STAPpp

1.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

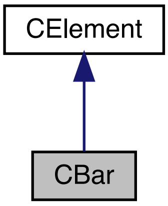
Class Documentation

4.1 CBar Class Reference

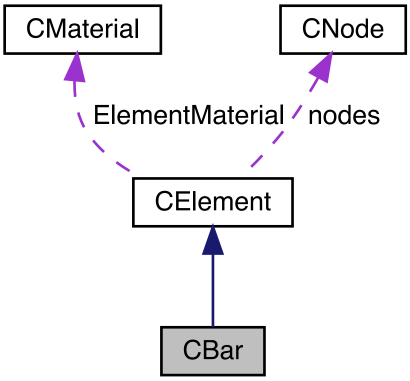
Bar element class.

```
#include <Bar.h>
```

Inheritance diagram for CBar:



Collaboration diagram for CBar:



Public Member Functions

- CBar ()

  Constructor.
- ∼CBar ()

  Desconstructor.
- virtual bool Read (ifstream &Input, unsigned int Ele, CMaterial ∗MaterialSets, CNode ∗NodeList)

  Read element data from stream Input.
- virtual void Write (ofstream &OutputFile, unsigned int Ele)

  Write element data to stream OutputFile.
- virtual void GenerateLocationMatrix ()

  Generate location matrix: the global equation number that corresponding to each DOF of the element.
- virtual void ElementStiffness (double ∗Matrix)

  Calculate element stiffness matrix.
- virtual void ElementStress (double ∗stress, double ∗Displacement)

  Calculate element stress.
- virtual unsigned int SizeOfStiffnessMatrix ()

  Return the size of the element stiffness matrix (stored as an array column by column)

Additional Inherited Members

4.1.1 Detailed Description

Bar element class.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 CBar()

```
CBar::CBar ( )
```

Constructor.

4.1.2.2 ∼CBar()

```
CBar::∼CBar ( )
```

Desconstructor.

4.1.3 Member Function Documentation

4.1.3.1 ElementStiffness()

```
void CBar::ElementStiffness (
             double * Matrix )  [virtual]
```

Calculate element stiffness matrix.

Implements CElement.

4.1.3.2 ElementStress()

```
void CBar::ElementStress (
             double * stress,
             double * Displacement )  [virtual]
```

Calculate element stress.

Implements CElement.

4.1.3.3 GenerateLocationMatrix()

```
void CBar::GenerateLocationMatrix ( )  [virtual]
```

Generate location matrix: the global equation number that corresponding to each DOF of the element.

Implements CElement.

4.1.3.4 Read()

```
bool CBar::Read (
             ifstream & Input,
             unsigned int Ele,
             CMaterial * MaterialSets,
             CNode * NodeList )  [virtual]
```

Read element data from stream Input.

Implements CElement.

4.1.3.5 SizeOfStiffnessMatrix()

```
unsigned int CBar::SizeOfStiffnessMatrix ( )  [virtual]
```

Return the size of the element stiffness matrix (stored as an array column by column)

Implements CElement.

4.1.3.6 Write()

```
void CBar::Write (
             ofstream & OutputFile,
             unsigned int Ele )  [virtual]
```

Write element data to stream OutputFile.

Implements CElement.

The documentation for this class was generated from the following files:

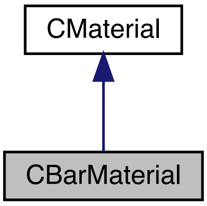
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp

4.2 CBarMaterial Class Reference

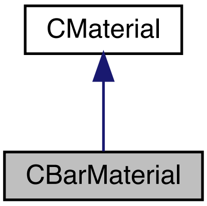
Material class for bar element.

```
#include <Material.h>
```

Inheritance diagram for CBarMaterial:



Collaboration diagram for CBarMaterial:



Public Member Functions

- virtual bool Read (ifstream &Input, unsigned int mset)

  Read material data from stream Input.
- virtual void Write (ofstream &OutputFile, unsigned int mset)

  Write material data to Stream OutputFile.

Public Attributes

- double Area

  Sectional area of a bar element.

4.2.1 Detailed Description

Material class for bar element.

4.2.2 Member Function Documentation

4.2.2.1 Read()

```
bool CBarMaterial::Read (
            ifstream & Input,
            unsigned int mset )  [virtual]
```

Read material data from stream Input.

Implements CMaterial.

4.2.2.2 Write()

```
void CBarMaterial::Write (
            ofstream & OutputFile,
            unsigned int mset )  [virtual]
```

Write material data to Stream OutputFile.

Implements CMaterial.

4.2.3 Member Data Documentation

4.2.3.1 Area

```
double CBarMaterial::Area
```

Sectional area of a bar element.

The documentation for this class was generated from the following files:

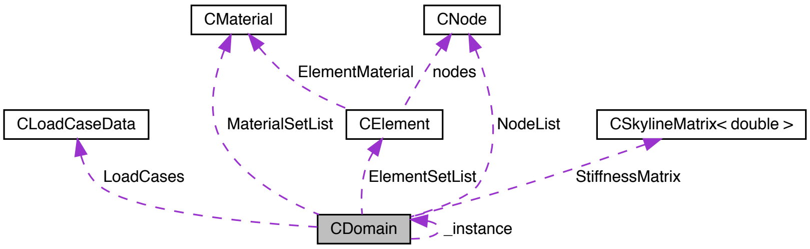
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp

4.3 CDomain Class Reference

Domain class : Define the problem domain.

```
#include <Domain.h>
```

Collaboration diagram for CDomain:



Public Member Functions

- CDomain ()

  Constructor.
- ∼CDomain ()

  Desconstructor.
- bool ReadData (string FileName, string OutFile)

  Read domain data from the input data file.
- bool ReadNodalPoints ()

  Read nodal point data.
- bool ReadLoadCases ()

  Read load case data.
- bool ReadElements ()

  Read element data.
- bool ReadBarElementData (unsigned int EleGrp)

  Read bar element data from the input data file.
- void CalculateEquationNumber ()

  Calculate global equation numbers corresponding to every degree of freedom of each node.
- void CalculateColumnHeights ()

  Calculate column heights.
- void CalculateDiagnoalAddress ()

  Calculate address of diagonal elements in banded matrix.
- void AllocateMatrices ()

  Allocate storage for matrices.
- void AssembleStiffnessMatrix ()

  Assemble the banded gloabl stiffness matrix.
- bool AssembleForce (unsigned int LoadCase)

  Assemble the global nodal force vector for load case LoadCase.
- unsigned int GetMODEX ()

  Return solution mode.

- string GetTitle ()

  Return the title of problem.
- unsigned int GetNEQ ()

  Return the total number of equations.
- unsigned int GetNUMNP ()

  Return the total number of nodal points.
- unsigned int GetNWK ()

  Return the number of banded global stiffness matrix elements.
- unsigned int GetMK ()

  Return the maximum half bandwith.
- CNode $*$ GetNodeList ()

  Return the node list.
- unsigned int $*$ GetNUME ()

  Return the number of elements in each element group.
- unsigned int GetNUMEG ()

  Return total number of element groups.
- unsigned int $*$ GetElementTypes ()

  Element type of each group.
- CElement $**$ GetElementSetList ()

  Return element Set List.
- unsigned int $*$ GetNUMMAT ()

  Return number of different sets of material/section properties in each element group.
- CMaterial $**$ GetMaterialSetList ()

  Return material set list.
- double $*$ GetForce ()

  Return pointer to the global nodal force vector.
- double $*$ GetDisplacement ()

  Return pointer to the global nodal displacement vector.
- unsigned int GetNLCASE ()

  Return the total number of load cases.
- unsigned int $*$ GetNLOAD ()

  Return the number of concentrated loads applied in each load case.
- CLoadCaseData $*$ GetLoadCases ()

  Return the list of load cases.
- CSkylineMatrix$<$ double $>$ $*$ GetStiffnessMatrix ()

  Return pointer to the banded stiffness matrix.

Static Public Member Functions

- static CDomain $*$ Instance ()

  Return pointer to the instance of the Domain class.

Private Attributes

- ifstream Input

  Input file stream for reading data from input data file.
- char Title [256]

  Heading information for use in labeling the outpu.
- unsigned int MODEX

  Solution MODEX.
- unsigned int NUMNP

  Total number of nodal points.
- CNode ∗ NodeList

  List of all nodes in the domain.
- unsigned int NUMEG

  Total number of element groups.
- unsigned int ∗ ElementTypes

  Element type of each group.
- unsigned int ∗ NUME

  Number of elements in each element group.
- CElement ∗∗ ElementSetList

  Element Set List.
- unsigned int ∗ NUMMAT

  Number of different sets of material/section properties in each element group.
- CMaterial ∗∗ MaterialSetList

  Material set list.
- unsigned int NLCASE

  Number of load cases.
- CLoadCaseData ∗ LoadCases

  List of all load cases.
- unsigned int ∗ NLOAD

  Number of concentrated loads applied in each load case.
- unsigned int NEQ

  Total number of equations in the system.
- unsigned int NWK

  Number of elements in banded global stiffness matrix.
- unsigned int MK

  Maximum half bandwith.
- CSkylineMatrix< double > ∗ StiffnessMatrix

  Banded stiffness matrix.
- double ∗ Force

  Global nodal force/displacement vector.

Static Private Attributes

- static CDomain ∗ _instance = NULL

  The instance of the Domain class.

4.3.1 Detailed Description

Domain class : Define the problem domain.

Only a single instance of Domain class can be created

4.3.2 Constructor & Destructor Documentation

4.3.2.1 CDomain()

```
CDomain::CDomain ( )
```

Constructor.

4.3.2.2 ∼CDomain()

```
CDomain::∼CDomain ( )
```

Desconstructor.

4.3.3 Member Function Documentation

4.3.3.1 AllocateMatrices()

```
void CDomain::AllocateMatrices ( )
```

Allocate storage for matrices.

Allocate Force, ColumnHeights, DiagonalAddress and StiffnessMatrix and calculate the column heights and address of diagonal elements

4.3.3.2 AssembleForce()

```
bool CDomain::AssembleForce (
             unsigned int LoadCase )
```

Assemble the global nodal force vector for load case LoadCase.

4.3.3.3 AssembleStiffnessMatrix()

```
void CDomain::AssembleStiffnessMatrix ( )
```

Assemble the banded gloabl stiffness matrix.

4.3.3.4 CalculateColumnHeights()

```
void CDomain::CalculateColumnHeights ( )
```

Calculate column heights.

4.3.3.5 CalculateDiagnoalAddress()

```
void CDomain::CalculateDiagnoalAddress ( )
```

Calculate address of diagonal elements in banded matrix.

4.3.3.6 CalculateEquationNumber()

```
void CDomain::CalculateEquationNumber ( )
```

Calculate global equation numbers corresponding to every degree of freedom of each node.

4.3.3.7 GetDisplacement()

```
double* CDomain::GetDisplacement ( ) [inline]
```

Return pointer to the global nodal displacement vector.

4.3.3.8 GetElementSetList()

```
CElement** CDomain::GetElementSetList ( ) [inline]
```

Return element Set List.

4.3.3.9 GetElementTypes()

```
unsigned int* CDomain::GetElementTypes ( ) [inline]
```

Element type of each group.

4.3.3.10 GetForce()

```
double* CDomain::GetForce ( ) [inline]
```

Return pointer to the global nodal force vector.

4.3.3.11 GetLoadCases()

```
CLoadCaseData* CDomain::GetLoadCases ( ) [inline]
```

Return the list of load cases.

4.3.3.12 GetMaterialSetList()

```
CMaterial** CDomain::GetMaterialSetList ( ) [inline]
```

Return material set list.

4.3.3.13 GetMK()

```
unsigned int CDomain::GetMK ( ) [inline]
```

Return the maximum half bandwith.

4.3.3.14 GetMODEX()

```
unsigned int CDomain::GetMODEX ( ) [inline]
```

Return solution mode.

4.3.3.15 GetNEQ()

```
unsigned int CDomain::GetNEQ ( ) [inline]
```

Return the total number of equations.

4.3.3.16 GetNLCASE()

```
unsigned int CDomain::GetNLCASE ( )  [inline]
```

Return the total number of load cases.

4.3.3.17 GetNLOAD()

```
unsigned int* CDomain::GetNLOAD ( )  [inline]
```

Return the number of concentrated loads applied in each load case.

4.3.3.18 GetNodeList()

```
CNode* CDomain::GetNodeList ( )  [inline]
```

Return the node list.

4.3.3.19 GetNUME()

```
unsigned int* CDomain::GetNUME ( )  [inline]
```

Return the number of elements in each element group.

4.3.3.20 GetNUMEG()

```
unsigned int CDomain::GetNUMEG ( )  [inline]
```

Return total number of element groups.

4.3.3.21 GetNUMMAT()

```
unsigned int* CDomain::GetNUMMAT ( )  [inline]
```

Return number of different sets of material/section properties in each element group.

4.3.3.22 GetNUMNP()

```
unsigned int CDomain::GetNUMNP ( )  [inline]
```

Return the total number of nodal points.

4.3.3.23 GetNWK()

```
unsigned int CDomain::GetNWK ( )  [inline]
```

Return the number of banded global stiffness matrix elements.

4.3.3.24 GetStiffnessMatrix()

```
CSkylineMatrix<double>* CDomain::GetStiffnessMatrix ( )  [inline]
```

Return pointer to the banded stiffness matrix.

4.3.3.25 GetTitle()

```
string CDomain::GetTitle ( )  [inline]
```

Return the title of problem.

4.3.3.26 Instance()

```
CDomain * CDomain::Instance ( )  [static]
```

Return pointer to the instance of the Domain class.

4.3.3.27 ReadBarElementData()

```
bool CDomain::ReadBarElementData (
             unsigned int EleGrp )
```

Read bar element data from the input data file.

4.3.3.28 ReadData()

```
bool CDomain::ReadData (
             string FileName,
             string OutFile )
```

Read domain data from the input data file.

4.3.3.29 ReadElements()

```
bool CDomain::ReadElements ( )
```

Read element data.

4.3.3.30 ReadLoadCases()

```
bool CDomain::ReadLoadCases ( )
```

Read load case data.

4.3.3.31 ReadNodalPoints()

```
bool CDomain::ReadNodalPoints ( )
```

Read nodal point data.

4.3.4 Member Data Documentation

4.3.4.1 _instance

```
CDomain * CDomain::_instance = NULL  [static], [private]
```

The instance of the Domain class.

4.3.4.2 ElementSetList

```
CElement** CDomain::ElementSetList  [private]
```

Element Set List.

ElementSetList[i] - ith element set

ElementSetList[i][j] - jth element in ith set

4.3.4.3 ElementTypes

```
unsigned int* CDomain::ElementTypes  [private]
```

Element type of each group.

4.3.4.4 Force

```
double* CDomain::Force  [private]
```

Global nodal force/displacement vector.

4.3.4.5 Input

```
ifstream CDomain::Input  [private]
```

Input file stream for reading data from input data file.

4.3.4.6 LoadCases

```
CLoadCaseData* CDomain::LoadCases  [private]
```

List of all load cases.

4.3.4.7 MaterialSetList

```
CMaterial** CDomain::MaterialSetList  [private]
```

Material set list.

MaterialSetList[i] - ith material set

MaterialSetList[i][j] - jth material in ith set

4.3.4.8 MK

```
unsigned int CDomain::MK [private]
```

Maximum half bandwith.

4.3.4.9 MODEX

```
unsigned int CDomain::MODEX [private]
```

Solution MODEX.

0 : Data check only; 1 : Execution

4.3.4.10 NEQ

```
unsigned int CDomain::NEQ [private]
```

Total number of equations in the system.

4.3.4.11 NLCASE

```
unsigned int CDomain::NLCASE [private]
```

Number of load cases.

4.3.4.12 NLOAD

```
unsigned int* CDomain::NLOAD [private]
```

Number of concentrated loads applied in each load case.

4.3.4.13 NodeList

```
CNode* CDomain::NodeList [private]
```

List of all nodes in the domain.

4.3.4.14 NUME

```
unsigned int* CDomain::NUME  [private]
```

Number of elements in each element group.

4.3.4.15 NUMEG

```
unsigned int CDomain::NUMEG  [private]
```

Total number of element groups.

An element group consists of a convenient collection of elements with same type

4.3.4.16 NUMMAT

```
unsigned int* CDomain::NUMMAT  [private]
```

Number of different sets of material/section properties in each element group.

4.3.4.17 NUMNP

```
unsigned int CDomain::NUMNP  [private]
```

Total number of nodal points.

4.3.4.18 NWK

```
unsigned int CDomain::NWK  [private]
```

Number of elements in banded global stiffness matrix.

4.3.4.19 StiffnessMatrix

```
CSkylineMatrix<double>* CDomain::StiffnessMatrix  [private]
```

Banded stiffness matrix.

A one-dimensional array storing only the elements below the skyline of the global stiffness matrix.

4.3.4.20 Title

```
char CDomain::Title[256]  [private]
```

Heading information for use in labeling the outpu.

The documentation for this class was generated from the following files:

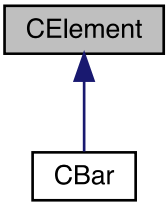
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp

4.4 CElement Class Reference

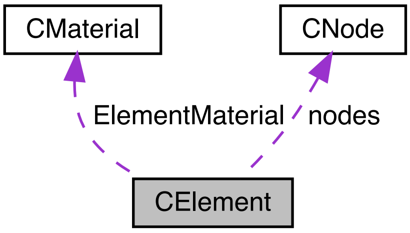
Element base class.

```
#include <Element.h>
```

Inheritance diagram for CElement:



Collaboration diagram for CElement:



Public Member Functions

- CElement ()

  Constructor.
- virtual bool Read (ifstream &Input, unsigned int Ele, CMaterial ∗MaterialSets, CNode ∗NodeList)=0

  Read element data from stream Input.
- virtual void Write (ofstream &OutputFile, unsigned int Ele)=0

  Write element data to stream OutputFile.
- virtual void GenerateLocationMatrix ()=0

  Generate location matrix: the global equation number that corresponding to each DOF of the element.
- void CalculateColumnHeight (unsigned int ∗ColumnHeight)

  Calculate the column height, used with the skyline storage scheme.
- void assembly (double ∗Matrix, CSkylineMatrix< double > ∗StiffnessMatrix)

  Assemble the element stiffness matrix to the global stiffness matrix.
- virtual void ElementStiffness (double ∗stiffness)=0

  Calculate element stiffness matrix (Upper triangular matrix, stored as an array column by colum)
- virtual void ElementStress (double ∗stress, double ∗Displacement)=0

  Calculate element stress.
- CNode ∗∗ GetNodes ()

  Return nodes of the element.
- CMaterial ∗ GetElementMaterial ()

  Return material of the element.
- virtual unsigned int SizeOfStiffnessMatrix ()=0

  Return the size of the element stiffness matrix (stored as an array column by column)

Protected Attributes

- unsigned int NEN

  Number of nodes per element.
- CNode ∗∗ nodes

  Nodes of the element.
- CMaterial ∗ ElementMaterial

  Material of the element.
- unsigned int ∗ LocationMatrix

  Location Matrix of the element.
- unsigned int ND

  Dimension of the location matrix.

Friends

- class CDomain

4.4.1 Detailed Description

Element base class.

All type of element classes should be derived from this base class

4.4.2 Constructor & Destructor Documentation

4.4.2.1 CElement()

```
CElement::CElement ( )  [inline]
```

Constructor.

4.4.3 Member Function Documentation

4.4.3.1 assembly()

```
void CElement::assembly (
             double * Matrix,
             CSkylineMatrix< double > * StiffnessMatrix )
```

Assemble the element stiffness matrix to the global stiffness matrix.

4.4.3.2 CalculateColumnHeight()

```
void CElement::CalculateColumnHeight (
             unsigned int * ColumnHeight )
```

Calculate the column height, used with the skyline storage scheme.

4.4.3.3 ElementStiffness()

```
virtual void CElement::ElementStiffness (
             double * stiffness )  [pure virtual]
```

Calculate element stiffness matrix (Upper triangular matrix, stored as an array column by colum)

Implemented in CBar.

4.4.3.4 ElementStress()

```
virtual void CElement::ElementStress (
             double * stress,
             double * Displacement )  [pure virtual]
```

Calculate element stress.

Implemented in CBar.

4.4.3.5 GenerateLocationMatrix()

```
virtual void CElement::GenerateLocationMatrix ( )  [pure virtual]
```

Generate location matrix: the global equation number that corresponding to each DOF of the element.

Implemented in CBar.

4.4.3.6 GetElementMaterial()

```
CMaterial* CElement::GetElementMaterial ( )  [inline]
```

Return material of the element.

4.4.3.7 GetNodes()

```
CNode** CElement::GetNodes ( )  [inline]
```

Return nodes of the element.

4.4.3.8 Read()

```
virtual bool CElement::Read (
             ifstream & Input,
             unsigned int Ele,
             CMaterial * MaterialSets,
             CNode * NodeList )  [pure virtual]
```

Read element data from stream Input.

Implemented in CBar.

4.4.3.9 SizeOfStiffnessMatrix()

```
virtual unsigned int CElement::SizeOfStiffnessMatrix ( )  [pure virtual]
```

Return the size of the element stiffness matrix (stored as an array column by column)

Implemented in CBar.

4.4.3.10 Write()

```
virtual void CElement::Write (
             ofstream & OutputFile,
             unsigned int Ele )  [pure virtual]
```

Write element data to stream OutputFile.

Implemented in CBar.

4.4.4 Friends And Related Function Documentation

4.4.4.1 CDomain

```
friend class CDomain  [friend]
```

4.4.5 Member Data Documentation

4.4.5.1 ElementMaterial

```
CMaterial* CElement::ElementMaterial  [protected]
```

Material of the element.

Pointer to an element of MaterialSetList[][]

4.4.5.2 LocationMatrix

```
unsigned int* CElement::LocationMatrix  [protected]
```

Location Matrix of the element.

4.4.5.3 ND

```
unsigned int CElement::ND [protected]
```

Dimension of the location matrix.

4.4.5.4 NEN

```
unsigned int CElement::NEN [protected]
```

Number of nodes per element.

4.4.5.5 nodes

```
CNode** CElement::nodes [protected]
```

Nodes of the element.

The documentation for this class was generated from the following files:

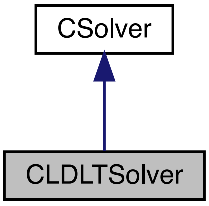
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Element.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Element.cpp

4.5 CLDLTSolver Class Reference

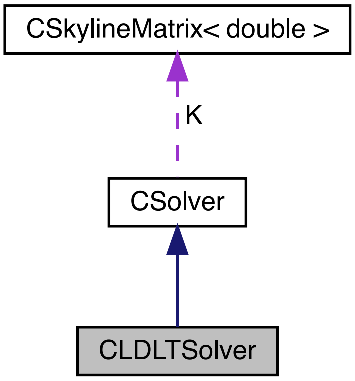
LDLT solver: A in core solver using skyline storage and column reduction scheme.

```
#include <Solver.h>
```

Inheritance diagram for CLDLTSolver:



Collaboration diagram for CLDLTSolver:



Public Member Functions

- CLDLTSolver (CSkylineMatrix< double > ∗K)

  Constructor.
- void LDLT ()

  Perform L∗D∗L(T) factorization of the stiffness matrix.
- void BackSubstitution (double ∗Force)

  Reduce right-hand-side load vector and back substitute.

Additional Inherited Members

4.5.1 Detailed Description

LDLT solver: A in core solver using skyline storage and column reduction scheme.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 CLDLTSolver()

```
CLDLTSolver::CLDLTSolver (
            CSkylineMatrix< double > ∗ K )   [inline]
```

Constructor.

4.5.3 Member Function Documentation

4.5.3.1 BackSubstitution()

```
void CLDLTSolver::BackSubstitution (
             double * Force )
```

Reduce right-hand-side load vector and back substitute.

4.5.3.2 LDLT()

```
void CLDLTSolver::LDLT ( )
```

Perform L∗D∗L(T) factorization of the stiffness matrix.

The documentation for this class was generated from the following files:

- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Solver.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Solver.cpp

4.6 CLoadCaseData Class Reference

Class LoadData is used to store load data.

```
#include <LoadCaseData.h>
```

Public Member Functions

- CLoadCaseData ()
- ∼CLoadCaseData ()
- void Allocate (unsigned int num)

  Set nloads, and new array node, dof and load.
- bool Read (ifstream &Input, unsigned int lcase)

  Read load case data from stream Input.
- void Write (ofstream &OutputFile, unsigned int lcase)

  Write load case data to stream OutputFile.

Public Attributes

- unsigned int nloads

  Number of concentrated loads in this load case.
- unsigned int ∗ node

  Node number to which this load is applied.
- unsigned int ∗ dof

  Degree of freedom number for this load component.
- double ∗ load

  Magnitude of load.

4.6.1 Detailed Description

Class LoadData is used to store load data.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 CLoadCaseData()

```
CLoadCaseData::CLoadCaseData ( )  [inline]
```

4.6.2.2 ∼CLoadCaseData()

```
CLoadCaseData::∼CLoadCaseData ( )
```

4.6.3 Member Function Documentation

4.6.3.1 Allocate()

```
void CLoadCaseData::Allocate (
             unsigned int num )
```

Set nloads, and new array node, dof and load.

4.6.3.2 Read()

```
bool CLoadCaseData::Read (
             ifstream & Input,
             unsigned int lcase )
```

Read load case data from stream Input.

4.6.3.3 Write()

```
void CLoadCaseData::Write (
             ofstream & OutputFile,
             unsigned int lcase )
```

Write load case data to stream OutputFile.

4.6.4 Member Data Documentation

4.6.4.1 dof

```
unsigned int* CLoadCaseData::dof
```

Degree of freedom number for this load component.

4.6.4.2 load

```
double* CLoadCaseData::load
```

Magnitude of load.

4.6.4.3 nloads

```
unsigned int CLoadCaseData::nloads
```

Number of concentrated loads in this load case.

4.6.4.4 node

```
unsigned int* CLoadCaseData::node
```

Node number to which this load is applied.

The documentation for this class was generated from the following files:

- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/LoadCaseData.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/LoadCaseData.cpp

4.7 Clock Class Reference

Clock class for timing.

```
#include <Clock.h>
```

Public Member Functions

- Clock ()

  Constructor.
- void Start ()

  Start the clock.
- void Stop ()

  Stop the clock.
- void Resume ()

  Resume the stoped clock.
- void Clear ()

  Clear the clock.
- double ElapsedTime ()

  Return the elapsed time since the clock started.

Private Attributes

- clock_t t0_
- clock_t t1_
- double ct_
- bool st0_

  Flag for Start method.
- bool st1_

  Flag for Stop method.

4.7.1 Detailed Description

Clock class for timing.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Clock()

```
Clock::Clock ( )
```

Constructor.

4.7.3 Member Function Documentation

4.7.3.1 Clear()

```
void Clock::Clear ( )
```

Clear the clock.

4.7.3.2 ElapsedTime()

```
double Clock::ElapsedTime ( )
```

Return the elapsed time since the clock started.

4.7.3.3 Resume()

```
void Clock::Resume ( )
```

Resume the stoped clock.

4.7.3.4 Start()

```
void Clock::Start ( )
```

Start the clock.

4.7.3.5 Stop()

```
void Clock::Stop ( )
```

Stop the clock.

4.7.4 Member Data Documentation

4.7.4.1 ct_

```
double Clock::ct_  [private]
```

4.7.4.2 st0_

```
bool Clock::st0_  [private]
```

Flag for Start method.

4.7.4.3 st1_

```
bool Clock::st1_  [private]
```

Flag for Stop method.

4.7.4.4 t0_

```
clock_t Clock::t0_  [private]
```

4.7.4.5 t1_

```
clock_t Clock::t1_  [private]
```

The documentation for this class was generated from the following files:

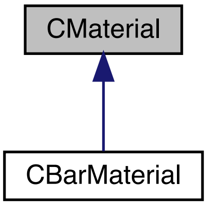
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Clock.cpp

4.8 CMaterial Class Reference

Material base class which only define one data member.

```
#include <Material.h>
```

Inheritance diagram for CMaterial:



Public Member Functions

- virtual bool Read (ifstream &Input, unsigned int mset)=0

  Read material data from stream Input.
- virtual void Write (ofstream &OutputFile, unsigned int mset)=0

  Write material data to Stream OutputFile.

Public Attributes

- unsigned int nset

  Number of set.
- double E

  Young's modulus.

4.8.1 Detailed Description

Material base class which only define one data member.

All type of material classes should be derived from this base class

4.8.2 Member Function Documentation

4.8.2.1 Read()

```
virtual bool CMaterial::Read (
             ifstream & Input,
             unsigned int mset )  [pure virtual]
```

Read material data from stream Input.

Implemented in CBarMaterial.

4.8.2.2 Write()

```
virtual void CMaterial::Write (
             ofstream & OutputFile,
             unsigned int mset )  [pure virtual]
```

Write material data to Stream OutputFile.

Implemented in CBarMaterial.

4.8.3 Member Data Documentation

4.8.3.1 E

```
double CMaterial::E
```

Young's modulus.

4.8.3.2 nset

```
unsigned int CMaterial::nset
```

Number of set.

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

- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h

4.9 CNode Class Reference

Node class.

```
#include <Node.h>
```

Public Member Functions

- CNode (double X=0, double Y=0, double Z=0)

  Constructor.
- bool Read (ifstream &Input, unsigned int np)

  Read nodal point data from stream Input.
- void Write (ofstream &OutputFile, unsigned int np)

  Output nodal point data to stream OutputFile.
- void WriteEquationNo (ofstream &OutputFile, unsigned int np)

  Output equation numbers of nodal point to stream OutputFile.
- void WriteNodalDisplacement (ofstream &OutputFile, unsigned int np, double ∗Displacement)

  Write nodal displacement.

Public Attributes

- unsigned int NodeNumber

  Node numer.
- double XYZ [NDF]

  x, y and z coordinates of the node
- unsigned int bcode [NDF]

  Boundary code of each degree of freedom of the node.

Static Public Attributes

- static const unsigned int NDF = 3

  Maximum number of degrees of freedom per node.

4.9.1 Detailed Description

Node class.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 CNode()

```
CNode::CNode (
             double X = 0,
             double Y = 0,
             double Z = 0 )
```

Constructor.

4.9.3 Member Function Documentation

4.9.3.1 Read()

```
bool CNode::Read (
             ifstream & Input,
             unsigned int np )
```

Read nodal point data from stream Input.

4.9.3.2 Write()

```
void CNode::Write (
             ofstream & OutputFile,
             unsigned int np )
```

Output nodal point data to stream OutputFile.

4.9.3.3 WriteEquationNo()

```
void CNode::WriteEquationNo (
             ofstream & OutputFile,
             unsigned int np )
```

Output equation numbers of nodal point to stream OutputFile.

4.9.3.4 WriteNodalDisplacement()

```
void CNode::WriteNodalDisplacement (
             ofstream & OutputFile,
             unsigned int np,
             double * Displacement )
```

Write nodal displacement.

4.9.4 Member Data Documentation

4.9.4.1 bcode

```
unsigned int CNode::bcode[NDF]
```

Boundary code of each degree of freedom of the node.

0: The corresponding degree of freedom is active (defined in the global system)

1: The corresponding degree of freedom in nonactive (not defined)

After call Domain::CalculateEquationNumber(), bcode stores the global equation number

corresponding to each degree of freedom of the node

4.9.4.2 NDF

```
const unsigned int CNode::NDF = 3  [static]
```

Maximum number of degrees of freedom per node.

For 3D bar and solid elements, NDF = 3. For 3D beam or shell elements, NDF = 5 or 6

4.9.4.3 NodeNumber

```
unsigned int CNode::NodeNumber
```

Node numer.

4.9.4.4 XYZ

```
double CNode::XYZ[NDF]
```

x, y and z coordinates of the node

The documentation for this class was generated from the following files:

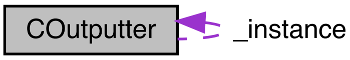
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Node.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Node.cpp

4.10 COutputter Class Reference

Outputer class is used to output results.

```
#include <Outputter.h>
```

Collaboration diagram for COutputter:



Public Member Functions

- ofstream ∗ GetOutputFile ()

    Return pointer to the output file stream.
- void PrintTime (const struct tm ∗ptm, ostream &output)

    Output current time and date.
- void OutputHeading ()

    Output logo and heading.
- void OutputNodeInfo ()

    Output nodal point data.
- void OutputEquationNumber ()

    Output equation numbers.
- void OutputElementInfo ()

    Output element data.
- void PrintBarElementData (unsigned int EleGrp)

    Output bar element data.
- void OutputLoadInfo ()

    Output load data.
- void OutputNodalDisplacement (unsigned int lcase)

    Output displacement data.
- void OutputElementStress ()

    Output element stresses.
- void OutputTotalSystemData ()

    Print total system data.

Static Public Member Functions

- static COutputter ∗ Instance (string FileName=" ")

  Return the single instance of the class.

Protected Member Functions

- COutputter (string FileName)

  Constructor.

Static Protected Attributes

- static COutputter ∗ _instance = NULL

  Designed as a single instance class.

Private Attributes

- ofstream OutputFile

  File stream for output.

4.10.1 Detailed Description

Outputer class is used to output results.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 COutputter()

```
COutputter::COutputter (
          string FileName )  [protected]
```

Constructor.

4.10.3 Member Function Documentation

4.10.3.1 GetOutputFile()

```
ofstream* COutputter::GetOutputFile ( )  [inline]
```

Return pointer to the output file stream.

4.10.3.2 Instance()

```
COutputter * COutputter::Instance (
             string FileName = " " )  [static]
```

Return the single instance of the class.

4.10.3.3 OutputElementInfo()

```
void COutputter::OutputElementInfo ( )
```

Output element data.

4.10.3.4 OutputElementStress()

```
void COutputter::OutputElementStress ( )
```

Output element stresses.

4.10.3.5 OutputEquationNumber()

```
void COutputter::OutputEquationNumber ( )
```

Output equation numbers.

4.10.3.6 OutputHeading()

```
void COutputter::OutputHeading ( )
```

Output logo and heading.

4.10.3.7 OutputLoadInfo()

```
void COutputter::OutputLoadInfo ( )
```

Output load data.

4.10.3.8 OutputNodalDisplacement()

```
void COutputter::OutputNodalDisplacement (
            unsigned int lcase )
```

Output displacement data.

4.10.3.9 OutputNodeInfo()

```
void COutputter::OutputNodeInfo ( )
```

Output nodal point data.

4.10.3.10 OutputTotalSystemData()

```
void COutputter::OutputTotalSystemData ( )
```

Print total system data.

4.10.3.11 PrintBarElementData()

```
void COutputter::PrintBarElementData (
            unsigned int EleGrp )
```

Output bar element data.

4.10.3.12 PrintTime()

```
void COutputter::PrintTime (
            const struct tm * ptm,
            ostream & output )
```

Output current time and date.

4.10.4 Member Data Documentation

4.10.4.1 _instance

```
COutputter * COutputter::_instance = NULL  [static], [protected]
```

Designed as a single instance class.

4.10.4.2 OutputFile

```
ofstream COutputter::OutputFile  [private]
```

File stream for output.

The documentation for this class was generated from the following files:

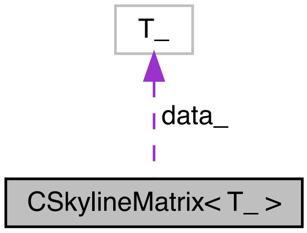
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Outputter.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp

4.11 CSkylineMatrix< T_ > Class Template Reference

CSkylineMatrix class is used to store the FEM stiffness matrix in skyline storage.

```
#include <SkylineMatrix.h>
```

Collaboration diagram for CSkylineMatrix< T_ >:



Public Member Functions

- CSkylineMatrix ()

  constructors
- CSkylineMatrix (unsigned int N)
- ∼CSkylineMatrix ()

  destructor
- T_ & operator() (unsigned int i, unsigned int j)

  operator function (i,j) where i and j number from 1
- T_ operator() (unsigned int i)

  operator function (i) where i numbers from 1
- void Allocate ()

  Allocate storage for the skyline matrix.
- unsigned int ∗ GetColumnHeights ()

  Return pointer to the ColumnHeights_.
- unsigned int ∗ GetDiagonalAddress ()

  Return pointer to the DiagonalAddress_.
- unsigned int dim () const

  Return the dimension of the stiffness matrix.
- unsigned int size () const

  Return the size of the storage used to store the stiffness matrkix in skyline.

Private Attributes

- T_ ∗ data_

  Store the stiffness matrkix in skyline storage.
- unsigned int NEQ_

  Dimension of the stiffness matrix.
- unsigned int NWK_

  Size of the storage used to store the stiffness matrkix in skyline.
- unsigned int ∗ ColumnHeights_

  Column hights.
- unsigned int ∗ DiagonalAddress_

  Diagonal address of all columns in data_.

4.11.1 Detailed Description

template<class T_>
class CSkylineMatrix< T_ >

CSkylineMatrix class is used to store the FEM stiffness matrix in skyline storage.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 CSkylineMatrix() [1/2]

```
template<class T_ >
CSkylineMatrix< T_ >::CSkylineMatrix ( )  [inline]
```

constructors

constructor functions

4.11.2.2 CSkylineMatrix() [2/2]

```
template<class T_ >
CSkylineMatrix< T_ >::CSkylineMatrix (
             unsigned int N )  [inline]
```

4.11.2.3 ∼CSkylineMatrix()

```
template<class T_ >
CSkylineMatrix< T_ >::∼CSkylineMatrix< T_ > ( )  [inline]
```

destructor

destructor function

4.11.3 Member Function Documentation

4.11.3.1 Allocate()

```
template<class T_ >
void CSkylineMatrix< T_ >::Allocate ( )  [inline]
```

Allocate storage for the skyline matrix.

Allocate storage for the matrix.

4.11.3.2 dim()

```
template<class T_ >
unsigned int CSkylineMatrix< T_ >::dim ( ) const  [inline]
```

Return the dimension of the stiffness matrix.

4.11.3.3 GetColumnHeights()

```
template<class T_ >
unsigned int * CSkylineMatrix< T_ >::GetColumnHeights ( ) [inline]
```

Return pointer to the ColumnHeights_.

4.11.3.4 GetDiagonalAddress()

```
template<class T_ >
unsigned int * CSkylineMatrix< T_ >::GetDiagonalAddress ( ) [inline]
```

Return pointer to the DiagonalAddress_.

4.11.3.5 operator()() [1/2]

```
template<class T_ >
T_ & CSkylineMatrix< T_ >::operator() (
             unsigned int i,
             unsigned int j ) [inline]
```

operator function (i,j) where i and j number from 1

operator (i,j) where i and j number from 1 For the sake of efficiency, the index bounds are not checked

4.11.3.6 operator()() [2/2]

```
template<class T_ >
T_ CSkylineMatrix< T_ >::operator() (
             unsigned int i ) [inline]
```

operator function (i) where i numbers from 1

operator (i) where i numbers from 1 For the sake of efficiency, the index bounds are not checked

4.11.3.7 size()

```
template<class T_ >
unsigned int CSkylineMatrix< T_ >::size ( ) const [inline]
```

Return the size of the storage used to store the stiffness matrkix in skyline.

4.11.4 Member Data Documentation

4.11.4.1 ColumnHeights_

```
template<class T_>
unsigned int* CSkylineMatrix< T_ >::ColumnHeights_  [private]
```

Column hights.

4.11.4.2 data_

```
template<class T_>
T_* CSkylineMatrix< T_ >::data_  [private]
```

Store the stiffness matrkix in skyline storage.

4.11.4.3 DiagonalAddress_

```
template<class T_>
unsigned int* CSkylineMatrix< T_ >::DiagonalAddress_  [private]
```

Diagonal address of all columns in data_.

4.11.4.4 NEQ_

```
template<class T_>
unsigned int CSkylineMatrix< T_ >::NEQ_  [private]
```

Dimension of the stiffness matrix.

4.11.4.5 NWK_

```
template<class T_>
unsigned int CSkylineMatrix< T_ >::NWK_  [private]
```

Size of the storage used to store the stiffness matrkix in skyline.

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

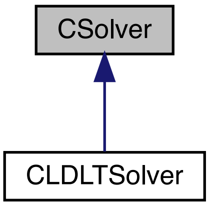
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/SkylineMatrix.h

4.12 CSolver Class Reference

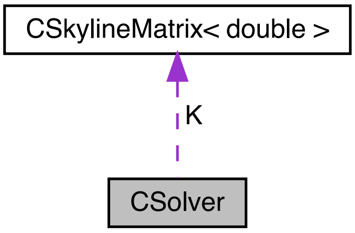
Base class for a solver.

```
#include <Solver.h>
```

Inheritance diagram for CSolver:



Collaboration diagram for CSolver:



Public Member Functions

- CSolver (CSkylineMatrix< double > ∗K)

Protected Attributes

- CSkylineMatrix< double > ∗ K

4.12.1 Detailed Description

Base class for a solver.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 CSolver()

```
CSolver::CSolver (
            CSkylineMatrix< double > * K )
```

4.12.3 Member Data Documentation

4.12.3.1 K

```
CSkylineMatrix<double>* CSolver::K  [protected]
```

The documentation for this class was generated from the following files:

- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Solver.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Solver.cpp

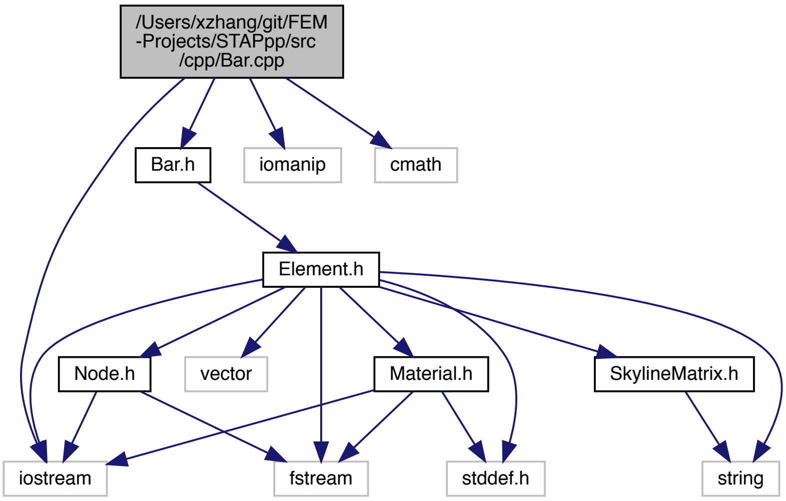
Chapter 5

File Documentation

5.1 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp File Reference

```
#include "Bar.h"
#include <iostream>
#include <iomanip>
#include <cmath>
```

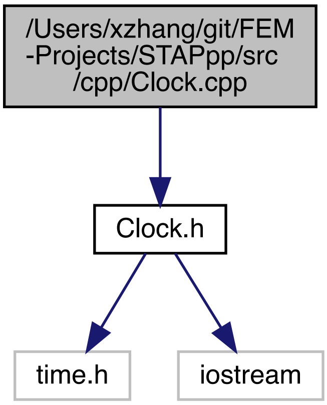
Include dependency graph for Bar.cpp:



5.2 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Clock.cpp File Reference

```
#include "Clock.h"
```

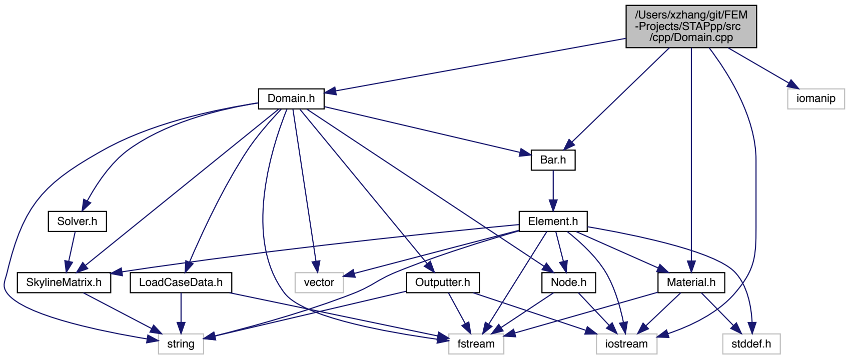
Include dependency graph for Clock.cpp:



5.3 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp File Reference

```
#include "Domain.h"
#include "Bar.h"
#include "Material.h"
#include <iomanip>
#include <iostream>
```

Include dependency graph for Domain.cpp:



Functions

- template<class type >
  void clear (type ∗a, unsigned int N)

  Clear an array.

5.3.1 Function Documentation

5.3.1.1 clear()

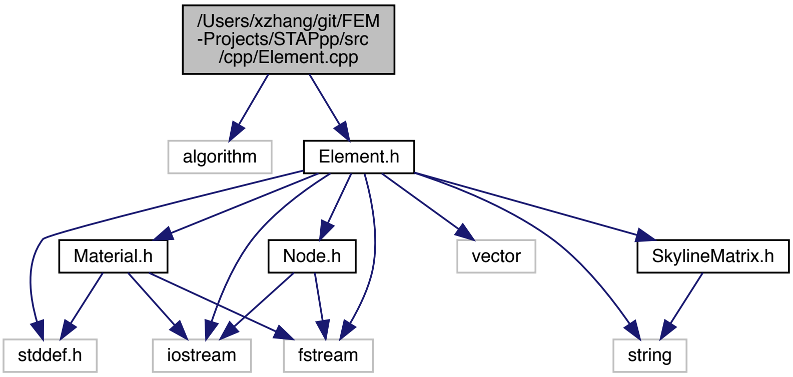
```
template<class type >
void clear (
            type * a,
            unsigned int N )
```

Clear an array.

5.4 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Element.cpp File Reference

```
#include <algorithm>
#include "Element.h"
```

Include dependency graph for Element.cpp:

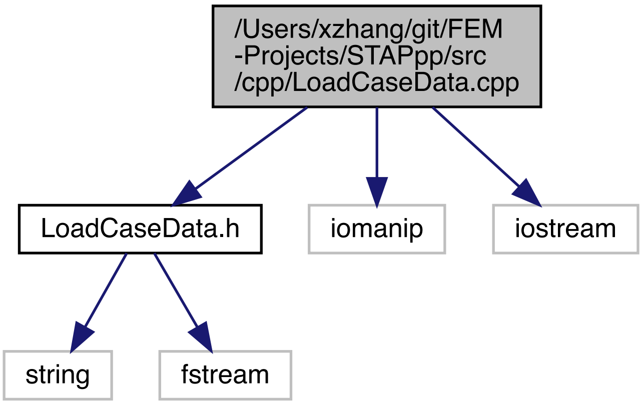


5.5 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/LoadCaseData.cpp File Reference

```
#include "LoadCaseData.h"
#include <iomanip>
```

```
#include <iostream>
```

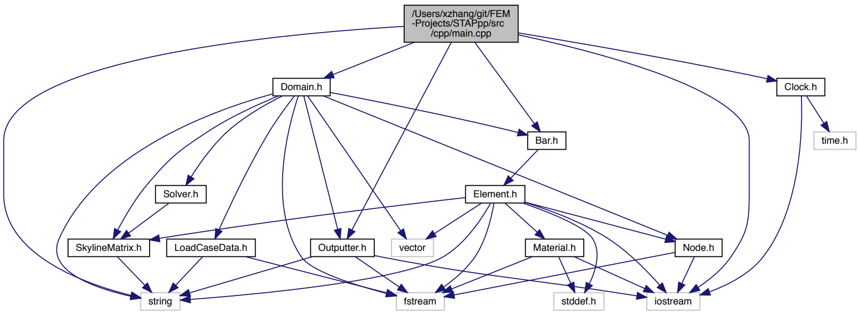
Include dependency graph for LoadCaseData.cpp:



5.6 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp File Reference

```
#include <string>
#include <iostream>
#include "Domain.h"
#include "Bar.h"
#include "Outputter.h"
#include "Clock.h"
```

Include dependency graph for main.cpp:



Functions

- int main (int argc, char ∗argv[ ])

5.6.1 Function Documentation

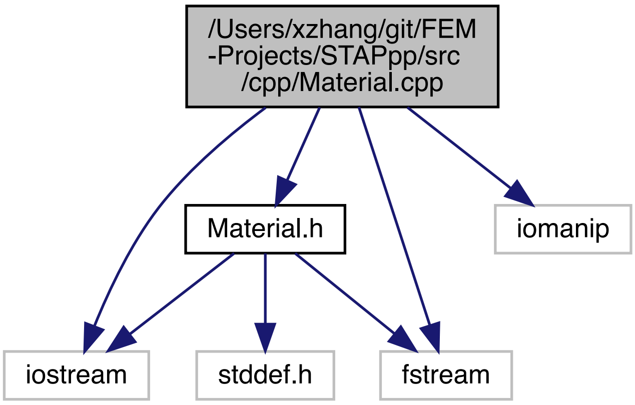
5.6.1.1 main()

```
int main (
            int argc,
            char * argv[] )
```

5.7 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp File Reference

```
#include "Material.h"
#include <iostream>
#include <fstream>
#include <iomanip>
```

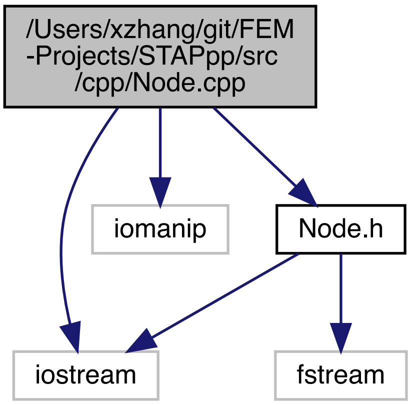
Include dependency graph for Material.cpp:



5.8 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Node.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include "Node.h"
```

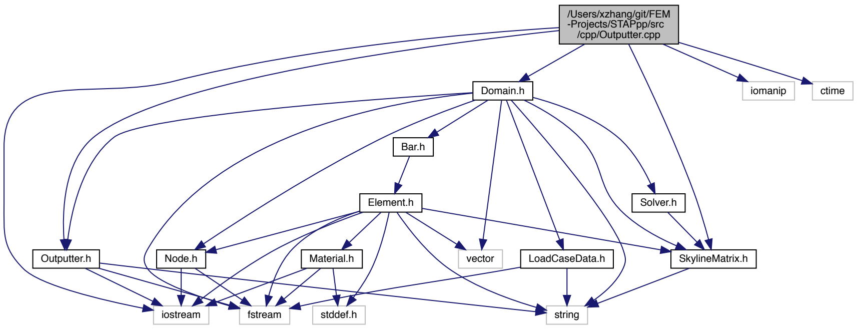
Include dependency graph for Node.cpp:



5.9 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp File Reference

```
#include "Domain.h"
#include "Outputter.h"
#include "SkylineMatrix.h"
#include <iostream>
#include <iomanip>
#include <ctime>
```

Include dependency graph for Outputter.cpp:

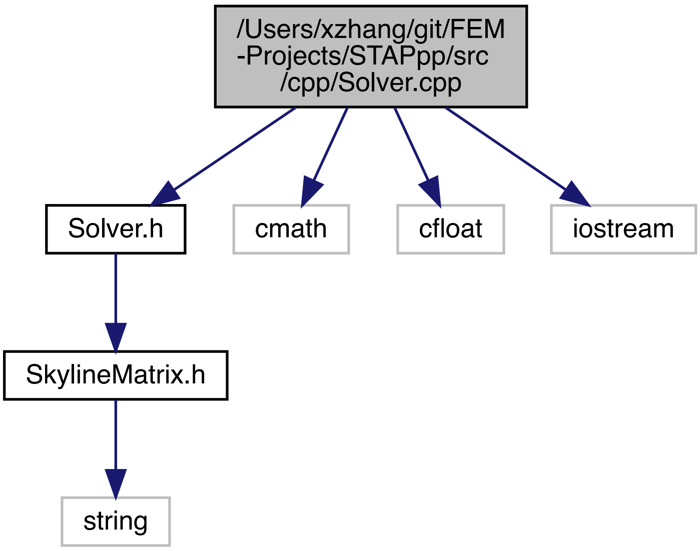


5.10 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Solver.cpp File Reference

```
#include "Solver.h"
#include <cmath>
#include <cfloat>
```

```
#include <iostream>
```

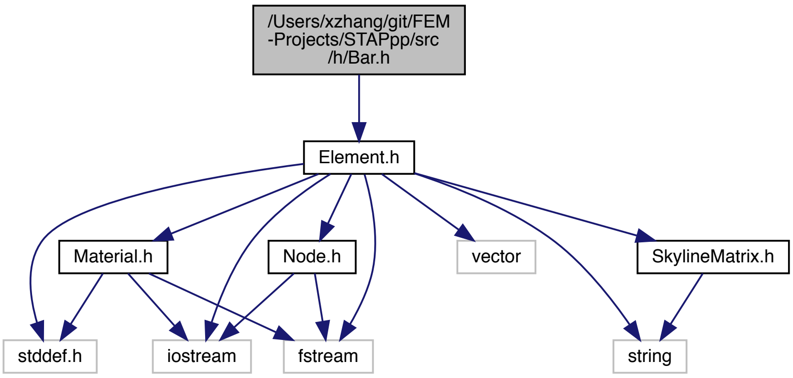
Include dependency graph for Solver.cpp:



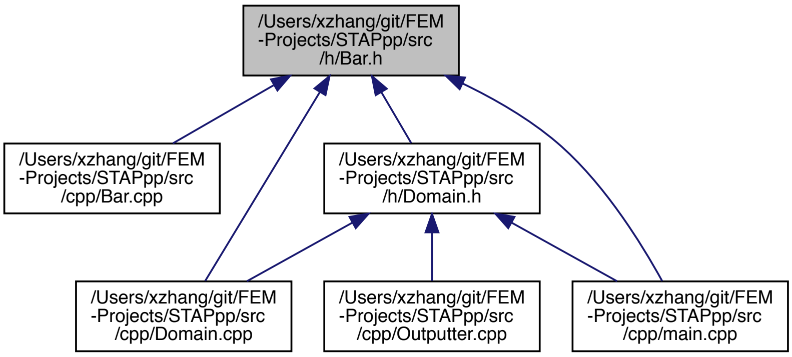
5.11 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h File Reference

```
#include "Element.h"
```

Include dependency graph for Bar.h:



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


/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp

Classes

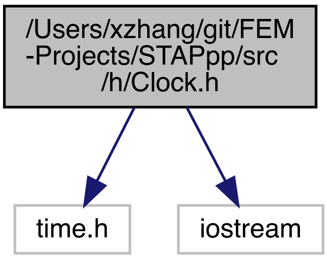
- class CBar

  Bar element class.

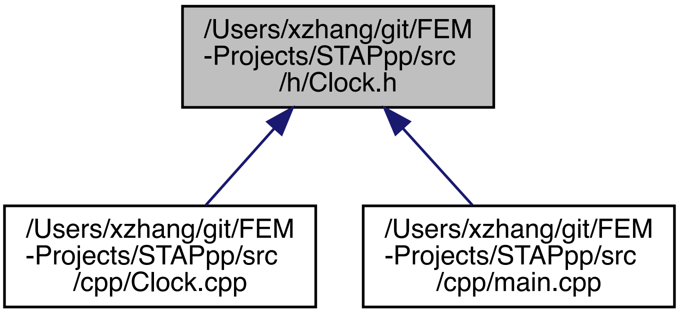
5.12 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h File Reference

```
#include <time.h>
#include <iostream>
```

Include dependency graph for Clock.h:



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



Classes

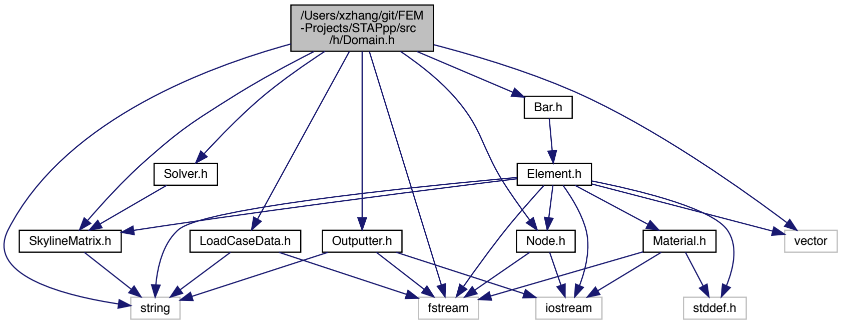
- class Clock

  Clock class for timing.

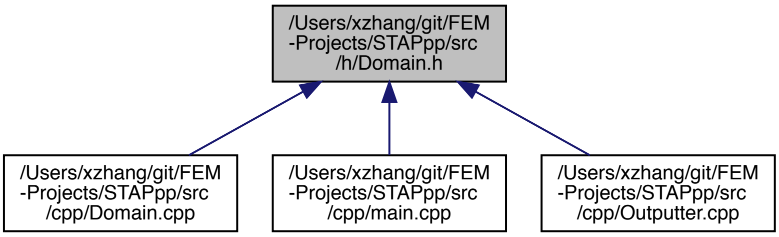
5.13 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h File Reference

```
#include <string>
#include <fstream>
#include <vector>
#include "Node.h"
#include "Bar.h"
#include "Outputter.h"
#include "Solver.h"
#include "LoadCaseData.h"
#include "SkylineMatrix.h"
```

Include dependency graph for Domain.h:



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


/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp

Classes

- class CDomain

  Domain class : Define the problem domain.

Functions

- template<class type >
  void clear (type ∗a, unsigned int N)

  Clear an array.

5.13.1 Function Documentation

5.13.1.1 clear()

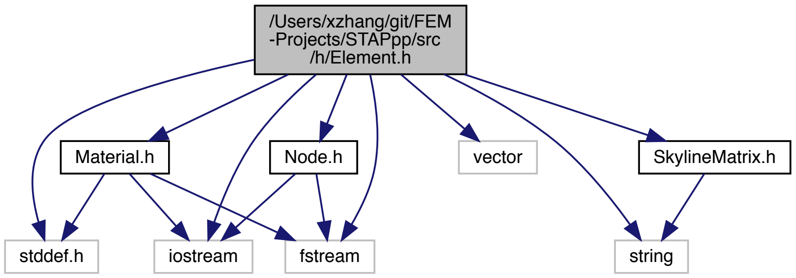
```
template<class type >
void clear (
             type ∗ a,
             unsigned int N )
```

Clear an array.

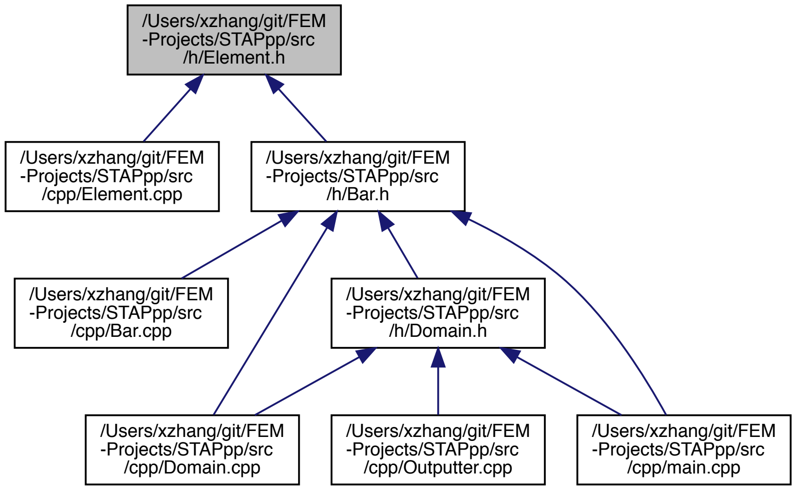
5.14 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Element.h File Reference

```
#include <stddef.h>
#include <vector>
#include <string>
#include <iostream>
#include <fstream>
#include "Node.h"
#include "Material.h"
#include "SkylineMatrix.h"
```

Include dependency graph for Element.h:



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



Classes

- class CElement

    Element base class.

Functions

- template<class type >
  void clear (type ∗a, unsigned int N)

  Clear an array.

5.14.1 Function Documentation

5.14.1.1 clear()

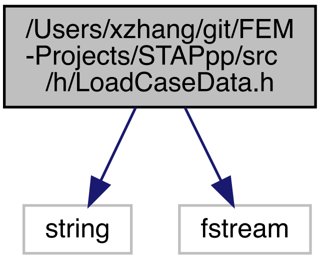
```
template<class type >
void clear (
            type * a,
            unsigned int N )
```

Clear an array.

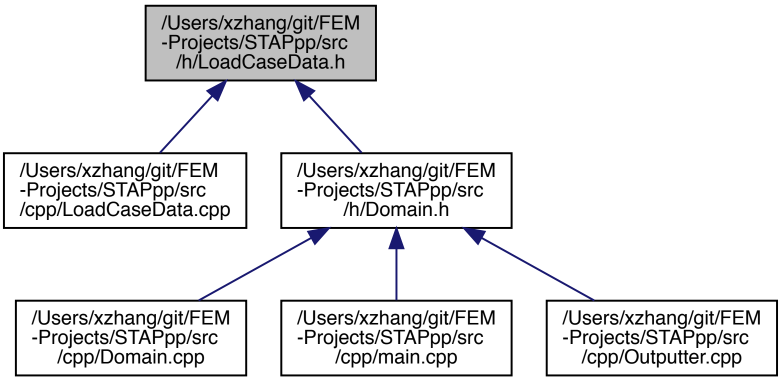
5.15 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/LoadCaseData.h File Reference

```
#include <string>
#include <fstream>
```

Include dependency graph for LoadCaseData.h:



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


/Users/xzhang/git/FEM-Projects/STAPpp/src/h/LoadCaseData.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/LoadCaseData.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp

Classes

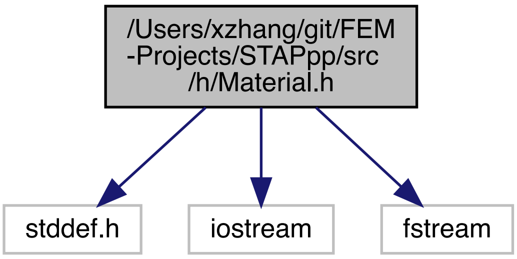
- class CLoadCaseData

  Class LoadData is used to store load data.

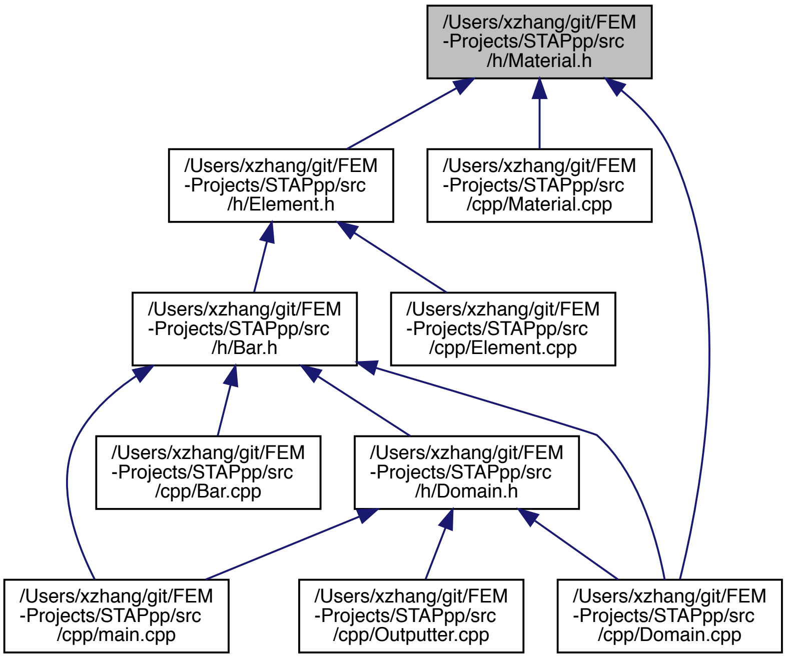
5.16 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h File Reference

```
#include <stddef.h>
#include <iostream>
#include <fstream>
```

Include dependency graph for Material.h:



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


/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Element.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Element.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp

Classes

- class CMaterial

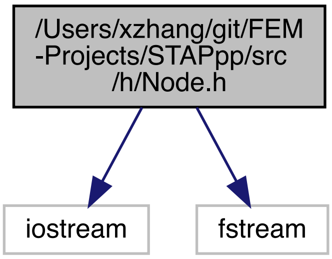
  Material base class which only define one data member.
- class CBarMaterial

  Material class for bar element.

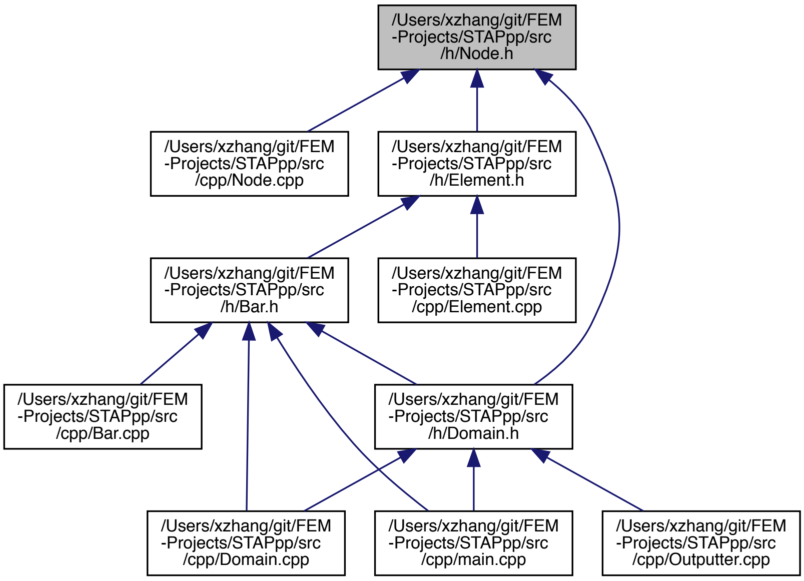
5.17 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Node.h File Reference

```
#include <iostream>
#include <fstream>
```

Include dependency graph for Node.h:



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



Classes

- class CNode

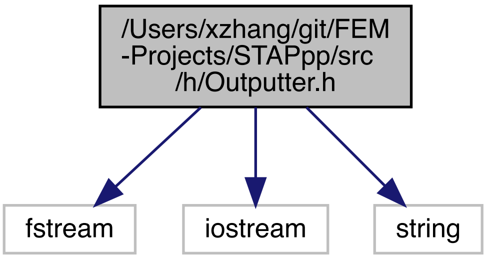
    Node class.

5.18 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Outputter.h File Reference

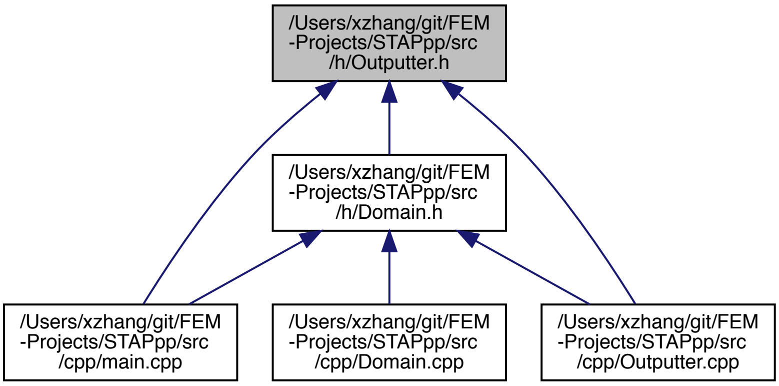
```
#include <fstream>
#include <iostream>
```

```
#include <string>
```

Include dependency graph for Outputter.h:



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



Classes

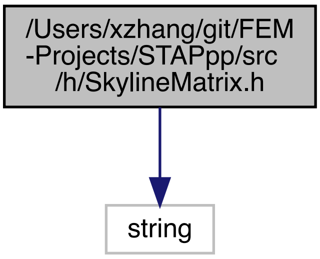
- class COutputter

  Outputer class is used to output results.

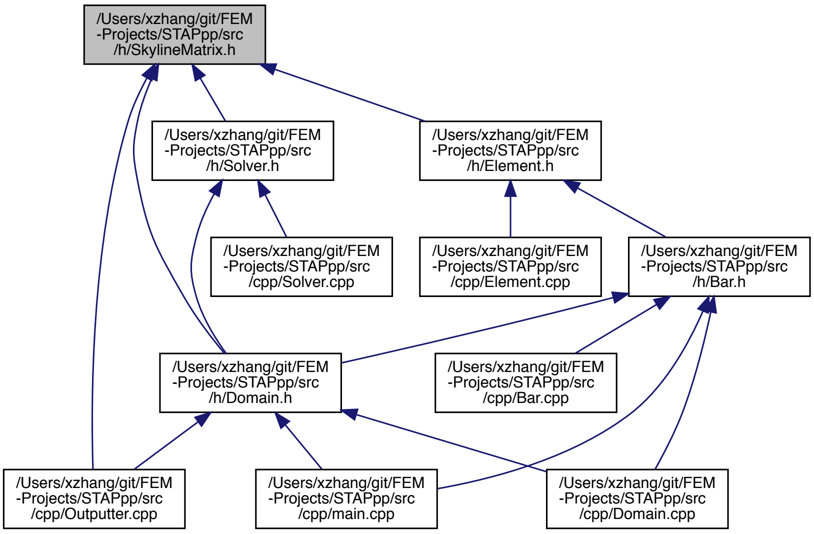
5.19 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/SkylineMatrix.h File Reference

```
#include <string>
```

Include dependency graph for SkylineMatrix.h:



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



Classes

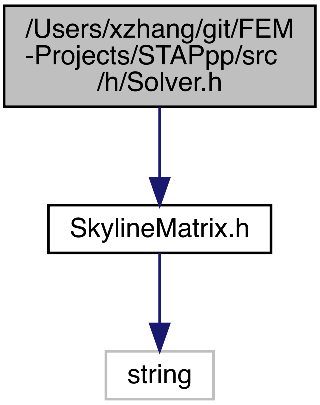
- class CSkylineMatrix< T_ >

    CSkylineMatrix class is used to store the FEM stiffness matrix in skyline storage.

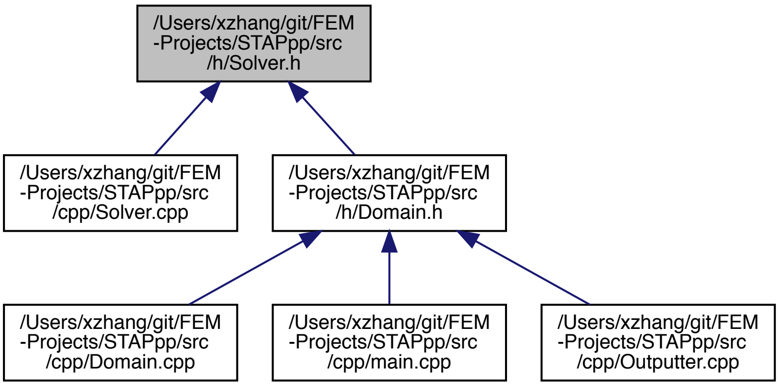
5.20 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Solver.h File Reference

```
#include "SkylineMatrix.h"
```

Include dependency graph for Solver.h:



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



Classes

- class CSolver

  Base class for a solver.
- class CLDLTSolver

  LDLT solver: A in core solver using skyline storage and column reduction scheme.