QCOS

Generated by Doxygen 1.9.1

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 QCOSCscMatrix Struct Reference

Compressed sparse column format matrices.

```
#include <structs.h>
```

Public Attributes

- QCOSInt m
- QCOSInt n
- QCOSInt nnz
- QCOSInt ∗ i
- QCOSInt ∗ p
- QCOSFloat ∗ x

3.1.1 Detailed Description

Compressed sparse column format matrices.

3.1.2 Member Data Documentation

3.1.2.1 i

```
QCOSInt* QCOSCscMatrix::i
```

Row indices (length: nnz).

3.1.2.2 m

```
QCOSInt QCOSCscMatrix::m
```

Number of rows.

3.1.2.3 n

```
QCOSInt QCOSCscMatrix::n
```

Number of columns.

3.1.2.4 nnz

```
QCOSInt QCOSCscMatrix::nnz
```

Number of nonzero elements.

3.1.2.5 p

```
QCOSInt* QCOSCscMatrix::p
```

Column pointers (length: n+1).

3.1.2.6 x

```
QCOSFloat* QCOSCscMatrix::x
```

Data (length: nnz).

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

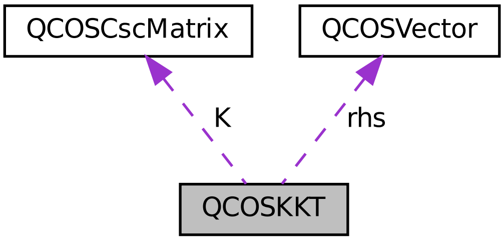
- include/structs.h

3.2 QCOSKKT Struct Reference

Contains all data needed for constructing and modifying KKT matrix and performing predictor-corrector step.

```
#include <structs.h>
```

Collaboration diagram for QCOSKKT:



Public Attributes

- QCOSCscMatrix ∗ K
- QCOSVector ∗ rhs
- QCOSInt ∗ nt2kkt

3.2.1 Detailed Description

Contains all data needed for constructing and modifying KKT matrix and performing predictor-corrector step.

3.2.2 Member Data Documentation

3.2.2.1 K

```
QCOSCscMatrix* QCOSKKT::K
```

KKT matrix in CSC form.

3.2.2.2 nt2kkt

```
QCOSInt* QCOSKKT::nt2kkt
```

Mapping from elements in the Nesterov-Todd scaling matrix to elements in the KKT matrix.

3.2.2.3 rhs

```
QCOSVector* QCOSKKT::rhs
```

Temporary variable for rhs of KKT system.

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

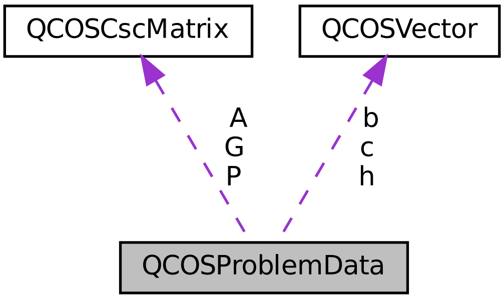
- include/structs.h

3.3 QCOSProblemData Struct Reference

SOCP problem data.

```
#include <structs.h>
```

Collaboration diagram for QCOSProblemData:



Public Attributes

- QCOSCscMatrix ∗ P
- QCOSVector ∗ c
- QCOSCscMatrix ∗ A
- QCOSVector ∗ b
- QCOSCscMatrix ∗ G
- QCOSVector ∗ h
- QCOSInt l
- QCOSInt ncones
- QCOSInt ∗ q
- QCOSInt n
- QCOSInt m
- QCOSInt p

3.3.1 Detailed Description

SOCP problem data.

3.3.2 Member Data Documentation

3.3.2.1 A

```
QCOSCscMatrix* QCOSProblemData::A
```

Affine equality constraint matrix.

3.3.2.2 b

```
QCOSVector* QCOSProblemData::b
```

Affine equality constraint offset.

3.3.2.3 c

```
QCOSVector* QCOSProblemData::c
```

Linear cost term.

3.3.2.4 G

```
QCOSCscMatrix* QCOSProblemData::G
```

Conic constraint matrix.

3.3.2.5 h

```
QCOSVector* QCOSProblemData::h
```

Conic constraint offset.

3.3.2.6 l

```
QCOSInt QCOSProblemData::l
```

Dimension of non-negative orthant in cone C.

3.3.2.7 m

```
QCOSInt QCOSProblemData::m
```

Number of conic constraints.

3.3.2.8 n

```
QCOSInt QCOSProblemData::n
```

Number of primal variables.

3.3.2.9 ncones

```
QCOSInt QCOSProblemData::ncones
```

Number of second-order cones in C

3.3.2.10 P

```
QCOSCscMatrix* QCOSProblemData::P
```

Quadratic cost term.

3.3.2.11 p

```
QCOSInt QCOSProblemData::p
```

Number of affine equality constraints.

3.3.2.12 q

```
QCOSInt* QCOSProblemData::q
```

Dimension of each second-order cone (length of ncones)

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

- include/structs.h

3.4 QCOSSettings Struct Reference

QCOS solver settings.

```
#include <structs.h>
```

Public Attributes

- QCOSFloat **tol**
- unsigned char **verbose**

3.4.1 Detailed Description

QCOS solver settings.

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

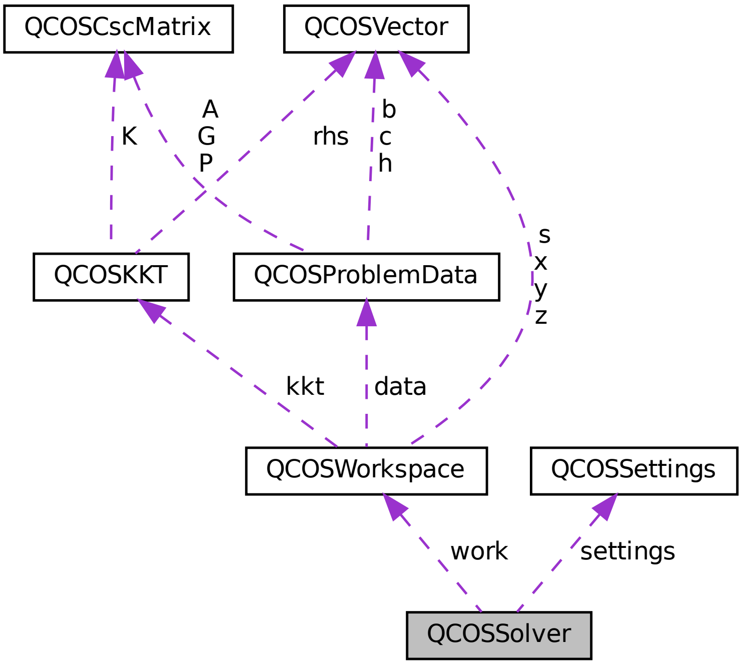
- include/structs.h

3.5 QCOSSolver Struct Reference

QCOS Solver struct. Contains all information about the state of the solver.

```
#include <structs.h>
```

Collaboration diagram for QCOSSolver:



Public Attributes

- QCOSSettings ∗ settings
- QCOSWorkspace ∗ work

3.5.1 Detailed Description

QCOS Solver struct. Contains all information about the state of the solver.

3.5.2 Member Data Documentation

3.5.2.1 settings

```
QCOSSettings* QCOSSolver::settings
```

Solver settings.

3.5.2.2 work

```
QCOSWorkspace* QCOSSolver::work
```

Solver workspace.

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

- include/structs.h

3.6 QCOSVector Struct Reference

Internal QCOS vector.

```
#include <structs.h>
```

Public Attributes

- QCOSFloat ∗ x
- QCOSInt n

3.6.1 Detailed Description

Internal QCOS vector.

3.6.2 Member Data Documentation

3.6.2.1 n

```
QCOSInt QCOSVector::n
```

Length of vector.

3.6.2.2 x

```
QCOSFloat* QCOSVector::x
```

Data.

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

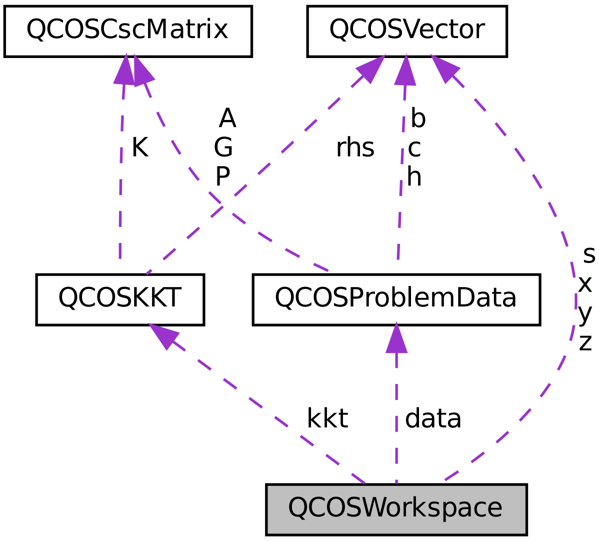
- include/structs.h

3.7 QCOSWorkspace Struct Reference

QCOS Workspace.

```
#include <structs.h>
```

Collaboration diagram for QCOSWorkspace:



Public Attributes

- QCOSProblemData ∗ data
- QCOSKKT ∗ kkt
- QCOSVector ∗ x
- QCOSVector ∗ s
- QCOSVector ∗ y
- QCOSVector ∗ z

3.7.1 Detailed Description

QCOS Workspace.

3.7.2 Member Data Documentation

3.7.2.1 data

```
QCOSProblemData* QCOSWorkspace::data
```

Contains SOCP problem data.

3.7.2.2 kkt

```
QCOSKKT* QCOSWorkspace::kkt
```

Contains all data related to KKT system.

3.7.2.3 s

```
QCOSVector* QCOSWorkspace::s
```

Iterate of slack variables associated with conic constraint.

3.7.2.4 x

```
QCOSVector* QCOSWorkspace::x
```

Iterate of primal variables.

3.7.2.5 y

```
QCOSVector* QCOSWorkspace::y
```

Iterate of dual variables associated with affine equality constraint.

3.7.2.6 z

```
QCOSVector* QCOSWorkspace::z
```

Iterate of dual variables associated with conic constraint.

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

- include/structs.h

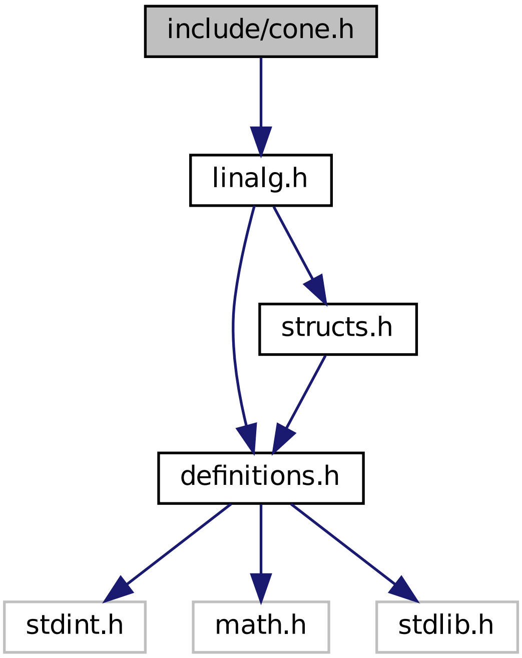
Chapter 4

File Documentation

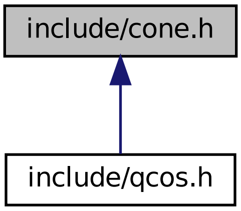
4.1 include/cone.h File Reference

```
#include "linalg.h"
```

Include dependency graph for cone.h:



This graph shows which files directly or indirectly include this file:



Functions

- void cone_product (QCOSFloat ∗u, QCOSFloat ∗v, QCOSFloat ∗p, QCOSProblemData ∗data)

  Computes cone product $u * v = p$ with respect to C.
- void soc_product (QCOSFloat ∗u, QCOSFloat ∗v, QCOSFloat ∗p, QCOSInt n)

  Computes second-order cone product $u * v = p$.
- QCOSFloat soc_residual (QCOSFloat ∗u, QCOSInt n)

  Computes residual of vector u with respect to the second order cone of dimension n.
- QCOSFloat cone_residual (QCOSFloat ∗u, QCOSProblemData ∗data)

  Computes residual of vector u with respect to cone C.
- void bring2cone (QCOSFloat ∗u, QCOSProblemData ∗data)

  Performs $u = u + (1 + a) * e$ where e is the cannonical vector for each cone LP Cone: e = ones(n), second-order cone: e = (1,0,0,...) and a is the minimum scalar value such that $u + (1 + a) * e$ is in cone C.

4.1.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.1.2 LICENSE

4.1.3 DESCRIPTION

Includes various functions necessary for cone operations.

4.1.4 Function Documentation

4.1.4.1 bring2cone()

```
void bring2cone (
             QCOSFloat * u,
             QCOSProblemData * data )
```

Performs u = u + (1 + a) ∗ e where e is the cannonical vector for each cone LP Cone: e = ones(n), second-order cone: e = (1,0,0,...) and a is the minimum scalar value such that u + (1 + a) ∗ e is in cone C.

Parameters

u	Vector to bring to cone.
data	Pointer to problem data.

4.1.4.2 cone_product()

```
void cone_product (
             QCOSFloat * u,
             QCOSFloat * v,
             QCOSFloat * p,
             QCOSProblemData * data )
```

Computes cone product u ∗ v = p with respect to C.

Parameters

u	Input vector.
v	Input vector.
p	Cone product of u and v.
data	Pointer to problem data.

4.1.4.3 cone_residual()

```
QCOSFloat cone_residual (
             QCOSFloat * u,
             QCOSProblemData * data )
```

Computes residual of vector u with respect to cone C.

Parameters

u	Vector to be tested.
data	Pointer to problem data.

Returns

Residual: Negative if the vector is in the cone and positive otherwise.

4.1.4.4 soc_product()

```
void soc_product (
            QCOSFloat * u,
            QCOSFloat * v,
            QCOSFloat * p,
            QCOSInt n )
```

Computes second-order cone product u ∗ v = p.

Parameters

u	Input vector.
v	Input vector.
p	Cone product of u and v.
n	Length of vectors.

4.1.4.5 soc_residual()

```
QCOSFloat soc_residual (
            QCOSFloat * u,
            QCOSInt n )
```

Computes residual of vector u with respect to the second order cone of dimension n.

Parameters

u	u = (u0, u1) is a vector in second order cone of dimension n.
n	Dimension of second order cone.

Returns

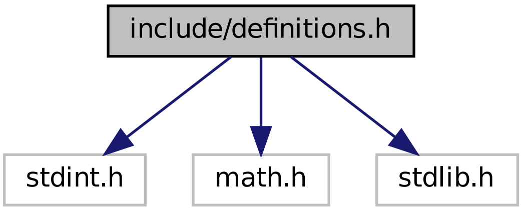
Residual: norm(u1) - u0. Negative if the vector is in the cone and positive otherwise.

4.2 include/definitions.h File Reference

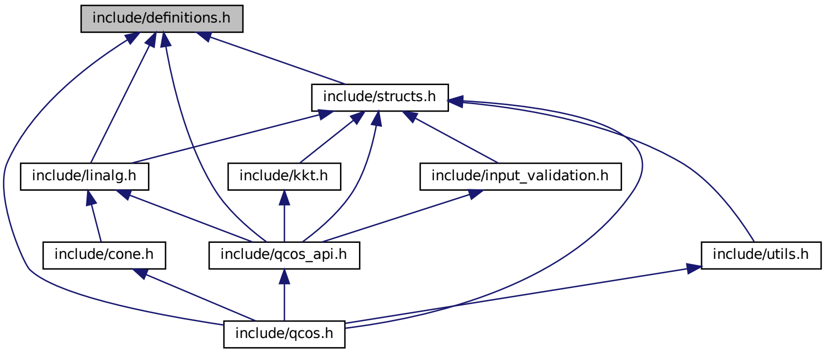
```
#include "stdint.h"
#include "math.h"
```

`#include <stdlib.h>`

Include dependency graph for definitions.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **qcos_max**(a, b) (a > b) ? a : b
- #define **qcos_sqrt**(a) sqrt(a)
- #define **qcos_malloc** malloc
- #define **qcos_calloc** calloc
- #define **qcos_free** free

Typedefs

- typedef int32_t **QCOSInt**
- typedef double **QCOSFloat**

4.2.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.2.2 LICENSE

Copyright (c) 2024, Govind M. Chari This source code is licensed under the BSD 2-Clause License

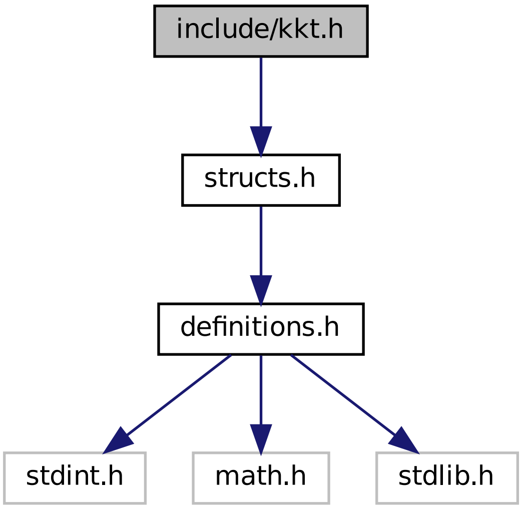
4.2.3 DESCRIPTION

Defines QCOSInt, QCOSFloat, qcos_malloc, qcos_calloc, and qcos_free.

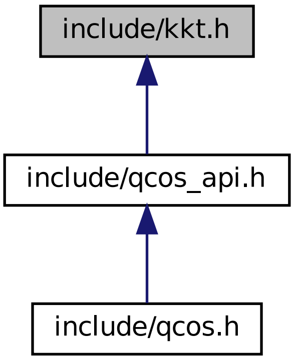
4.3 include/kkt.h File Reference

```
#include "structs.h"
```

Include dependency graph for kkt.h:



This graph shows which files directly or indirectly include this file:



Functions

- QCOSCscMatrix ∗ initialize_kkt (QCOSProblemData ∗data)

    Allocate memory for KKT matrix.
- void construct_kkt (QCOSWorkspace ∗work)

    Constructs upper triangular part of KKT matrix with -I for Nestrov-Todd scaling matrix (the (3,3) block)

4.3.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.3.2 LICENSE

Copyright (c) 2024, Govind M. Chari This source code is licensed under the BSD 2-Clause License

4.3.3 DESCRIPTION

Provides various functions for constructing and updating blocks for the KKT matrix.

4.3.4 Function Documentation

4.3.4.1 construct_kkt()

```
void construct_kkt (
            QCOSWorkspace * work )
```

Constructs upper triangular part of KKT matrix with -I for Nestrov-Todd scaling matrix (the (3,3) block)

clang-format off

```
[ P   A^T   G^T ]
```

K = | A 0 0 | [G 0 -I]

clang-format on

Parameters

work	Pointer to QCOSWorkspace

4.3.4.2 initialize_kkt()

```
QCOSCscMatrix* initialize_kkt (
            QCOSProblemData * data )
```

Allocate memory for KKT matrix.

Parameters

data	Pointer to problem data.

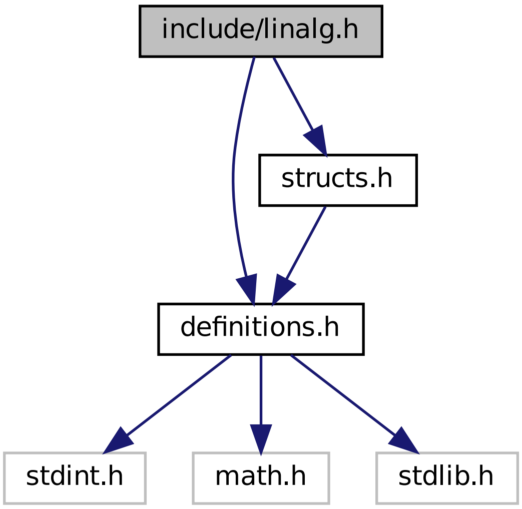
Returns

Pointer to initialized KKT matrix.

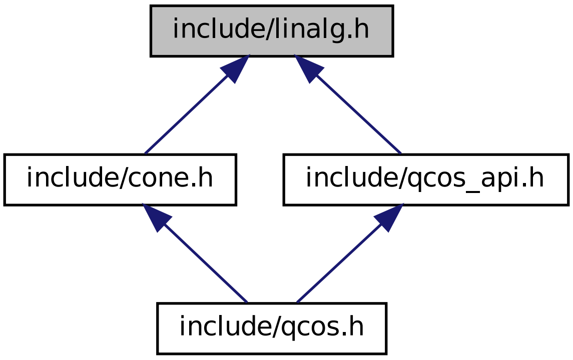
4.4 include/linalg.h File Reference

```
#include "definitions.h"
#include "structs.h"
```

Include dependency graph for linalg.h:



This graph shows which files directly or indirectly include this file:



Functions

- QCOSVector ∗ qcos_vector_calloc (QCOSInt n)

  Allocates a QCOSVector of length n and zeros out the data.
- QCOSCscMatrix ∗ new_qcos_csc_matrix (QCOSCscMatrix ∗A)

  Allocates a new csc matrix and copies A to it.
- QCOSVector ∗ new_qcos_vector_from_array (QCOSFloat ∗x, QCOSInt n)

Constructs a new QCOSVector from raw array.

- void copy_arrayf (const QCOSFloat ∗x, QCOSFloat ∗y, QCOSInt n)

  Copies array of QCOSFloats from x to array y.
- void copy_arrayi (const QCOSInt ∗x, QCOSInt ∗y, QCOSInt n)

  Copies array of QCOSInts from x to array y.
- QCOSFloat dot (QCOSFloat ∗u, QCOSFloat ∗v, QCOSInt n)

  Computes dot product of u and v.

4.4.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.4.2 LICENSE

4.4.3 DESCRIPTION

Provides various linear algebra operations.

4.4.4 Function Documentation

4.4.4.1 copy_arrayf()

```
void copy_arrayf (
             const QCOSFloat * x,
             QCOSFloat * y,
             QCOSInt n )
```

Copies array of QCOSFloats from x to array y.

Parameters

x	Source array.
y	Destination array.
n	Length of arrays.

4.4.4.2 copy_arrayi()

```
void copy_arrayi (
             const QCOSInt * x,
             QCOSInt * y,
             QCOSInt n )
```

Copies array of QCOSInts from x to array y.

Parameters

x	Source array.
y	Destination array.
n	Length of arrays.

4.4.4.3 dot()

```
QCOSFloat dot (
             QCOSFloat * u,
             QCOSFloat * v,
             QCOSInt n )
```

Computes dot product of u and v.

Parameters

u	Input vector.
v	Input vector.
n	Length of vectors.

Returns

Dot product of u and v.

4.4.4.4 new_qcos_csc_matrix()

```
QCOSCscMatrix* new_qcos_csc_matrix (
             QCOSCscMatrix * A )
```

Allocates a new csc matrix and copies A to it.

Parameters

A	Matrix to copy.

Returns

Pointer to new constructed matrix.

4.4.4.5 new_qcos_vector_from_array()

```
QCOSVector* new_qcos_vector_from_array (
            QCOSFloat * x,
            QCOSInt n )
```

Constructs a new QCOSVector from raw array.

Parameters

x	Raw array of data.
n	Length of raw array.

Returns

Pointer to new vector.

4.4.4.6 qcos_vector_calloc()

```
QCOSVector* qcos_vector_calloc (
            QCOSInt n )
```

Allocates a QCOSVector of length n and zeros out the data.

Parameters

n	Length of vector to allocate.

Returns

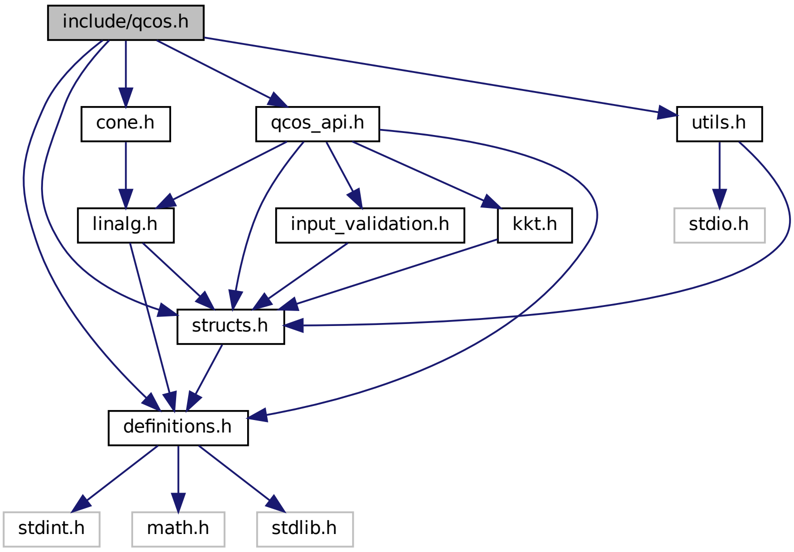
Pointer to allocated QCOSVector.

4.5 include/qcos.h File Reference

```
#include "cone.h"
#include "definitions.h"
#include "qcos_api.h"
#include "structs.h"
```

```
#include "utils.h"
```

Include dependency graph for qcos.h:



4.5.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.5.2 LICENSE

4.5.3 DESCRIPTION

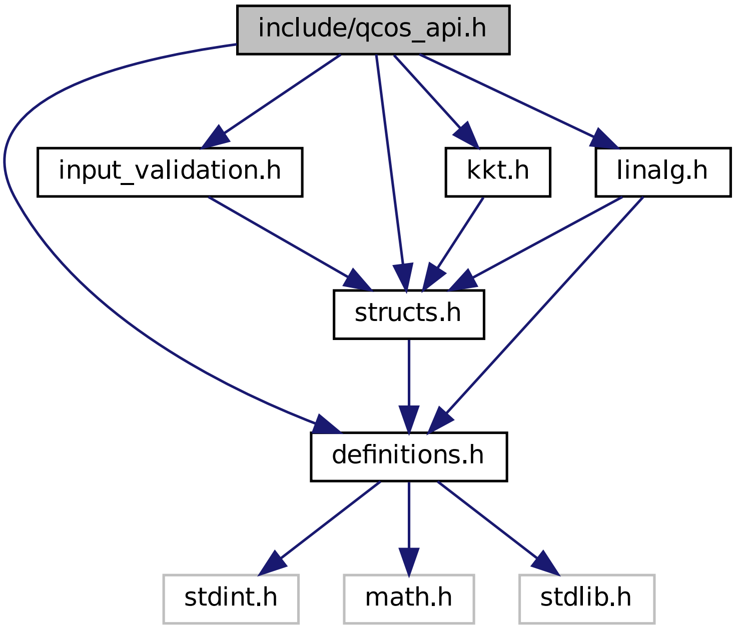
This is the file that should be included when using QCOS.

4.6 include/qcos_api.h File Reference

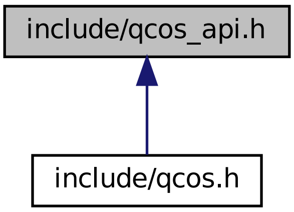
```
#include "definitions.h"
#include "input_validation.h"
#include "kkt.h"
#include "linalg.h"
```

```
#include "structs.h"
```

Include dependency graph for qcos_api.h:



This graph shows which files directly or indirectly include this file:



Functions

- QCOSSolver ∗ qcos_setup (QCOSCscMatrix ∗P, QCOSFloat ∗c, QCOSCscMatrix ∗A, QCOSFloat ∗b, QCOSCscMatrix ∗G, QCOSFloat ∗h, QCOSInt l, QCOSInt ncones, QCOSInt ∗q, QCOSSettings ∗settings)

  Allocates all memory needed for QCOS to solve the SOCP.
- void qcos_set_csc (QCOSCscMatrix ∗A, QCOSInt m, QCOSInt n, QCOSInt Annz, QCOSFloat ∗Ax, QCOSInt ∗Ap, QCOSInt ∗Ai)

  Sets the data for a compressed sparse column matrix.

- void set_default_settings (QCOSSettings ∗settings)

    Set the default settings struct.
- QCOSInt **qcos_solve** ()
- QCOSInt **qcos_cleanup** (QCOSSolver ∗solver)

4.6.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.6.2 LICENSE

4.6.3 DESCRIPTION

Exposes the API for QCOS.

4.6.4 Function Documentation

4.6.4.1 qcos_set_csc()

```
void qcos_set_csc (
            QCOSCscMatrix * A,
            QCOSInt m,
            QCOSInt n,
            QCOSInt Annz,
            QCOSFloat * Ax,
            QCOSInt * Ap,
            QCOSInt * Ai )
```

Sets the data for a compressed sparse column matrix.

Parameters

A	Pointer to the CSC matrix
m	Number of rows in the matrix
n	Number of columns in the matrix
Annz	Number of nonzero elements in the matrix
Ax	Array of data for the matrix
Ap	Array of column pointers for the data
Ai	Array of row indices for data

4.6.4.2 qcos_setup()

```
QCOSSolver* qcos_setup (
             QCOSCscMatrix * P,
             QCOSFloat * c,
             QCOSCscMatrix * A,
             QCOSFloat * b,
             QCOSCscMatrix * G,
             QCOSFloat * h,
             QCOSInt l,
             QCOSInt ncones,
             QCOSInt * q,
             QCOSSettings * settings )
```

Allocates all memory needed for QCOS to solve the SOCP.

Parameters

P	Upper triangular part of quadratic cost Hessian in CSC form
c	Linear cost vector
A	Affine equality constraint matrix in CSC form
b	Affine equality constraint offset vector
G	Conic constraint matrix in CSC form
h	Conic constraint offset vector
l	Dimension of non-negative orthant
ncones	Number of second-order cones
q	Dimension of each second-order cone
settings	Settings struct

Returns

Pointer to solver

4.6.4.3 set_default_settings()

```
void set_default_settings (
             QCOSSettings * settings )
```

Set the default settings struct.

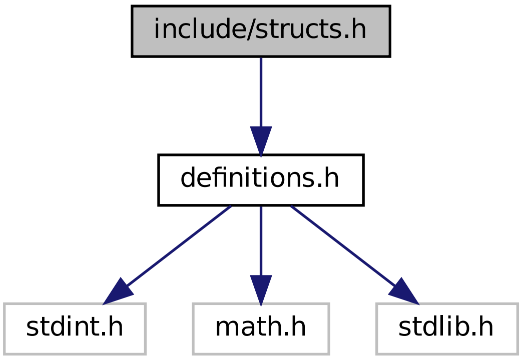
Parameters

settings	Pointer to settings struct

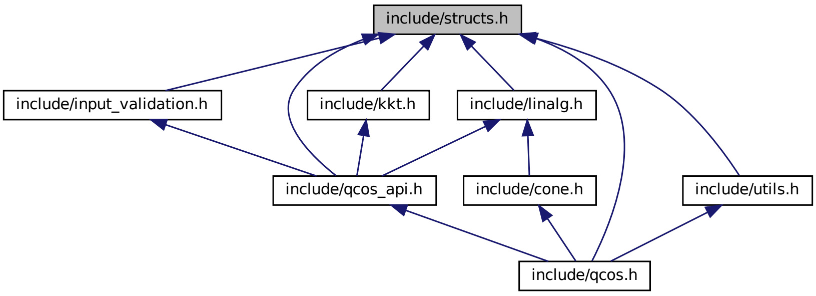
4.7 include/structs.h File Reference

```
#include "definitions.h"
```

Include dependency graph for structs.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct QCOSCscMatrix

  Compressed sparse column format matrices.
- struct QCOSVector

  Internal QCOS vector.
- struct QCOSProblemData

  SOCP problem data.
- struct QCOSSettings

  QCOS solver settings.
- struct QCOSKKT

  Contains all data needed for constructing and modifying KKT matrix and performing predictor-corrector step.

- struct QCOSWorkspace

  QCOS Workspace.
- struct QCOSSolver

  QCOS Solver struct. Contains all information about the state of the solver.

4.7.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.7.2 LICENSE

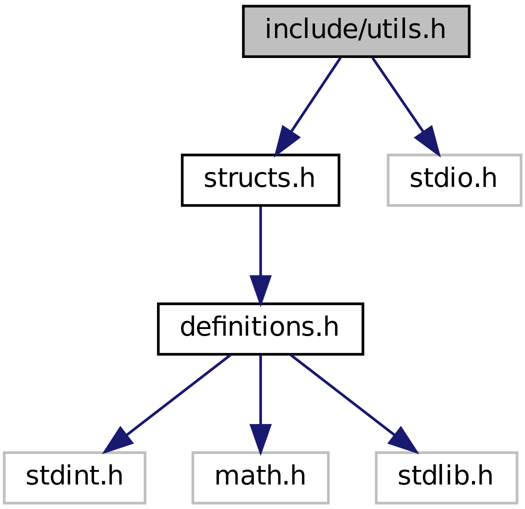
4.7.3 DESCRIPTION

Defines all structs used by QCOS.

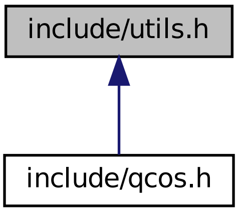
4.8 include/utils.h File Reference

```
#include "structs.h"
#include <stdio.h>
```

Include dependency graph for utils.h:



This graph shows which files directly or indirectly include this file:



Functions

- void print_qcos_csc_matrix (QCOSCscMatrix ∗M)

  Prints QCOSCscMatrix.
- void print_qcos_vector (QCOSVector ∗V)

  Prints QCOSVector.
- void print_header ()

  Prints qcos header.

4.8.1 Detailed Description

Author

Govind M. Chari govindchari1@gmail.com

4.8.2 LICENSE

4.8.3 DESCRIPTION

Provides useful utility functions.

4.8.4 Function Documentation

4.8.4.1 print_qcos_csc_matrix()

```
void print_qcos_csc_matrix (
            QCOSCscMatrix * M )
```

Prints QCOSCscMatrix.

Parameters

M	Pointer to QCOSCscMatrix that will be printed.

4.8.4.2 print_qcos_vector()

```
void print_qcos_vector (
            QCOSVector * V )
```

Prints QCOSVector.

Parameters

V	Pointer to QCOSVector that will be printed.

Index