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

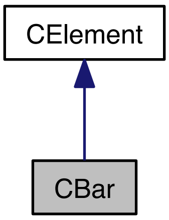
Class Documentation

4.1 CBar Class Reference

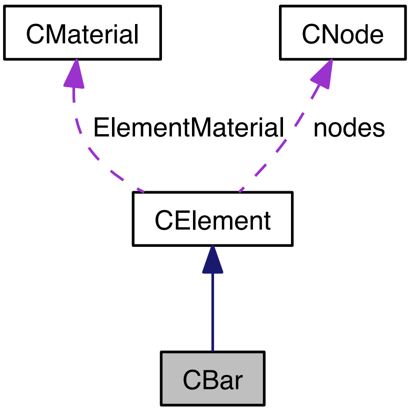
Bar element class.

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

Inheritance diagram for CBar:



Collaboration diagram for CBar:



Public Member Functions

- CBar ()

  Constructor.
- ∼CBar ()

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

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

  Write element data to stream OutputFile.
- 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 Read()

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

Read element data from stream Input.

Implements CElement.

4.1.3.4 SizeOfStiffnessMatrix()

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

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

Implements CElement.

4.1.3.5 Write()

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

Write element data to stream OutputFile.

Implements CElement.

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

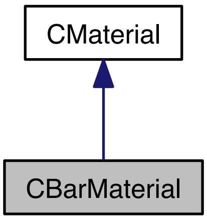
- /Users/xzhang/GitHub/stappp/src/h/Bar.h
- /Users/xzhang/GitHub/stappp/src/cpp/Bar.cpp

4.2 CBarMaterial Class Reference

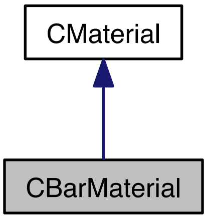
Material class for bar element.

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

Inheritance diagram for CBarMaterial:



Collaboration diagram for CBarMaterial:



Public Member Functions

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

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

Read material data from stream Input.

Implements CMaterial.

4.2.2.2 Write()

```
void CBarMaterial::Write (
             ofstream & OutputFile,
             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:

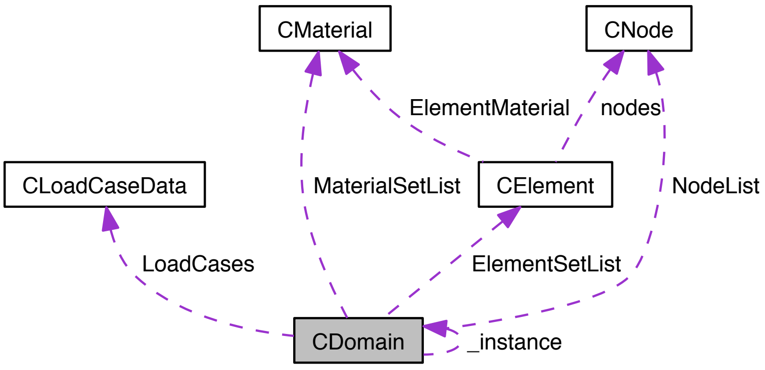
- /Users/xzhang/GitHub/stappp/src/h/Material.h
- /Users/xzhang/GitHub/stappp/src/cpp/Material.cpp

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 (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.
- 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.

- unsigned int ∗ GetColumnHeights ()

  Return column heights.

- double ∗ GetStiffnessMatrix ()

  Return pointer to the banded stiffness matrix.

- unsigned int ∗ GetDiagonalAddress ()

  Return pointer to the array storing the address of diagonal elements.

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.

- 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.
- double ∗ StiffnessMatrix

  Banded stiffness matrix.
- unsigned int ∗ ColumnHeights

  Column heights.
- unsigned int ∗ DiagonalAddress

  Address of diagonal elements in 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.

Allicate 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 GetColumnHeights()

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

Return column heights.

4.3.3.8 GetDiagonalAddress()

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

Return pointer to the array storing the address of diagonal elements.

4.3.3.9 GetDisplacement()

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

Return pointer to the global nodal displacement vector.

4.3.3.10 GetElementSetList()

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

Return element Set List.

4.3.3.11 GetElementTypes()

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

Element type of each group.

4.3.3.12 GetForce()

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

Return pointer to the global nodal force vector.

4.3.3.13 GetLoadCases()

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

Return the list of load cases.

4.3.3.14 GetMaterialSetList()

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

Return material set list.

4.3.3.15 GetMK()

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

Return the maximum half bandwith.

4.3.3.16 GetMODEX()

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

Return solution mode.

4.3.3.17 GetNEQ()

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

Return the total number of equations.

4.3.3.18 GetNLCASE()

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

Return the total number of load cases.

4.3.3.19 GetNLOAD()

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

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

4.3.3.20 GetNodeList()

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

Return the node list.

4.3.3.21 GetNUME()

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

Return the number of elements in each element group.

4.3.3.22 GetNUMEG()

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

Return total number of element groups.

4.3.3.23 GetNUMMAT()

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

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

4.3.3.24 GetNUMNP()

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

Return the total number of nodal points.

4.3.3.25 GetNWK()

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

Return the number of banded global stiffness matrix elements.

4.3.3.26 GetStiffnessMatrix()

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

Return pointer to the banded stiffness matrix.

4.3.3.27 GetTitle()

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

Return the title of problem.

4.3.3.28 Instance()

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

Return pointer to the instance of the Domain class.

4.3.3.29 ReadBarElementData()

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

Read bar element data from the input data file.

4.3.3.30 ReadData()

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

Read domain data from the input data file.

4.3.3.31 ReadElements()

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

Read element data.

4.3.3.32 ReadLoadCases()

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

Read load case data.

4.3.3.33 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 ColumnHeights

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

Column heights.

4.3.4.3 DiagonalAddress

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

Address of diagonal elements in banded stiffness matrix.

4.3.4.4 ElementSetList

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

Element Set List.

ElementSetList[i] - ith element set

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

4.3.4.5 ElementTypes

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

Element type of each group.

4.3.4.6 Force

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

Global nodal force/displacement vector.

4.3.4.7 Input

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

Input file stream for reading data from input data file.

4.3.4.8 LoadCases

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

List of all load cases.

4.3.4.9 MaterialSetList

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

Material set list.

MaterialSetList[i] - ith material set

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

4.3.4.10 MK

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

Maximum half bandwith.

4.3.4.11 MODEX

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

Solution MODEX.

0 : Data check only; 1 : Execution

4.3.4.12 NEQ

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

Total number of equations in the system.

4.3.4.13 NLCASE

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

Number of load cases.

4.3.4.14 NLOAD

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

Number of concentrated loads applied in each load case.

4.3.4.15 NodeList

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

List of all nodes in the domain.

4.3.4.16 NUME

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

Number of elements in each element group.

4.3.4.17 NUMEG

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

Total number of element groups.

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

4.3.4.18 NUMMAT

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

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

4.3.4.19 NUMNP

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

Total number of nodal points.

4.3.4.20 NWK

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

Number of elements in banded global stiffness matrix.

4.3.4.21 StiffnessMatrix

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

Banded stiffness matrix.

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

4.3.4.22 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:

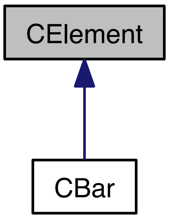
- /Users/xzhang/GitHub/stappp/src/h/Domain.h
- /Users/xzhang/GitHub/stappp/src/cpp/Domain.cpp

4.4 CElement Class Reference

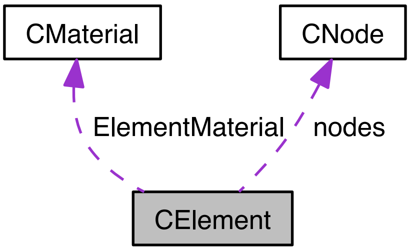
Element base class.

`#include <Element.h>`

Inheritance diagram for CElement:



Collaboration diagram for CElement:



Public Member Functions

- CElement ()

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

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

  Write element data to stream OutputFile.
- void CalculateColumnHeight (unsigned int ∗ColumnHeight)

  Calculate the column height, used with the skyline storage scheme.
- void assembly (double ∗Matrix, double ∗StiffnessMatrix, unsigned int ∗DiagonalAddress)

  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)

Public Attributes

- friend CDomain

Protected Attributes

- int NEN

  Number of nodes per element.
- CNode ∗∗ nodes

  Nodes of the element.
- CMaterial ∗ ElementMaterial

  Material of the element.

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,
            double * StiffnessMatrix,
            unsigned int * DiagonalAddress )
```

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 GetElementMaterial()

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

Return material of the element.

4.4.3.6 GetNodes()

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

Return nodes of the element.

4.4.3.7 Read()

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

Read element data from stream Input.

Implemented in CBar.

4.4.3.8 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.

4.4.3.9 Write()

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

Write element data to stream OutputFile.

Implemented in CBar.

4.4.4 Member Data Documentation

4.4.4.1 CDomain

```
friend CElement::CDomain
```

4.4.4.2 ElementMaterial

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

Material of the element.

Pointer to an element of MaterialSetList[][]

4.4.4.3 NEN

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

Number of nodes per element.

4.4.4.4 nodes

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

Nodes of the element.

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

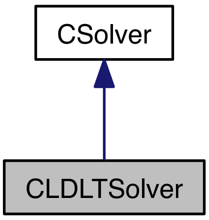
- /Users/xzhang/GitHub/stappp/src/h/Element.h
- /Users/xzhang/GitHub/stappp/src/cpp/Element.cpp

4.5 CLDLTSolver Class Reference

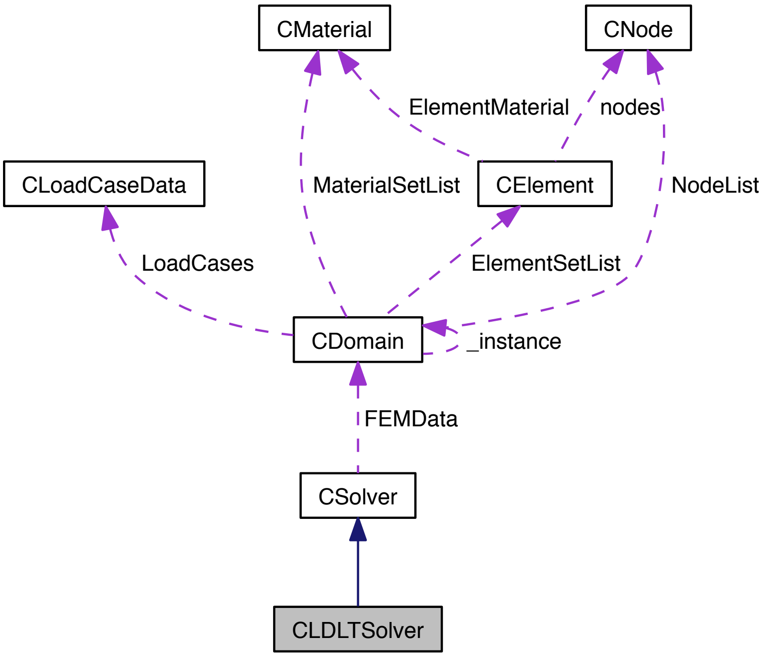
LDLT solver: A in core solver using skyline storage and column reduction scheme.

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

Inheritance diagram for CLDLTSolver:



Collaboration diagram for CLDLTSolver:



Public Member Functions

- CLDLTSolver (CDomain ∗FEMData)

  Constructor.
- void LDLT ()

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

  Reduce right-hand-side load vector and back substitute.
- virtual void Solve ()

  Solve the equilibrium equations.

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 (
            CDomain * FEMData )  [inline]
```

Constructor.

4.5.3 Member Function Documentation

4.5.3.1 BackSubstitution()

```
void CLDLTSolver::BackSubstitution ( )
```

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.

4.5.3.3 Solve()

```
void CLDLTSolver::Solve ( )  [virtual]
```

Solve the equilibrium equations.

Implements CSolver.

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

- /Users/xzhang/GitHub/stappp/src/h/Solver.h
- /Users/xzhang/GitHub/stappp/src/cpp/Solver.cpp

4.6 CLoadCaseData Class Reference

Structure LoadData is used to store load data.

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

Public Member Functions

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

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

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

Structure 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 (
             int num )
```

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

4.6.3.2 Read()

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

Read load case data from stream Input.

4.6.3.3 Write()

```
void CLoadCaseData::Write (
             ofstream & OutputFile,
             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:

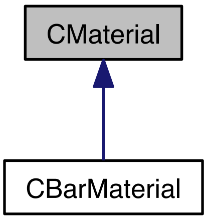
- /Users/xzhang/GitHub/stappp/src/h/LoadCaseData.h
- /Users/xzhang/GitHub/stappp/src/cpp/LoadCaseData.cpp

4.7 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, int mset)=0

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

    Write material data to Stream OutputFile.

Public Attributes

- unsigned int nset

  Number of set.
- double E

  Young's modulus.

4.7.1 Detailed Description

Material base class which only define one data member.

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

4.7.2 Member Function Documentation

4.7.2.1 Read()

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

Read material data from stream Input.

Implemented in CBarMaterial.

4.7.2.2 Write()

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

Write material data to Stream OutputFile.

Implemented in CBarMaterial.

4.7.3 Member Data Documentation

4.7.3.1 E

```
double CMaterial::E
```

Young's modulus.

4.7.3.2 nset

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

Number of set.

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

- /Users/xzhang/GitHub/stappp/src/h/Material.h

4.8 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, int np)

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

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

  Output equation numbers of nodal point to stream OutputFile.
- void WriteNodalDisplacement (ofstream &OutputFile, 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
- 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.

4.8.1 Detailed Description

Node class.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 CNode()

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

Constructor.

4.8.3 Member Function Documentation

4.8.3.1 Read()

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

Read nodal point data from stream Input.

4.8.3.2 Write()

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

Output nodal point data to stream OutputFile.

4.8.3.3 WriteEquationNo()

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

Output equation numbers of nodal point to stream OutputFile.

4.8.3.4 WriteNodalDisplacement()

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

Write nodal displacement.

4.8.4 Member Data Documentation

4.8.4.1 bcode

```
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

4.8.4.2 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

4.8.4.3 NodeNumber

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

Node numer.

4.8.4.4 XYZ

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

x, y and z coordinates of the node

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

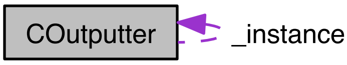
- /Users/xzhang/GitHub/stappp/src/h/Node.h
- /Users/xzhang/GitHub/stappp/src/cpp/Node.cpp

4.9 COutputter Class Reference

Outputer class for outputing 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 (int EleGrp)

  Output bar element data.
- void OutputLoadInfo ()

  Output load data.
- void OutputNodalDisplacement (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.

4.9.1 Detailed Description

Outputer class for outputing results.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 COutputter()

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

Constructor.

4.9.3 Member Function Documentation

4.9.3.1 GetOutputFile()

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

Return pointer to the output file stream.

4.9.3.2 Instance()

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

Return the single instance of the class.

4.9.3.3 OutputElementInfo()

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

Output element data.

4.9.3.4 OutputElementStress()

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

Output element stresses.

4.9.3.5 OutputEquationNumber()

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

Output equation numbers.

4.9.3.6 OutputHeading()

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

Output logo and heading.

4.9.3.7 OutputLoadInfo()

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

Output load data.

4.9.3.8 OutputNodalDisplacement()

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

Output displacement data.

4.9.3.9 OutputNodeInfo()

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

Output nodal point data.

4.9.3.10 OutputTotalSystemData()

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

Print total system data.

4.9.3.11 PrintBarElementData()

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

Output bar element data.

4.9.3.12 PrintTime()

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

Output current time and date.

4.9.4 Member Data Documentation

4.9.4.1 _instance

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

Designed as a single instance class.

4.9.4.2 OutputFile

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

File stream for output.

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

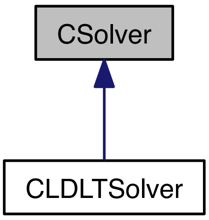
- /Users/xzhang/GitHub/stappp/src/h/Outputter.h
- /Users/xzhang/GitHub/stappp/src/cpp/Outputter.cpp

4.10 CSolver Class Reference

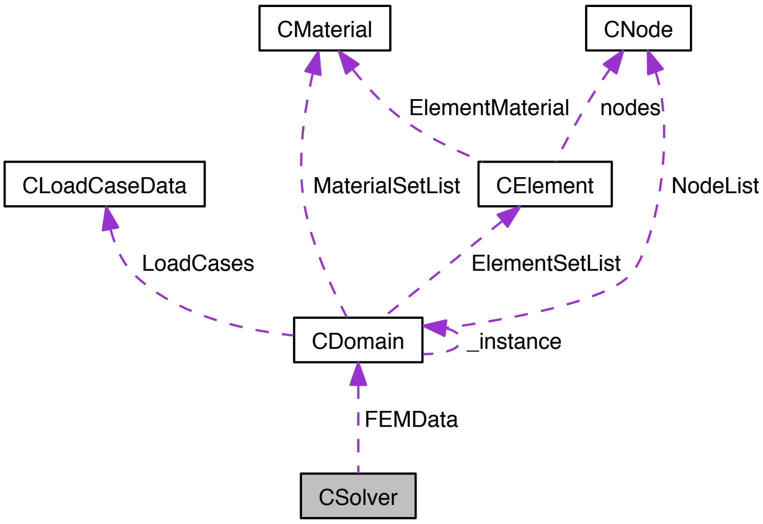
Base class for a solver.

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

Inheritance diagram for CSolver:



Collaboration diagram for CSolver:



Public Member Functions

- CSolver (CDomain ∗FEMData)
- virtual void Solve ()=0

Protected Attributes

- CDomain ∗ FEMData

4.10.1 Detailed Description

Base class for a solver.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 CSolver()

```
CSolver::CSolver (
            CDomain ∗ FEMData )
```

4.10.3 Member Function Documentation

4.10.3.1 Solve()

```
virtual void CSolver::Solve ( )  [pure virtual]
```

Implemented in CLDLTSolver.

4.10.4 Member Data Documentation

4.10.4.1 FEMData

```
CDomain* CSolver::FEMData  [protected]
```

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

- /Users/xzhang/GitHub/stappp/src/h/Solver.h
- /Users/xzhang/GitHub/stappp/src/cpp/Solver.cpp

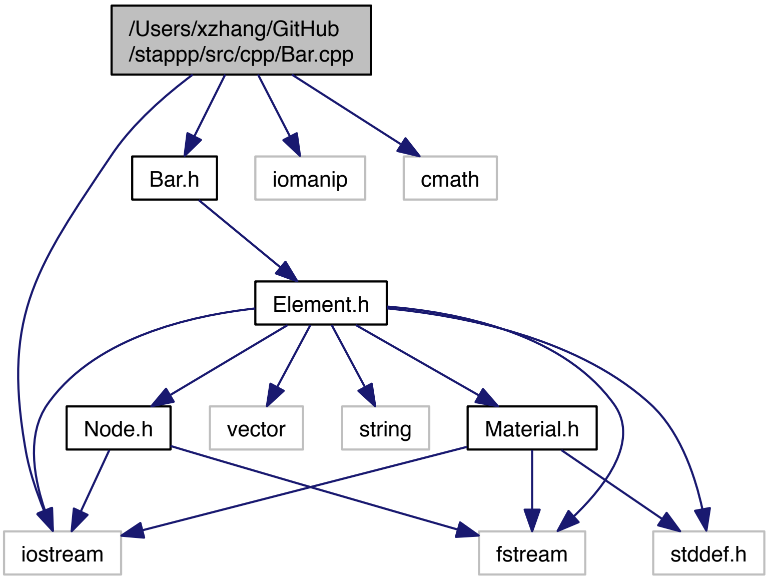
Chapter 5

File Documentation

5.1 /Users/xzhang/GitHub/stappp/src/cpp/Bar.cpp File Reference

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

Include dependency graph for Bar.cpp:

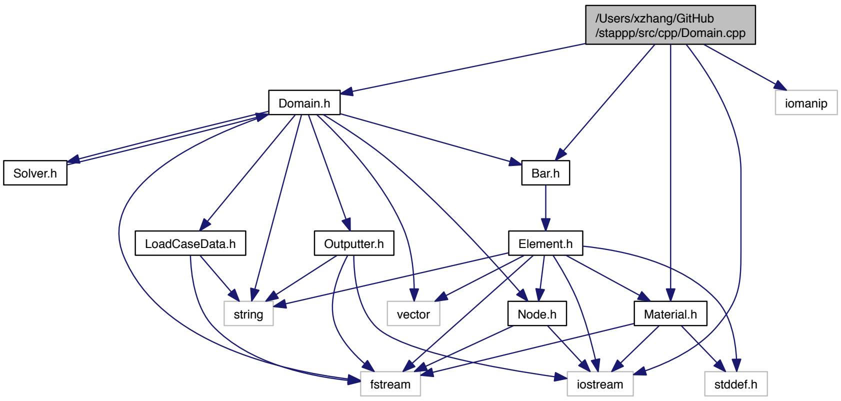


5.2 /Users/xzhang/GitHub/stappp/src/cpp/Domain.cpp File Reference

```
#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, int N)

  Clear an array.

5.2.1 Function Documentation

5.2.1.1 clear()

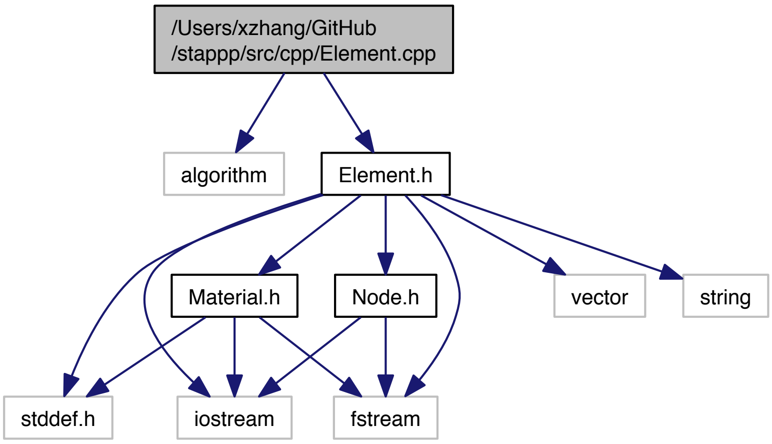
```
template<class type >
void clear (
             type ∗ a,
             int N )
```

Clear an array.

5.3 /Users/xzhang/GitHub/stappp/src/cpp/Element.cpp File Reference

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

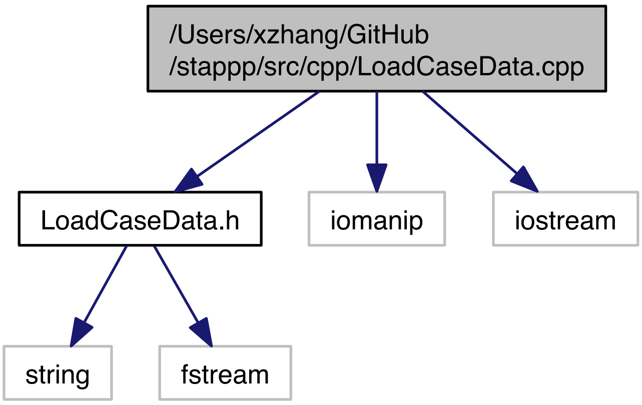
Include dependency graph for Element.cpp:



5.4 /Users/xzhang/GitHub/stappp/src/cpp/LoadCaseData.cpp File Reference

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

Include dependency graph for LoadCaseData.cpp:

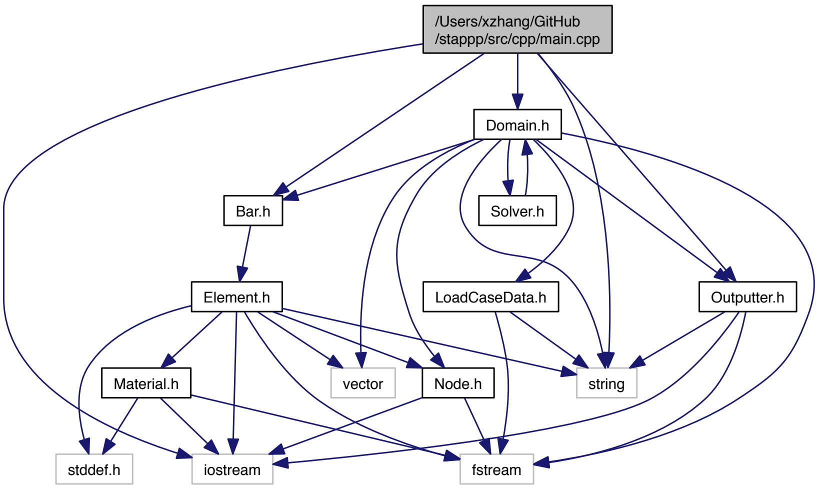


5.5 /Users/xzhang/GitHub/stappp/src/cpp/main.cpp File Reference

```
#include "Domain.h"
#include "Bar.h"
```

```
#include "Outputter.h"
#include <string>
#include <iostream>
```

Include dependency graph for main.cpp:



Functions

- int main (int argc, char ∗argv[ ])

5.5.1 Function Documentation

5.5.1.1 main()

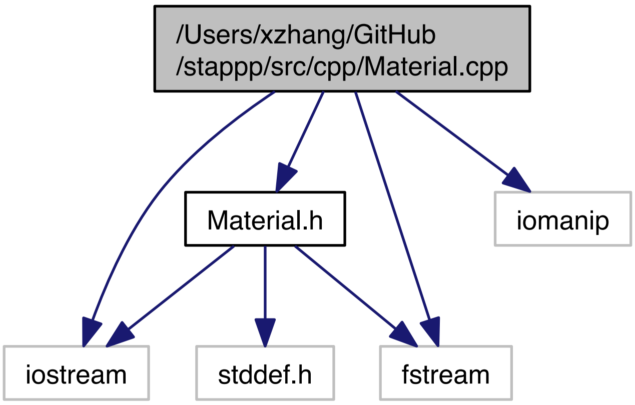
```
int main (
            int argc,
            char * argv[ ] )
```

5.6 /Users/xzhang/GitHub/stappp/src/cpp/Material.cpp File Reference

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

```
#include <iomanip>
```

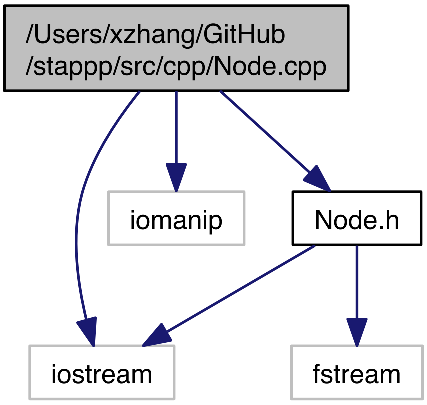
Include dependency graph for Material.cpp:



5.7 /Users/xzhang/GitHub/stappp/src/cpp/Node.cpp File Reference

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

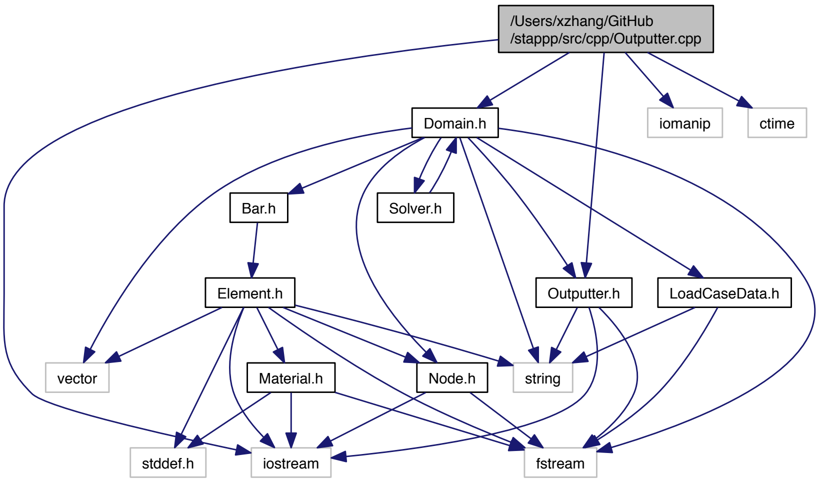
Include dependency graph for Node.cpp:



5.8 /Users/xzhang/GitHub/stappp/src/cpp/Outputter.cpp File Reference

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

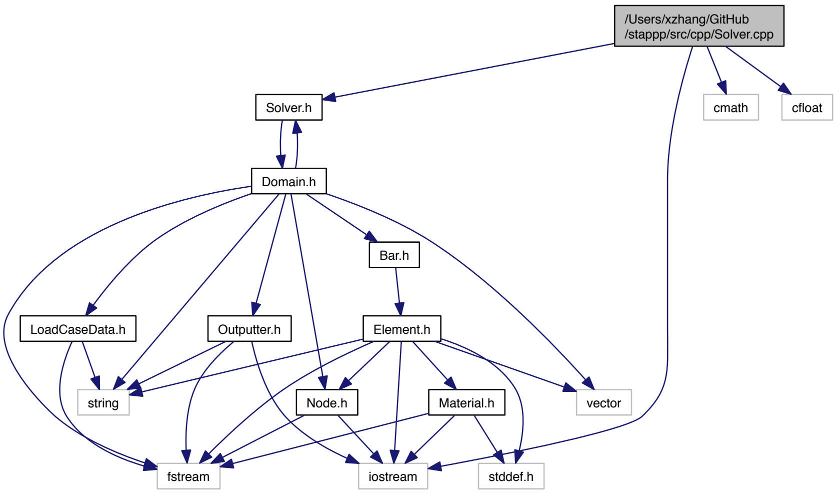
Include dependency graph for Outputter.cpp:



5.9 /Users/xzhang/GitHub/stappp/src/cpp/Solver.cpp File Reference

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

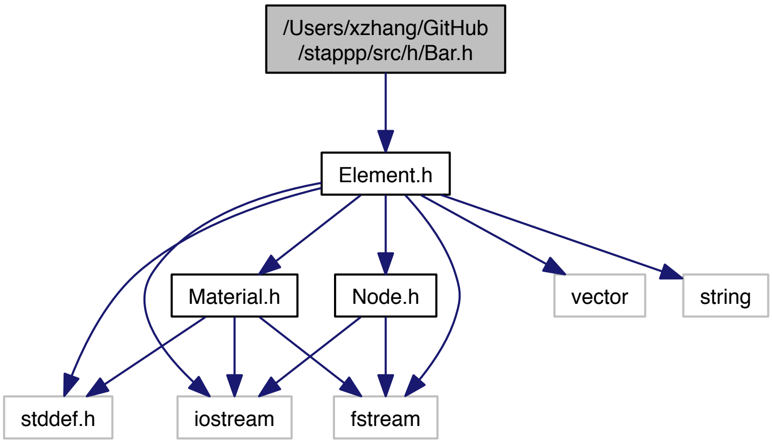
Include dependency graph for Solver.cpp:



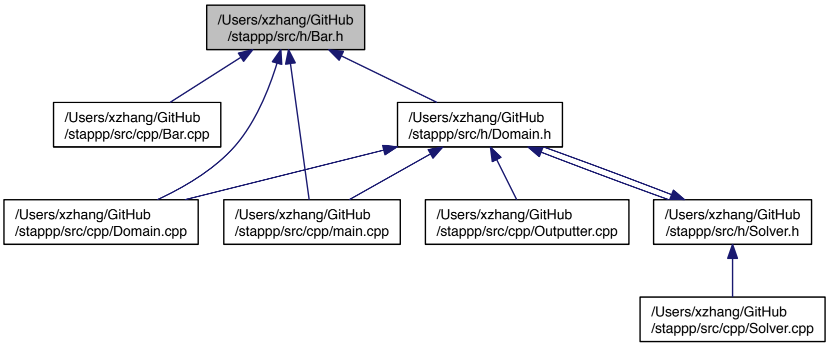
5.10 /Users/xzhang/GitHub/stappp/src/h/Bar.h File Reference

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

Include dependency graph for Bar.h:



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



Classes

- class CBar

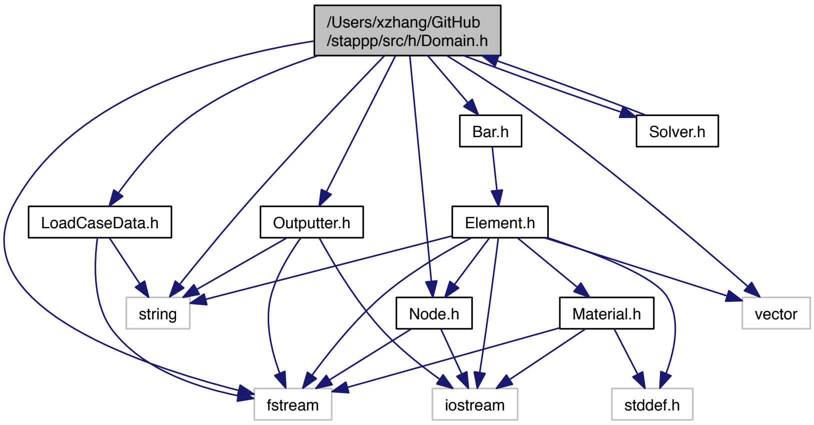
  Bar element class.

5.11 /Users/xzhang/GitHub/stappp/src/h/Domain.h File Reference

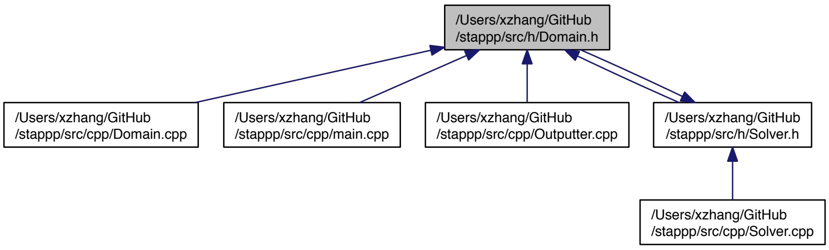
```
#include <string>
#include <fstream>
```

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

Include dependency graph for Domain.h:



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


/Users/xzhang/GitHub /stappp/src/h/Domain.h
/Users/xzhang/GitHub /stappp/src/cpp/Domain.cpp
/Users/xzhang/GitHub /stappp/src/cpp/main.cpp
/Users/xzhang/GitHub /stappp/src/cpp/Outputter.cpp
/Users/xzhang/GitHub /stappp/src/h/Solver.h
/Users/xzhang/GitHub /stappp/src/cpp/Solver.cpp

Classes

- class CDomain

  Domain class : Define the problem domain.

Functions

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

  Clear an array.

5.11.1 Function Documentation

5.11.1.1 clear()

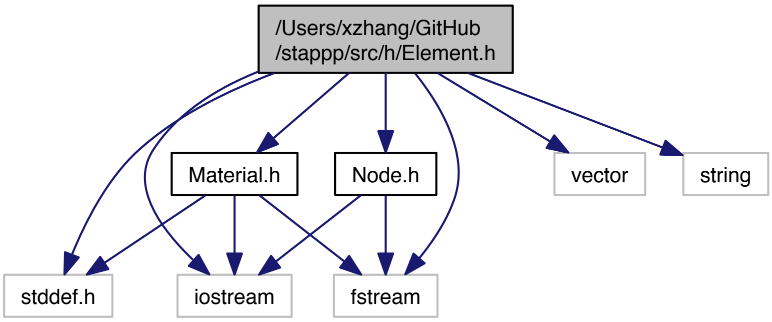
```
template<class type >
void clear (
            type * a,
            int N )
```

Clear an array.

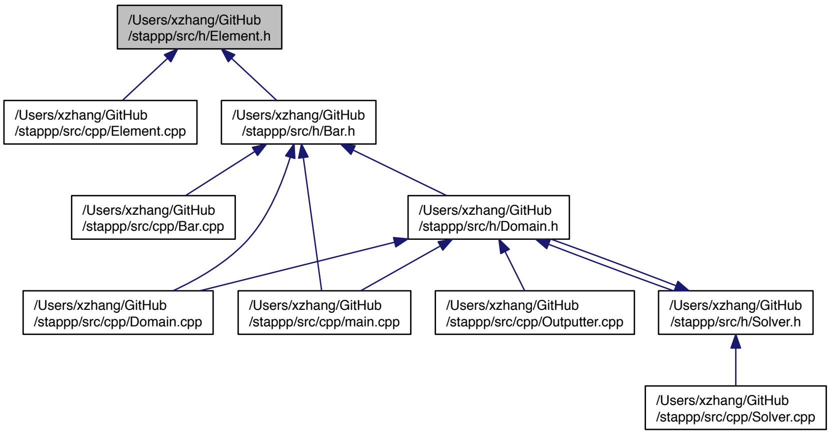
5.12 /Users/xzhang/GitHub/stappp/src/h/Element.h File Reference

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

Include dependency graph for Element.h:



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


/Users/xzhang/GitHub /stappp/src/h/Element.h
/Users/xzhang/GitHub /stappp/src/cpp/Element.cpp
/Users/xzhang/GitHub /stappp/src/h/Bar.h
/Users/xzhang/GitHub /stappp/src/cpp/Bar.cpp
/Users/xzhang/GitHub /stappp/src/h/Domain.h
/Users/xzhang/GitHub /stappp/src/cpp/Domain.cpp
/Users/xzhang/GitHub /stappp/src/cpp/main.cpp
/Users/xzhang/GitHub /stappp/src/cpp/Outputter.cpp
/Users/xzhang/GitHub /stappp/src/h/Solver.h
/Users/xzhang/GitHub /stappp/src/cpp/Solver.cpp

Classes

- class CElement

  Element base class.

Functions

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

  Clear an array.

5.12.1 Function Documentation

5.12.1.1 clear()

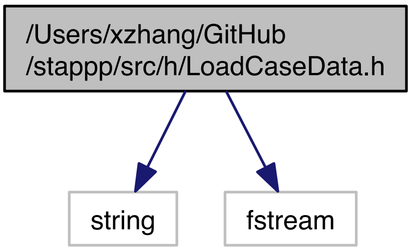
```
template<class type >
void clear (
             type ∗ a,
             int N )
```

Clear an array.

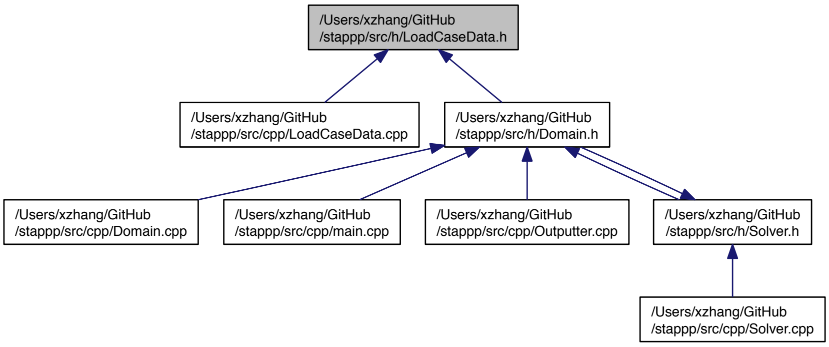
5.13 /Users/xzhang/GitHub/stappp/src/h/LoadCaseData.h File Reference

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

Include dependency graph for LoadCaseData.h:



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



Classes

- class CLoadCaseData

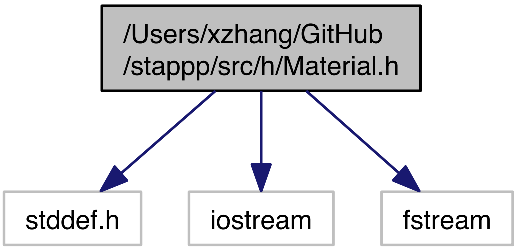
  Structure LoadData is used to store load data.

5.14 /Users/xzhang/GitHub/stappp/src/h/Material.h File Reference

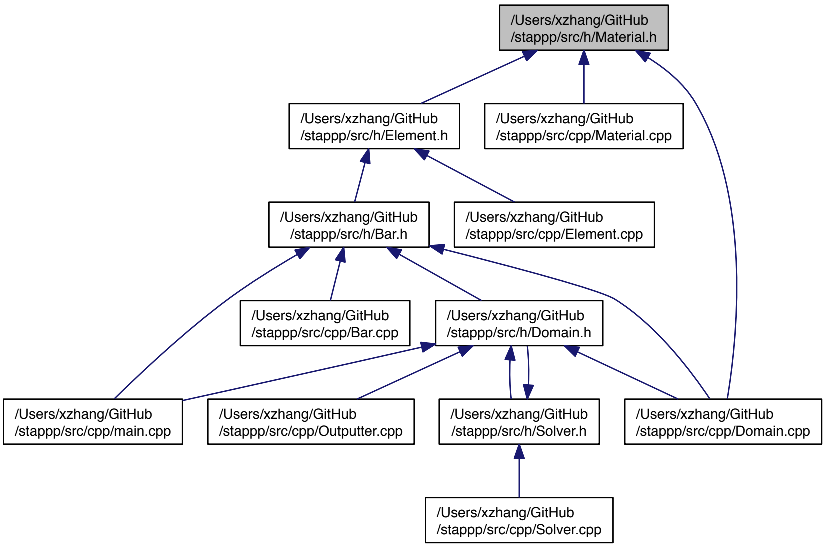
```
#include <stddef.h>
#include <iostream>
```

```
#include <fstream>
```

Include dependency graph for Material.h:



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



Classes

- class CMaterial

    Material base class which only define one data member.

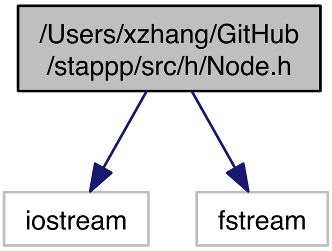
- class CBarMaterial

    Material class for bar element.

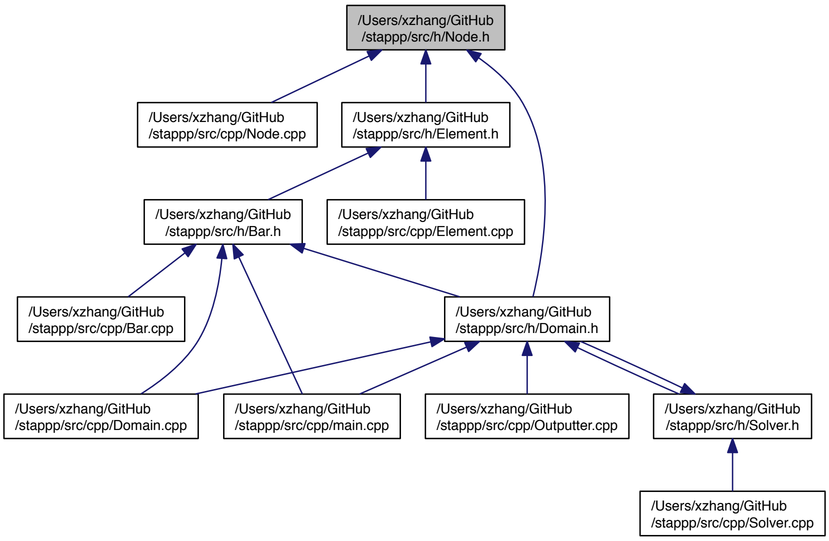
5.15 /Users/xzhang/GitHub/stappp/src/h/Node.h File Reference

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

Include dependency graph for Node.h:



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



Classes

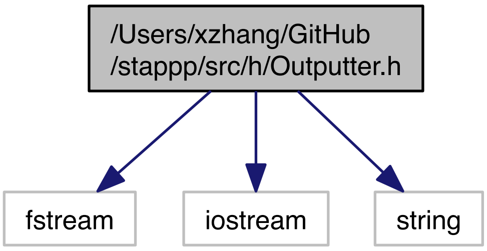
- class CNode

    Node class.

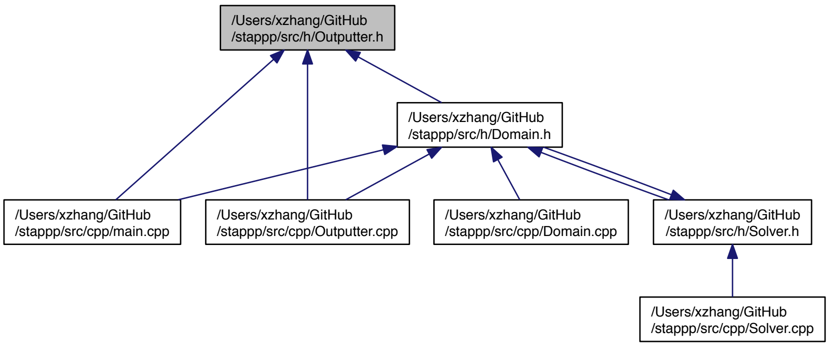
5.16 /Users/xzhang/GitHub/stappp/src/h/Outputter.h File Reference

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

Include dependency graph for Outputter.h:



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



Classes

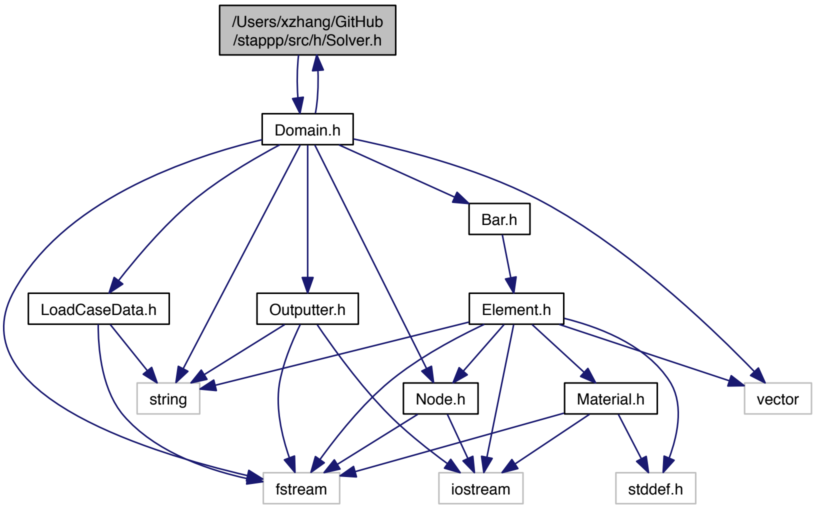
- class COutputter

  Outputer class for outputing results.

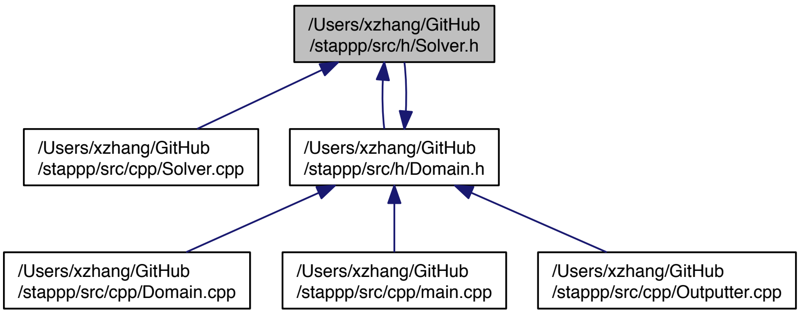
5.17 /Users/xzhang/GitHub/stappp/src/h/Solver.h File Reference

```
#include "Domain.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.

Index