1 read_coeffs()	10
2.3.2.2 solve_sqeq()	11
2.3.2.3 solve_linear()	11
2.3.2.4 is_equal()	12
2.3.2.5 print_res()	12
2.3.2.6 ret_code()	12
2.4 SqEq.cc	13

Index

# **Chapter 1**

# File Index

## 1.1 File List

Here is a list of all files with brief descriptions:

include/SqEq.hh	 3
lib/SqEq.cc	
File with most important SqEq functions	 9

2 File Index

## **Chapter 2**

## **File Documentation**

## 2.1 include/SqEq.hh File Reference

#### **Macros**

```
    #define SQEQ_ERROR 1
        return this from main on error
    #define MAX_ROOT_NUM 2
        maximal munber of roots
    #define COEFF_NUM 3
        number of coefficients
    #define EPS 1e-12
```

### **Enumerations**

epsilon

```
    enum ResType {
        RT_ERROR, RT_INV_COEFF_ERROR, RT_NULLPTR_ERROR, RT_VALID,
        RT_NO_ROOTS, RT_ONE_ROOT, RT_TWO_ROOTS, RT_INF_ROOTS }
        enum with result types of solve functions
```

#### **Functions**

```
    bool read_coeffs (const char *prompt, double coeffs[COEFF_NUM])
    Reading 3 coefficients from stdin.
```

void print\_res (int res\_type, const double results[MAX\_ROOT\_NUM])
 Print squaree equations roots.

int solve\_sqeq (const double coeffs[COEFF\_NUM], double results[MAX\_ROOT\_NUM])
 Solving square equation.

• int solve\_linear (double b, double c, double \*x\_ptr)

Solving linear equation.

• int is\_equal (double n1, double n2)

Compares two double numbers.

int ret\_code (int res\_type)

Generate program return code.

bool unit\_testing ()

Testing SqEq functions.

#### 2.1.1 Macro Definition Documentation

#### 2.1.1.1 SQEQ\_ERROR

```
#define SQEQ_ERROR 1
```

return this from main on error

Definition at line 8 of file SqEq.hh.

#### 2.1.1.2 MAX\_ROOT\_NUM

```
#define MAX_ROOT_NUM 2
```

maximal munber of roots

Definition at line 13 of file SqEq.hh.

#### 2.1.1.3 COEFF\_NUM

#define COEFF\_NUM 3

number of coefficients

Definition at line 18 of file SqEq.hh.

#### 2.1.1.4 EPS

#define EPS 1e-12

epsilon

Definition at line 23 of file SqEq.hh.

## 2.1.2 Enumeration Type Documentation

## 2.1.2.1 **ResType**

enum ResType

enum with result types of solve functions

#### Enumerator

RT_ERROR	
RT_INV_COEFF_ERROR	
RT_NULLPTR_ERROR	
RT_VALID	
RT_NO_ROOTS	
RT_ONE_ROOT	
RT_TWO_ROOTS	
RT_INF_ROOTS	

Definition at line 28 of file SqEq.hh.

## 2.1.3 Function Documentation

## 2.1.3.1 read\_coeffs()

Reading 3 coefficients from stdin.

#### **Parameters**

in	prompt	message to user
out	coeffs	array for coeffs, length must be $>= 3$

#### Returns

true if all OK false on error

Definition at line 13 of file SqEq.cc.

References COEFF\_NUM.

#### 2.1.3.2 print\_res()

Print squaree equations roots.

#### **Parameters**

in	res_type	defines error/number of roots
in	results	roots

Definition at line 92 of file SqEq.cc.

References RT\_ERROR, RT\_INF\_ROOTS, RT\_INV\_COEFF\_ERROR, RT\_NO\_ROOTS, RT\_NULLPTR\_ERROR, RT\_ONE\_ROOT, and RT\_TWO\_ROOTS.

#### 2.1.3.3 solve\_sqeq()

Solving square equation.

#### **Parameters**

in	coeffs	array with coefficients, length must must be $>=3$
out	results	array with roots, length must must be >= 2

#### Returns

result type

Definition at line 20 of file SqEq.cc.

References is\_equal(), RT\_ERROR, RT\_INV\_COEFF\_ERROR, RT\_NO\_ROOTS, RT\_NULLPTR\_ERROR, RT\_ONE\_ROOT, RT\_TWO\_ROOTS, and solve\_linear().

#### 2.1.3.4 solve\_linear()

```
int solve_linear ( \label{eq:condition} \mbox{double } b, \\ \mbox{double } c, \\ \mbox{double } * x\_ptr \mbox{)}
```

Solving linear equation.

#### **Parameters**

in	b	coeff on x
in	С	free member
out	x_ptr	result pointer

Returns

resutl type

Definition at line 71 of file SqEq.cc.

References is\_equal(), RT\_INF\_ROOTS, RT\_NO\_ROOTS, and RT\_ONE\_ROOT.

Referenced by solve\_sqeq().

#### 2.1.3.5 is\_equal()

```
int is_equal ( \label{eq:condition} \mbox{double $n1$,} \\ \mbox{double $n2$ )}
```

Compares two double numbers.

#### **Parameters**

in	n1	1st num
in	n2	2nd num

Returns

int

Definition at line 87 of file SqEq.cc.

References EPS.

Referenced by solve\_linear(), and solve\_sqeq().

## 2.1.3.6 ret\_code()

Generate program return code.

#### **Parameters**

in	res_type	value returned by solve function

#### Returns

int

Definition at line 124 of file SqEq.cc.

References RT\_VALID, and SQEQ\_ERROR.

#### 2.1.3.7 unit\_testing()

```
bool unit_testing ( )
```

Testing SqEq functions.

#### Returns

true if all tests passed false if tests not passed

## 2.2 SqEq.hh

```
00001 /**
00002 * @brief return this from main on error
00003 */
00004
00005 #ifndef SQEQ
00006 #define SQEQ
00007
00008 #define SQEQ_ERROR 1
00009
00011 * @brief maximal munber of roots
00012 */
00013 #define MAX_ROOT_NUM 2
00014
00015 /**
00016 * @brief number of coefficients
00018 #define COEFF_NUM 3
00019
00020 /**
00021 * @brief epsilon
00022 */
00023 #define EPS 1e-12
00024
00025 /**
00026 \,\, & @brief enum with result types of solve functions 00027 \,\, */
00028 enum ResType
00029 {
00030 RT_ERROR, // Errors from here 00031 RT_INV_COEFF_ERROR,
00032 RT_NULLPTR_ERROR,
00033
00034
         RT_VALID, // Valid codes from here
00035
         RT_NO_ROOTS,
00036
          RT_ONE_ROOT,
00037
          RT_TWO_ROOTS,
00038
         RT_INF_ROOTS
00039 };
00040
00041 /**
00041 /**
00042 * @brief Reading 3 coefficients from stdin
00043 *
00044 * @param[in] prompt message to user
00045 * @param[out] coeffs array for coeffs, length must be >= 3
00046 * @return true if all OK
00047 * @return false on error
00048 */
```

```
00049 bool read_coeffs(const char *prompt, double coeffs[COEFF_NUM]);
00051 /**
00052 ^{'}\star @brief Print squaree equations roots 00053 ^{\star}
00054 * @param[in] res_type defines error/number of roots
00055 * @param[in] results roots
00056 */
00057 void print_res(int res_type, const double results[MAX_ROOT_NUM]);
00058
00059 /**
00060 * @brief Solving square equation
00061 *
00062 * @param[in] coeffs array with coefficients, length must must be >= 3
00063 \star @param[out] results array with roots, length must must be >= 2
00064 * @return result type 00065 */
00066 int solve_sqeq(const double coeffs[COEFF_NUM], double results[MAX_ROOT_NUM]);
00067
00068 /**
00069 * @brief Solving linear equation 00070 *
00071 * @param[in] b coeff on x
00072 * @param[in] c free member
00073 * @param[out] x_ptr result pointer
00074 * @return result type
00075 */
00076 int solve_linear(double b, double c, double *x_ptr);
00077
00078 /**
00079 * @brief Compares two double numbers
00080 *
00081 * @param[in] n1 1st num
00082 * @param[in] n2 2nd num
00083 * @return int
00084 */
00085 int is_equal(double n1, double n2);
00087 /**
00088 * @brief Generate program return code
00090 \star @param[in] res_type value returned by solve function 00091 \star @return int 00092 \star/
00093 int ret_code(int res_type);
00094
00095 /**
00096 * @brief Testing SqEq functions 00097 *
00098 * @return true if all tests passed
00099 * @return false if tests not passed
00100 */
00101 bool unit_testing();
00102
00103
00104 #endif // SQEQ
```

## 2.3 lib/SqEq.cc File Reference

File with most important SqEq functions.

```
#include <assert.h>
#include <math.h>
#include <stdio.h>
#include "SqEq.hh"
```

#### **Functions**

- bool read\_coeffs (const char \*prompt, double coeffs[COEFF\_NUM])
   Reading 3 coefficients from stdin.
- int solve\_sqeq (const double coeffs[COEFF\_NUM], double results[MAX\_ROOT\_NUM])

Solving square equation.

• int solve\_linear (double b, double c, double \*x\_ptr)

Solving linear equation.

• int is\_equal (double n1, double n2)

Compares two double numbers.

void print\_res (int res\_type, const double results[MAX\_ROOT\_NUM])

Print squaree equations roots.

• int ret\_code (int res\_type)

Generate program return code.

## 2.3.1 Detailed Description

File with most important SqEq functions.

**Author** 

Tako

Definition in file SqEq.cc.

#### 2.3.2 Function Documentation

### 2.3.2.1 read\_coeffs()

Reading 3 coefficients from stdin.

#### **Parameters**

in	prompt	message to user
out	coeffs	array for coeffs, length must be $>= 3$

#### Returns

true if all OK

false on error

Definition at line 13 of file SqEq.cc.

References COEFF\_NUM.

#### 2.3.2.2 solve\_sqeq()

Solving square equation.

#### **Parameters**

in	coeffs	array with coefficients, length must must be $>=3$
out	results	array with roots, length must must be $>= 2$

#### Returns

result type

Definition at line 20 of file SqEq.cc.

References is\_equal(), RT\_ERROR, RT\_INV\_COEFF\_ERROR, RT\_NO\_ROOTS, RT\_NULLPTR\_ERROR, RT\_ONE\_ROOT, RT\_TWO\_ROOTS, and solve\_linear().

#### 2.3.2.3 solve\_linear()

```
int solve_linear ( \label{eq:condition} \mbox{double } b, \\ \mbox{double } c, \\ \mbox{double } * x\_ptr \; )
```

Solving linear equation.

#### **Parameters**

in	b	coeff on x
in	С	free member
out	x_ptr	result pointer

#### Returns

resutl type

Definition at line 71 of file SqEq.cc.

References is\_equal(), RT\_INF\_ROOTS, RT\_NO\_ROOTS, and RT\_ONE\_ROOT.

Referenced by solve\_sqeq().

#### 2.3.2.4 is\_equal()

```
int is_equal ( \label{eq:condition} \mbox{double $n1$,} \\ \mbox{double $n2$ )}
```

Compares two double numbers.

#### **Parameters**

in	n1	1st num
in	n2	2nd num

#### Returns

int

Definition at line 87 of file SqEq.cc.

References EPS.

Referenced by solve\_linear(), and solve\_sqeq().

#### 2.3.2.5 print\_res()

Print squaree equations roots.

#### **Parameters**

in	res_type	defines error/number of roots
in	results	roots

Definition at line 92 of file SqEq.cc.

References RT\_ERROR, RT\_INF\_ROOTS, RT\_INV\_COEFF\_ERROR, RT\_NO\_ROOTS, RT\_NULLPTR\_ERROR, RT\_ONE\_ROOT, and RT\_TWO\_ROOTS.

#### 2.3.2.6 ret\_code()

```
int ret_code (
          int res_type )
```

Generate program return code.

2.4 SqEq.cc 13

#### **Parameters**

Returns

int

Definition at line 124 of file SqEq.cc.

References RT\_VALID, and SQEQ\_ERROR.

## 2.4 SqEq.cc

```
00001 /**
00002 * @file SqEq.cc
00003 * @author Tako
00004 * @brief File with most important SqEq functions
00005 */
00006
00007 #include <assert.h>
00008 #include <math.h>
00009 #include <stdio.h>
00010
00011 #include "SqEq.hh"
00012
00013 bool read_coeffs(const char *prompt, double coeffs[COEFF_NUM])
00014 {
00015 puts(prompt);
        int scanned = scanf("%lf %lf %lf", coeffs, coeffs + 1, coeffs + 2);
00016
00017 return COEFF_NUM == scanned;
00018 }
00019
00020 int solve_sqeq(const double coeffs[COEFF_NUM], double results[MAX_ROOT_NUM])
00021 {
00022 if ((nullptr == coeffs) || (nullptr == results))
         return RT_NULLPTR_ERROR;
00024
00025
        double a = coeffs[0], b = coeffs[1], c = coeffs[2];
       if (!(isfinite(a) && isfinite(b) && isfinite(c)))
  return RT_INV_COEFF_ERROR;
00026
00027
00028
00029
        if (is equal(a, 0))
00030
          return (solve_linear(b, c, results));
00031
        else if (is_equal(c, 0))
00032
00033
          if (is_equal(b, 0))
00034
          {
00035
            results[0] = 0;
00036
            results[1] = NAN;
00037
            return RT_ONE_ROOT;
00038
00039
          else
00040
          {
00041
           results[0] = 0;
00042
            return (solve_linear(a, b, results + 1) + 1);
00043
00044
00045
00046
        else
00047
          double d = b * b - 4 * a * c;
00048
00049
00050
          if (is_equal(d, 0))
00051
            results[0] = -b / (2 * a);
results[1] = NAN;
00052
00053
00054
            return RT_ONE_ROOT;
00055
00056
          else if (d < 0)
00057
00058
            return RT_NO_ROOTS;
00059
00060
          else if (d > 0)
00061
```

```
double sqrt_d = sqrt(d);
            results[0] = (-b + sqrt_d) / (2 * a);
results[1] = (-b - sqrt_d) / (2 * a);
00063
00064
            return RT_TWO_ROOTS;
00065
00066
          }
00067
00068
        return RT_ERROR;
00069 }
00070
00071 int solve_linear(double b, double c, double *x_ptr)
00072 {
00073
        if (is_equal(b, 0))
          return is_equal(c, 0) ? RT_INF_ROOTS : RT_NO_ROOTS;
00074
00075
        else if (is_equal(c, 0))
00076
        {
        *x_ptr = 0;
return RT_ONE_ROOT;
00077
00078
00079
08000
        else
00081
        {
00082
        *x_ptr = -c / b;
          return RT_ONE_ROOT;
00083
00084
00085 }
00086
00087 int is_equal(double n1, double n2)
} 88000
00089
        return (fabs(n1 - n2) < EPS);</pre>
00090 }
00091
00092 void print_res(int res_type, const double results[MAX_ROOT_NUM])
00093 {
00094
       printf("Answer:\n");
00095
        switch (res_type)
00096
        case RT_NO_ROOTS:
00097
        printf("No roots\n");
break;
00098
00100
        case RT_ONE_ROOT:
        printf("x = %lg\n", results[0]);
00101
00102
          break;
        case RT_TWO_ROOTS:
00103
        printf("x1 = %lg\nx2 = %lg\n", results[0], results[1]);
00104
00105
          break;
00106
        case RT_INF_ROOTS:
        printf("Infinite number of roots\n");
  break;
case RT_INV_COEFF_ERROR:
00107
00108
00109
        printf("INVALID COEFFICIENT VALUE\n");
00110
00111
          break;
00112
        case RT_NULLPTR_ERROR:
        printf("COEFF OR ROOT ARRAY IS NULLPTR\n");
00113
00114
          break;
        case RT_ERROR:
00115
        printf("ERROR\n");
00116
00117
          break;
00118
        default:
        printf("UNKNOWN ERROR\n");
break;
00119
00120
        }
00121
00122 }
00123
00124 int ret_code(int res_type)
00125 {
00126
        return (res_type >= RT_VALID) ? 0 : SQEQ_ERROR;
00127 }
```

# Index

SqEq.hh, 4       is_equal, 11         EPS       sqEq.hh, 4         SqEq.hh, 4       red_coeffs, 10         include/SqEq.hh, 3, 8       solve_linear, 11         sqEq.cc, 11       SqEq.hh         SqEq.cc, 11       COEFF_NUM, 4         SqEq.hh, 7       EPS, 4         lib/SqEq.cc, 9, 13       Iib_COEFF_NUM, 4         MAX_ROOT_NUM       print_res, 5         sqEq.hh, 4       ret_code, 7         print_res       red_coeffs, 5         sqEq.hh, 5       RT_ERROR, 5         sqEq.cc, 12       RT_ERROR, 5         sqEq.cc, 10       RT_INP_ROOTS, 5         sqEq.hh, 5       RT_NO_ROOTS, 5         RT_WO_ROOTS, 5       RT_WO_ROOTS, 5         RT_WO_ROOTS, 5       RT_WO_ROOTS, 5         RT_INP_ROOTS       solve_linear, 6         sqEq.hh, 5       RT_WALID, 5         RT_INP_ROOTS       sqEq.hh, 4         ret_code       solve_sqeq, 6         sqEq.hh, 5       RT_NO_ROOTS, 5         RT_INP_ROOTS       sqEq.hh, 6         sqEq.hh, 5       RT_NO_ROOTS         sqEq.hh, 5       RT_NO_ROOTS         sqEq.hh, 5       RT_NO_ROOTS         sqEq.hh, 5       RT_NO_ROOTS         sqEq.hh, 5 <th>COEFF NUM</th> <th>SqEq.cc</th>	COEFF NUM	SqEq.cc
EPS         print_res, 12           SqEq.hh, 4         red_coeffs, 10           include/SqEq.hh, 3, 8         solve_linear, 11           is_equal         SqEq.hh           SqEq.cc, 11         COEFF_NUM, 4           SqEq.hh, 7         EPS, 4           lib/SqEq.cc, 9, 13         is_equal, 7           MAX_ROOT_NUM         print_res, 5           SqEq.hh, 4         read_coeffs, 5           PesType, 4         ret_code, 7           SqEq.cc, 12         RT_ERROR, 5           SqEq.hh, 5         RT_INY_COEFF_ERROR, 5           RESType, 4         ret_code, 7           SqEq.cc, 10         RT_INY_COEFF_ERROR, 5           RT_INY_COEFF_ERROR, 5         RT_NO_ROOTS, 5           SqEq.hh, 5         RT_NO_ROOTS, 5           RT_YALID, 5         Solve_siqe, 6           SqEq.hh, 6         SOLEQ_ERROR, 4           Writ_Lesting, 8         SQEQ_ERROR           RT_INY_COEFF_ERROR         SqEq.hh, 6           RT_INY_COEFF_ERROR         SqEq.hh, 6           SqEq.hh, 5         RT_NO_ROOTS           SqEq.hh, 5         RT_NO_ROOTS           SqEq.hh, 5         RT_NO_ROOTS           SqEq.hh, 5         RT_NO_ROOTS           SqEq.hh, 5         RT_NO_ROOTS </td <td><del>_</del></td> <td></td>	<del>_</del>	
EPS		
include/SqEq.hh, 3, 8 is_equal	EPS	• —
include/SqEq.hh, 3, 8 is_equal	SqEq.hh, 4	
include/SqEq.hh, 3, 8 is_equal SqEq.cc, 11 SqEq.hh, 7  lib/SqEq.cc, 9, 13  MAX_ROOT_NUM SqEq.hh, 4  print_res SqEq.cc, 12 SqEq.hh, 5  read_coeffs SqEq.cc, 10 SqEq.hh, 5  ResType SqEq.ch, 14  ret_code SqEq.cc, 12 SqEq.hh, 5  RT_NO_ROOTS, 5  RT_VALID, 5  RT_INF_ROOTS SqEq.hh, 6  Solve_Sqeq, 6  SQEQ_ERROR SqEq.hh, 5  RT_INV_COEFF_ERROR SqEq.hh, 5  RT_NO_ROOTS SqEq.hh, 5  RT_NO_ROOTS SqEq.hh, 5  RT_NULLPTR_ERROR SqEq.hh, 5  RT_NO_ROOTS SqEq.hh, 6  Solve_Sqeq SqEq.cc, 10		
is_equal         SqEq.hh           SqEq.cc, 11         COEFF_NUM, 4           EPS, 4         is_equal, 7           MAX_ROOT_NUM         Print_res, 5           SqEq.hh, 4         red_coeffs, 5           Print_res         ResType, 4           SqEq.cc, 12         RT_INF_ROOTS, 5           SqEq.cc, 16         RT_INF_ROOTS, 5           SqEq.hh, 5         RT_NO_ROOTS, 5           ResType         RT_NO_ROOTS, 5           SqEq.hh, 5         RT_NO_ROOTS, 5           RT_ERROR         RT_TWO_ROOTS, 5           SqEq.hh, 7         RT_ERROR           SqEq.hh, 5         RT_NO_ROOTS, 5           RT_INF_ROOTS         SQEQ_ERROR, 4           SqEq.hh, 5         SQEQ_ERROR           RT_INF_ROOTS         SqEq.hh, 4           SqEq.hh, 5         SQEQ_ERROR           RT_NO_ROOTS         SqEq.hh, 5           RT_NO_ROOTS         SqEq.hh, 5           RT_WO_ROOTS         Sq		
SqEq.cc, 11 SqEq.hh, 7  iib/SqEq.cc, 9, 13  MAX_ROOT_NUM SqEq.hh, 4  print_res, 5		SqEq.hh
iib/SqEq.cc, 9, 13		COEFF_NUM, 4
MAX_ROOT_NUM	SqEq.hh, 7	EPS, 4
MAX_ROOT_NUM	lib/SaEa oo 0 12	is_equal, 7
SqEq.hh, 4   read_coeffs, 5   ResType, 4	110/5qEq.cc, 9, 13	MAX_ROOT_NUM, 4
SqEq.hh, 4       read_coeffs, 5         print_res       ret_code, 7         SqEq.cc, 12       RT_ERROR, 5         SqEq.hh, 5       RT_INF_ROOTS, 5         read_coeffs       RT_INO_ROOTS, 5         SqEq.cc, 10       RT_NULLPTR_ERROR, 5         SqEq.hh, 5       RT_ONE_ROOT, 5         ResType       RT_TWO_ROOTS, 5         SqEq.hh, 4       RT_TWO_ROOTS, 5         SqEq.hh, 7       RT_ERROR         SqEq.hh, 5       SQEQ_ERROR, 4         RT_INF_ROOTS       SqEq.hh, 6         SqEq.hh, 5       SQEQ_ERROR         RT_INV_COEFF_ERROR       SqEq.hh, 4         SqEq.hh, 5       SQEQ_ERROR         RT_NO_ROOTS       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 8         RT_TWO_ROOTS       SqEq.hh, 5         RT_WO_ROOTS       SqEq.hh, 5         RT_WALID       SqEq.hh, 6         Solve_linear       SqEq.cc, 11         SqEq.cc, 10       SqEq.cc, 10	MAX ROOT NUM	print_res, 5
ResType, 4   ret_code, 7   SqEq.cc, 12   SqEq.ch, 5   RT_ERROR, 5   RT_INF_ROOTS, 5   RT_INV_COEFF_ERROR, 5   RT_INV_COEFF_ERROR, 5   RT_NO_ROOTS, 5   RT_VALID, 5   RT_VALID, 5   Solve_linear, 6   Solve_sqeq, 6   SqEq.ch, 7   SqEq.hh, 5   RT_NO_ROOTS   SqEq.hh, 5   RT_VALID   SqEq.hh, 5   SOlve_linear   SqEq.ch, 5   Solve_linear   SqEq.ch, 6   Solve_sqeq   SqEq.cc, 10   SqEq.hh, 6   Solve_sqeq   SqEq.cc, 10		read_coeffs, 5
SqEq.cc, 12       RT_ERROR, 5         SqEq.hh, 5       RT_INV_COEFF_ERROR, 5         read_coeffs       RT_NO_ROOTS, 5         SqEq.cc, 10       RT_NO_LENDT, 5         SqEq.hh, 5       RT_ONE_ROOT, 5         ResType       RT_TWO_ROOTS, 5         SqEq.hh, 4       RT_WALID, 5         ret_code       solve_linear, 6         SqEq.cc, 12       solve_linear, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SQEQ_ERROR         ST_INV_COEFF_ERROR       unit_testing         SqEq.hh, 5       SqEq.hh, 8         RT_NOLLPTR_ERROR       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 5         RT_TWO_ROOTS       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         RT_VALID       SqEq.c, 11         SqEq.c, 11       SqEq.c, 11         SqEq.c, 11       SqEq.c, 11         SqEq.c, 10       SqEq.c, 10	54 <u>2</u> 4, 1	ResType, 4
SqEq.cc, 12       RT_ERROR, 5         SqEq.hh, 5       RT_INF_ROOTS, 5         read_coeffs       RT_NO_ROOTS, 5         SqEq.cc, 10       RT_NO_ROOTS, 5         SqEq.hh, 5       RT_NULLPTR_ERROR, 5         ResType       RT_ONE_ROOT, 5         SqEq.hh, 4       RT_ONE_ROOT, 5         ret_code       solve_inear, 6         SqEq.cc, 12       solve_sqeq, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_INF_ROOTS       SQEQ_ERROR         SqEq.hh, 5       SQEQ_ERROR         RT_INV_COEFF_ERROR       SqEq.hh, 4         SqEq.hh, 5       SQEQ_ERROR         ST_INV_COEFF_ERROR       sqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 8         RT_NOLLPTR_ERROR       SqEq.hh, 8         SqEq.hh, 5       ST_ONE_ROOT         SqEq.hh, 5       ST_VALID         SqEq.hh, 5       ST_VALID         SqEq.hh, 5       ST_VALID         SqEq.c, 11       SqEq.c, 11         SqEq.cc, 11       SqEq.cc, 11         SqEq.cc, 10       SqEq.cc, 10	print res	ret_code, 7
SqEq.nh, 5       RT_INF_ROOTS, 5         read_coeffs       RT_NO_ROOTS, 5         SqEq.cc, 10       RT_NULLPTR_ERROR, 5         SqEq.nh, 5       RT_ONE_ROOT, 5         ResType       RT_TWO_ROOTS, 5         SqEq.nh, 4       RT_VALID, 5         ret_code       solve_sqeq, 6         SqEq.cc, 12       solve_sqeq, 6         SqEq.nh, 7       SQEQ_ERROR, 4         RT_INF_ROOTS       SQEQ_ERROR         SqEq.nh, 5       SQEQ_ERROR         RT_INV_COEFF_ERROR       SqEq.nh, 4         SqEq.nh, 5       SqEq.nh, 8         RT_NO_ROOTS       SqEq.nh, 8         SqEq.nh, 5       SqEq.nh, 8         RT_NULLPTR_ERROR       SqEq.nh, 8         SqEq.nh, 5       SqEq.nh, 5         RT_ONE_ROOTS       SqEq.nh, 5         SqEq.nh, 5       SqEq.nh, 5         RT_WO_ROOTS       SqEq.nh, 5         SqEq.nh, 5       SqEq.nh, 5         RT_WO_ROOTS       SqEq.nh, 5         SqEq.nh, 5       SqEq.nh, 5         RT_WO_ROOTS       SqEq.nh, 5<		RT_ERROR, 5
read_coeffs	·	RT_INF_ROOTS, 5
SqEq.cc, 10       RT_NULLPTR_ERROR, 5         SqEq.hh, 5       RT_ONE_ROOT, 5         ResType       RT_TWO_ROOTS, 5         SqEq.hh, 4       RT_VALID, 5         ret_code       solve_linear, 6         SqEq.cc, 12       solve_sqeq, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_INF_ROOTS       SqEq.hh, 5         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SqEq.hh, 5         RT_NO_ROOTS       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 5         RT_NULLPTR_ERROR       SqEq.hh, 8         SqEq.hh, 5       ST_ONE_ROOT         SqEq.hh, 5       ST_ONE_ROOT         SqEq.hh, 5       ST_VALID         SqEq.hh, 5       ST_VALID         SqEq.hh, 5       ST_VALID         SqEq.hh, 5       Stolve_linear         SqEq.cc, 11       SqEq.hh, 6         Solve_sqeq       SqEq.cc, 10		RT_INV_COEFF_ERROR, 5
SqEq.hh, 5       RT_ONE_ROOT, 5         ResType       RT_TWO_ROOTS, 5         SqEq.hh, 4       RT_VALID, 5         ret_code       solve_linear, 6         SqEq.cc, 12       solve_sqeq, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_ERROR       unit_testing, 8         SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 5         RT_ONE_ROOT       SqEq.hh, 5         RT_TWO_ROOTS       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         SqEq.hh, 5       ST_VALID         SqEq.cc, 11       SqEq.hh, 6         Solve_linear       SqEq.cc, 11         SqEq.hh, 6       Solve_sqeq         SqEq.cc, 10       SqEq.cc, 10	read_coeffs	RT_NO_ROOTS, 5
ResType       RT_TWO_ROOTS, 5         SqEq.hh, 4       RT_VALID, 5         ret_code       solve_linear, 6         SqEq.cc, 12       solve_sqeq, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_ERROR       unit_testing, 8         SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 8         RT_NULLPTR_ERROR       SqEq.hh, 5         RT_ONE_ROOT       SqEq.hh, 5         RT_TWO_ROOTS       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         Solve_linear       SqEq.cc, 11         SqEq.hh, 6       Solve_sqeq         SqEq.cc, 10       SqEq.cc, 10	SqEq.cc, 10	RT_NULLPTR_ERROR, 5
SqEq.hh, 4       RT_VALID, 5         ret_code       solve_linear, 6         SqEq.cc, 12       solve_sqeq, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_ERROR       unit_testing, 8         SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 5         RT_NULLPTR_ERROR       SqEq.hh, 5         SqEq.hh, 5       ST_ONE_ROOT         SqEq.hh, 5       SqEq.hh, 5         RT_TWO_ROOTS       SqEq.hh, 5         SqEq.hh, 5       ST_VALID         SqEq.cc, 11       SqEq.cc, 11         SqEq.hh, 6       Solve_sqeq         SqEq.cc, 10       SqEq.cc, 10	SqEq.hh, 5	RT_ONE_ROOT, 5
ret_code	ResType	RT_TWO_ROOTS, 5
SqEq.cc, 12       solve_sqeq, 6         SqEq.hh, 7       SQEQ_ERROR, 4         RT_ERROR       unit_testing, 8         SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       SqEq.hh, 4         RT_INV_COEFF_ERROR       unit_testing         SqEq.hh, 5       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 8         SqEq.hh, 5       RT_ONE_ROOT         SqEq.hh, 5       RT_TWO_ROOTS         SqEq.hh, 5       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         SqEq.hh, 5       Solve_linear         SqEq.cc, 11       SqEq.hh, 6         Solve_sqeq       SqEq.cc, 10	SqEq.hh, 4	RT_VALID, 5
SqEq.hh, 7       SQEQ_ERROR, 4         RT_ERROR       unit_testing, 8         SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       unit_testing         RT_INV_COEFF_ERROR       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 5         RT_ONE_ROOT       SqEq.hh, 5         RT_TWO_ROOTS       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         SqEq.hh, 5       Solve_linear         SqEq.cc, 11       SqEq.hh, 6         Solve_sqeq       SqEq.cc, 10	ret_code	solve_linear, 6
### RT_ERROR	SqEq.cc, 12	solve_sqeq, 6
SqEq.hh, 5       SQEQ_ERROR         RT_INF_ROOTS       SqEq.hh, 4         SqEq.hh, 5       unit_testing         RT_INV_COEFF_ERROR       SqEq.hh, 8         SqEq.hh, 5       SqEq.hh, 8         RT_NO_ROOTS       SqEq.hh, 5         RT_ONE_ROOT       SqEq.hh, 5         RT_TWO_ROOTS       SqEq.hh, 5         RT_VALID       SqEq.hh, 5         solve_linear       SqEq.cc, 11         SqEq.hh, 6       Solve_sqeq         SqEq.cc, 10       SqEq.cc, 10	SqEq.hh, 7	SQEQ_ERROR, 4
RT_INF_ROOTS	RT_ERROR	unit_testing, 8
SqEq.hh, 5  RT_INV_COEFF_ERROR SqEq.hh, 5  RT_NO_ROOTS SqEq.hh, 5  RT_NULLPTR_ERROR SqEq.hh, 5  RT_ONE_ROOT SqEq.hh, 5  RT_TWO_ROOTS SqEq.hh, 5  RT_TWO_ROOTS SqEq.hh, 5  RT_TWO_ROOTS SqEq.hh, 5  RT_VALID SqEq.hh, 5  Solve_linear SqEq.cc, 11 SqEq.hh, 6  solve_sqeq SqEq.cc, 10	SqEq.hh, 5	SQEQ_ERROR
RT_INV_COEFF_ERROR SqEq.hh, 5  RT_NO_ROOTS SqEq.hh, 5  RT_NULLPTR_ERROR SqEq.hh, 5  RT_ONE_ROOT SqEq.hh, 5  RT_TWO_ROOTS SqEq.hh, 5  RT_TWO_ROOTS SqEq.hh, 5  RT_VALID SqEq.hh, 5  Solve_linear SqEq.cc, 11 SqEq.hh, 6  Solve_sqeq SqEq.cc, 10	RT_INF_ROOTS	SqEq.hh, 4
SqEq.hh, 5  RT_NO_ROOTS     SqEq.hh, 5  RT_NULLPTR_ERROR     SqEq.hh, 5  RT_ONE_ROOT     SqEq.hh, 5  RT_TWO_ROOTS     SqEq.hh, 5  RT_TWO_ROOTS     SqEq.hh, 5  SqEq.hh, 5  Solve_linear     SqEq.cc, 11     SqEq.hh, 6  solve_sqeq     SqEq.cc, 10	SqEq.hh, 5	
RT_NO_ROOTS     SqEq.hh, 5 RT_NULLPTR_ERROR     SqEq.hh, 5 RT_ONE_ROOT     SqEq.hh, 5 RT_TWO_ROOTS     SqEq.hh, 5 RT_VALID     SqEq.hh, 5 Solve_linear     SqEq.cc, 11     SqEq.hh, 6 Solve_sqeq     SqEq.cc, 10	RT_INV_COEFF_ERROR	
SqEq.hh, 5 RT_NULLPTR_ERROR    SqEq.hh, 5 RT_ONE_ROOT    SqEq.hh, 5 RT_TWO_ROOTS    SqEq.hh, 5 RT_VALID    SqEq.hh, 5  solve_linear    SqEq.cc, 11    SqEq.hh, 6  solve_sqeq    SqEq.cc, 10	SqEq.hh, 5	SqEq.hh, 8
RT_NULLPTR_ERROR	RT_NO_ROOTS	
SqEq.hh, 5 RT_ONE_ROOT    SqEq.hh, 5 RT_TWO_ROOTS    SqEq.hh, 5 RT_VALID    SqEq.hh, 5 solve_linear    SqEq.cc, 11    SqEq.hh, 6 solve_sqeq    SqEq.cc, 10		
RT_ONE_ROOT     SqEq.hh, 5 RT_TWO_ROOTS     SqEq.hh, 5 RT_VALID     SqEq.hh, 5  solve_linear     SqEq.cc, 11     SqEq.hh, 6  solve_sqeq     SqEq.cc, 10		
SqEq.hh, 5 RT_TWO_ROOTS     SqEq.hh, 5 RT_VALID     SqEq.hh, 5  solve_linear     SqEq.cc, 11     SqEq.hh, 6  solve_sqeq     SqEq.cc, 10	SqEq.hh, 5	
RT_TWO_ROOTS     SqEq.hh, 5  RT_VALID     SqEq.hh, 5  solve_linear     SqEq.cc, 11     SqEq.hh, 6  solve_sqeq     SqEq.cc, 10		
SqEq.hh, 5 RT_VALID SqEq.hh, 5  solve_linear SqEq.cc, 11 SqEq.hh, 6  solve_sqeq SqEq.cc, 10		
RT_VALID SqEq.hh, 5  solve_linear SqEq.cc, 11 SqEq.hh, 6  solve_sqeq SqEq.cc, 10		
SqEq.hh, 5  solve_linear     SqEq.cc, 11     SqEq.hh, 6  solve_sqeq     SqEq.cc, 10	• •	
solve_linear SqEq.cc, 11 SqEq.hh, 6 solve_sqeq SqEq.cc, 10	<del>_</del>	
SqEq.cc, 11 SqEq.hh, 6 solve_sqeq SqEq.cc, 10	SqEq.hh, 5	
SqEq.cc, 11 SqEq.hh, 6 solve_sqeq SqEq.cc, 10	solve_linear	
SqEq.hh, 6 solve_sqeq SqEq.cc, 10		
solve_sqeq SqEq.cc, 10	·	
SqEq.cc, 10	• •	