

STAPpp

1.10

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	CBar Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	CBar()	9
4.1.2.2	~CBar()	9
4.1.3	Member Function Documentation	9
4.1.3.1	ElementStiffness()	9
4.1.3.2	ElementStress()	9
4.1.3.3	GenerateLocationMatrix()	9
4.1.3.4	Read()	10
4.1.3.5	SizeOfStiffnessMatrix()	10
4.1.3.6	Write()	10
4.2	CBarMaterial Class Reference	11
4.2.1	Detailed Description	12
4.2.2	Member Function Documentation	12

4.2.2.1	Read()	12
4.2.2.2	Write()	12
4.2.3	Member Data Documentation	12
4.2.3.1	Area	12
4.3	CDomain Class Reference	13
4.3.1	Detailed Description	15
4.3.2	Constructor & Destructor Documentation	15
4.3.2.1	CDomain()	15
4.3.2.2	~CDomain()	15
4.3.3	Member Function Documentation	15
4.3.3.1	AllocateMatrices()	16
4.3.3.2	AssembleForce()	16
4.3.3.3	AssembleStiffnessMatrix()	16
4.3.3.4	CalculateColumnHeights()	16
4.3.3.5	CalculateEquationNumber()	16
4.3.3.6	GetDisplacement()	16
4.3.3.7	GetEleGrpList()	17
4.3.3.8	GetForce()	17
4.3.3.9	GetLoadCases()	17
4.3.3.10	GetMODEX()	17
4.3.3.11	GetNEQ()	17
4.3.3.12	GetNLCASE()	17
4.3.3.13	GetNLOAD()	18
4.3.3.14	GetNodeList()	18
4.3.3.15	GetNUMEG()	18
4.3.3.16	GetNUMNP()	18
4.3.3.17	GetStiffnessMatrix()	18
4.3.3.18	GetTitle()	18
4.3.3.19	Instance()	19
4.3.3.20	ReadData()	19

4.3.3.21	ReadElements()	19
4.3.3.22	ReadLoadCases()	19
4.3.3.23	ReadNodalPoints()	19
4.3.4	Member Data Documentation	19
4.3.4.1	_instance	20
4.3.4.2	EleGrpList	20
4.3.4.3	Force	20
4.3.4.4	Input	20
4.3.4.5	LoadCases	20
4.3.4.6	MODEX	20
4.3.4.7	NEQ	21
4.3.4.8	NLCASE	21
4.3.4.9	NLOAD	21
4.3.4.10	NodeList	21
4.3.4.11	NUMEG	21
4.3.4.12	NUMNP	21
4.3.4.13	StiffnessMatrix	22
4.3.4.14	Title	22
4.4	CElement Class Reference	22
4.4.1	Detailed Description	23
4.4.2	Constructor & Destructor Documentation	23
4.4.2.1	CElement()	24
4.4.2.2	~CElement()	24
4.4.3	Member Function Documentation	24
4.4.3.1	ElementStiffness()	24
4.4.3.2	ElementStress()	24
4.4.3.3	GenerateLocationMatrix()	24
4.4.3.4	GetElementMaterial()	25
4.4.3.5	GetLocationMatrix()	25
4.4.3.6	GetND()	25

4.4.3.7	GetNodes()	25
4.4.3.8	Read()	25
4.4.3.9	SizeOfStiffnessMatrix()	26
4.4.3.10	Write()	26
4.4.4	Member Data Documentation	26
4.4.4.1	ElementMaterial_	26
4.4.4.2	LocationMatrix_	26
4.4.4.3	ND_	26
4.4.4.4	NEN_	27
4.4.4.5	nodes_	27
4.5	CElementGroup Class Reference	27
4.5.1	Detailed Description	29
4.5.2	Constructor & Destructor Documentation	29
4.5.2.1	CElementGroup()	29
4.5.2.2	~CElementGroup()	29
4.5.3	Member Function Documentation	29
4.5.3.1	AllocateElements()	29
4.5.3.2	AllocateMaterials()	29
4.5.3.3	CalculateMemberSize()	30
4.5.3.4	GetElementType()	30
4.5.3.5	GetMaterial()	30
4.5.3.6	GetNUME()	30
4.5.3.7	GetNUMMAT()	30
4.5.3.8	operator[]()	30
4.5.3.9	Read()	31
4.5.3.10	ReadElementData()	31
4.5.4	Member Data Documentation	31
4.5.4.1	ElementList_	31
4.5.4.2	ElementSize_	31
4.5.4.3	ElementType_	31

4.5.4.4	MaterialList_	32
4.5.4.5	MaterialSize_	32
4.5.4.6	NodeList_	32
4.5.4.7	NUME_	32
4.5.4.8	NUMMAT_	32
4.6	CLDLTSolver Class Reference	33
4.6.1	Detailed Description	34
4.6.2	Constructor & Destructor Documentation	34
4.6.2.1	CLDLTSolver()	34
4.6.3	Member Function Documentation	34
4.6.3.1	BackSubstitution()	34
4.6.3.2	LDLT()	34
4.7	CLoadCaseData Class Reference	34
4.7.1	Detailed Description	35
4.7.2	Constructor & Destructor Documentation	35
4.7.2.1	CLoadCaseData()	35
4.7.2.2	~CLoadCaseData()	35
4.7.3	Member Function Documentation	35
4.7.3.1	Allocate()	36
4.7.3.2	Read()	36
4.7.3.3	Write()	36
4.7.4	Member Data Documentation	36
4.7.4.1	dof	36
4.7.4.2	load	36
4.7.4.3	nloads	37
4.7.4.4	node	37
4.8	Clock Class Reference	37
4.8.1	Detailed Description	38
4.8.2	Constructor & Destructor Documentation	38
4.8.2.1	Clock()	38

4.8.3	Member Function Documentation	38
4.8.3.1	Clear()	38
4.8.3.2	ElapsedTime()	38
4.8.3.3	Resume()	38
4.8.3.4	Start()	39
4.8.3.5	Stop()	39
4.8.4	Member Data Documentation	39
4.8.4.1	ct_	39
4.8.4.2	st0_	39
4.8.4.3	st1_	39
4.8.4.4	t0_	39
4.8.4.5	t1_	40
4.9	CMaterial Class Reference	40
4.9.1	Detailed Description	41
4.9.2	Constructor & Destructor Documentation	41
4.9.2.1	~CMaterial()	41
4.9.3	Member Function Documentation	41
4.9.3.1	Read()	41
4.9.3.2	Write()	41
4.9.4	Member Data Documentation	41
4.9.4.1	E	42
4.9.4.2	nset	42
4.10	CNode Class Reference	42
4.10.1	Detailed Description	43
4.10.2	Constructor & Destructor Documentation	43
4.10.2.1	CNode()	43
4.10.3	Member Function Documentation	43
4.10.3.1	Read()	43
4.10.3.2	Write()	43
4.10.3.3	WriteEquationNo()	44

4.10.3.4	WriteNodalDisplacement()	44
4.10.4	Member Data Documentation	44
4.10.4.1	bcode	44
4.10.4.2	NDF	44
4.10.4.3	NodeNumber	44
4.10.4.4	XYZ	45
4.11	COutputter Class Reference	45
4.11.1	Detailed Description	46
4.11.2	Member Typedef Documentation	46
4.11.2.1	CharOstream	46
4.11.3	Constructor & Destructor Documentation	47
4.11.3.1	COutputter()	47
4.11.4	Member Function Documentation	47
4.11.4.1	GetOutputFile()	47
4.11.4.2	Instance()	47
4.11.4.3	operator<<() [1/2]	47
4.11.4.4	operator<<() [2/2]	47
4.11.4.5	OutputElementInfo()	48
4.11.4.6	OutputElementStress()	48
4.11.4.7	OutputEquationNumber()	48
4.11.4.8	OutputHeading()	48
4.11.4.9	OutputLoadInfo()	48
4.11.4.10	OutputNodalDisplacement()	48
4.11.4.11	OutputNodeInfo()	49
4.11.4.12	OutputTotalSystemData()	49
4.11.4.13	PrintBarElementData()	49
4.11.4.14	PrintTime()	49
4.11.5	Member Data Documentation	49
4.11.5.1	_instance	49
4.11.5.2	OutputFile	50

4.12 CSkylineMatrix< T_ > Class Template Reference	50
4.12.1 Detailed Description	51
4.12.2 Constructor & Destructor Documentation	51
4.12.2.1 CSkylineMatrix() [1/2]	51
4.12.2.2 CSkylineMatrix() [2/2]	52
4.12.2.3 ~CSkylineMatrix()	52
4.12.3 Member Function Documentation	52
4.12.3.1 Allocate()	52
4.12.3.2 Assembly()	52
4.12.3.3 CalculateColumnHeight()	52
4.12.3.4 CalculateDiagnoalAddress()	53
4.12.3.5 CalculateMaximumHalfBandwidth()	53
4.12.3.6 dim()	53
4.12.3.7 GetColumnHeights()	53
4.12.3.8 GetDiagonalAddress()	53
4.12.3.9 GetMaximumHalfBandwidth()	53
4.12.3.10 operator>() [1/2]	54
4.12.3.11 operator>() [2/2]	54
4.12.3.12 size()	54
4.12.4 Member Data Documentation	54
4.12.4.1 ColumnHeights_	54
4.12.4.2 data_	54
4.12.4.3 DiagonalAddress_	55
4.12.4.4 MK_	55
4.12.4.5 NEQ_	55
4.12.4.6 NWK_	55
4.13 CSolver Class Reference	56
4.13.1 Detailed Description	56
4.13.2 Constructor & Destructor Documentation	57
4.13.2.1 CSolver()	57
4.13.3 Member Data Documentation	57
4.13.3.1 K	57

5	File Documentation	59
5.1	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp File Reference	59
5.2	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Clock.cpp File Reference	60
5.3	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp File Reference	60
5.3.1	Function Documentation	61
5.3.1.1	clear()	61
5.4	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/ElementGroup.cpp File Reference	62
5.5	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/LoadCaseData.cpp File Reference	62
5.6	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp File Reference	63
5.6.1	Function Documentation	64
5.6.1.1	main()	64
5.7	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp File Reference	64
5.8	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Node.cpp File Reference	65
5.9	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp File Reference	65
5.10	/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Solver.cpp File Reference	66
5.11	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h File Reference	67
5.12	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h File Reference	68
5.13	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h File Reference	69
5.13.1	Function Documentation	70
5.13.1.1	clear()	70
5.14	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Element.h File Reference	70
5.14.1	Function Documentation	71
5.14.1.1	clear()	71
5.15	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/ElementGroup.h File Reference	72
5.15.1	Enumeration Type Documentation	73
5.15.1.1	ElementTypes	73
5.16	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/LoadCaseData.h File Reference	74
5.17	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h File Reference	75
5.18	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Node.h File Reference	76
5.19	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Outputter.h File Reference	77
5.20	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/SkylineMatrix.h File Reference	78
5.21	/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Solver.h File Reference	79
	Index	81

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

CDomain	13
CElement	22
CBar	7
CElementGroup	27
CLoadCaseData	34
Clock	37
CMaterial	40
CBarMaterial	11
CNode	42
COutputter	45
CSkylineMatrix< T_ >	50
CSkylineMatrix< double >	50
CSolver	56
CLDLTSolver	33

Chapter 2

Class Index

2.1 Class List

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

CBar	Bar element class	7
CBarMaterial	Material class for bar element	11
CDomain	Domain class : Define the problem domain	13
CElement	Element base class	22
CElementGroup	Element group class	27
CLDLTSolver	LDLT solver: A in core solver using skyline storage and column reduction scheme	33
CLoadCaseData	Class LoadData is used to store load data	34
Clock	Clock class for timing	37
CMaterial	Material base class which only define one data member	40
CNode	Node class	42
COutputter	Outputter class is used to output results	45
CSkylineMatrix< T_ >	CSkylineMatrix class is used to store the FEM stiffness matrix in skyline storage	50
CSolver	Base class for a solver	56

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp	59
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Clock.cpp	60
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp	60
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/ElementGroup.cpp	62
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/LoadCaseData.cpp	62
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp	63
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp	64
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Node.cpp	65
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp	65
/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Solver.cpp	66
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h	67
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h	68
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h	69
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Element.h	70
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/ElementGroup.h	72
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/LoadCaseData.h	74
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h	75
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Node.h	76
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Outputter.h	77
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/SkylineMatrix.h	78
/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Solver.h	79

Chapter 4

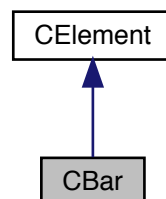
Class Documentation

4.1 CBar Class Reference

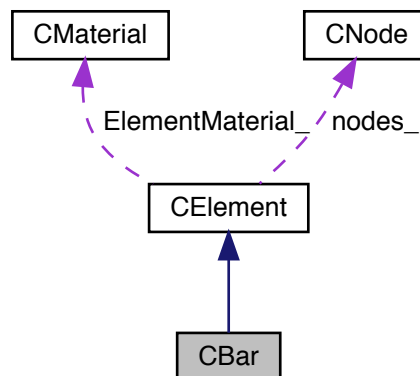
Bar element class.

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

Inheritance diagram for CBar:



Collaboration diagram for CBar:



Public Member Functions

- [CBar](#) ()
Constructor.
- [~CBar](#) ()
Destructor.
- virtual bool [Read](#) (ifstream &Input, unsigned int Ele, [CMaterial](#) *MaterialSets, [CNode](#) *NodeList)
Read element data from stream Input.
- virtual void [Write](#) ([COutputter](#) &output, unsigned int Ele)
Write element data to stream.
- 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 (
    COutputter & output,
    unsigned int Ele ) [virtual]
```

Write element data to stream.

Implements [CElement](#).

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

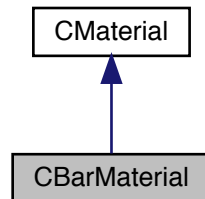
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h](#)
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.cpp](#)

4.2 CBarMaterial Class Reference

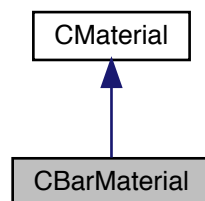
Material class for bar element.

```
#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](#) (COutputter &output, unsigned int mset)
Write material data to Stream.

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 (
    COutputter & output,
    unsigned int mset ) [virtual]
```

Write material data to Stream.

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:

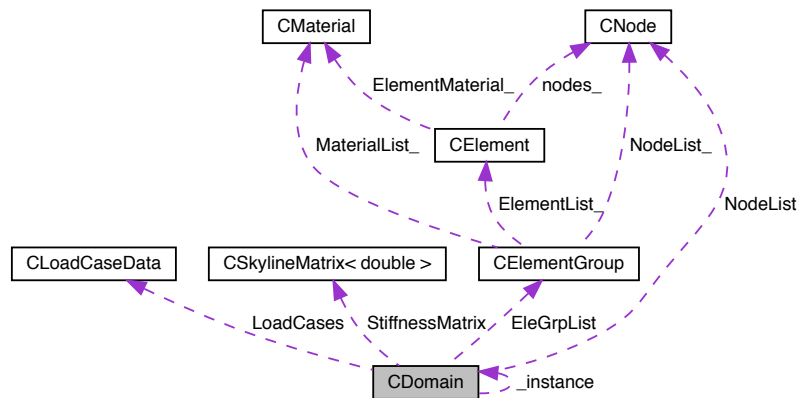
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Material.h](#)
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp](#)

4.3 CDomain Class Reference

Domain class : Define the problem domain.

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

Collaboration diagram for CDomain:



Public Member Functions

- **CDomain** ()
Constructor.
- **~CDomain** ()
Destructor.
- 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.
- void **CalculateEquationNumber** ()
Calculate global equation numbers corresponding to every degree of freedom of each node.
- void **CalculateColumnHeights** ()
Calculate column heights.
- void **AllocateMatrices** ()
Allocate storage for matrices.
- void **AssembleStiffnessMatrix** ()
Assemble the banded global 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.
- [CNode](#) * [GetNodeList](#) ()
Return the node list.
- unsigned int [GetNUMEG](#) ()
Return total number of element groups.
- [CElementGroup](#) * [GetEleGrpList](#) ()
Return element group 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 output.
- 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.
- [CElementGroup](#) * [EleGrpList](#)
Element group 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.

- [CSkylineMatrix](#)< double > * [StiffnessMatrix](#)

Banded stiffness matrix.

- double * [Force](#)

Global nodal force/displacement vector.

Static Private Attributes

- static [CDomain](#) * [_instance](#) = nullptr

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 ( )
```

Destructor.

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 global stiffness matrix.

4.3.3.4 CalculateColumnHeights()

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

Calculate column heights.

4.3.3.5 CalculateEquationNumber()

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

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

4.3.3.6 GetDisplacement()

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

Return pointer to the global nodal displacement vector.

4.3.3.7 GetEleGrpList()

```
CElementGroup* CDomain::GetEleGrpList ( ) [inline]
```

Return element group list.

4.3.3.8 GetForce()

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

Return pointer to the global nodal force vector.

4.3.3.9 GetLoadCases()

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

Return the list of load cases.

4.3.3.10 GetMODEX()

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

Return solution mode.

4.3.3.11 GetNEQ()

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

Return the total number of equations.

4.3.3.12 GetNLCASE()

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

Return the total number of load cases.

4.3.3.13 GetNLOAD()

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

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

4.3.3.14 GetNodeList()

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

Return the node list.

4.3.3.15 GetNUMEG()

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

Return total number of element groups.

4.3.3.16 GetNUMNP()

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

Return the total number of nodal points.

4.3.3.17 GetStiffnessMatrix()

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

Return pointer to the banded stiffness matrix.

4.3.3.18 GetTitle()

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

Return the title of problem.

4.3.3.19 Instance()

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

Return pointer to the instance of the Domain class.

4.3.3.20 ReadData()

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

Read domain data from the input data file.

4.3.3.21 ReadElements()

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

Read element data.

4.3.3.22 ReadLoadCases()

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

Read load case data.

4.3.3.23 ReadNodalPoints()

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

Read nodal point data.

4.3.4 Member Data Documentation

4.3.4.1 `_instance`

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

The instance of the Domain class.

4.3.4.2 `EleGrpList`

```
CElementGroup* CDomain::EleGrpList [private]
```

Element group list.

4.3.4.3 `Force`

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

Global nodal force/displacement vector.

4.3.4.4 `Input`

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

Input file stream for reading data from input data file.

4.3.4.5 `LoadCases`

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

List of all load cases.

4.3.4.6 `MODEX`

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

Solution MODEX.

0 : Data check only; 1 : Execution

4.3.4.7 NEQ

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

Total number of equations in the system.

4.3.4.8 NLCASE

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

Number of load cases.

4.3.4.9 NLOAD

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

Number of concentrated loads applied in each load case.

4.3.4.10 NodeList

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

List of all nodes in the domain.

4.3.4.11 NUMEG

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

Total number of element groups.

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

4.3.4.12 NUMNP

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

Total number of nodal points.

4.3.4.13 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.

4.3.4.14 Title

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

Heading information for use in labeling the output.

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

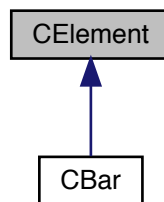
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Domain.h](#)
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp](#)

4.4 CElement Class Reference

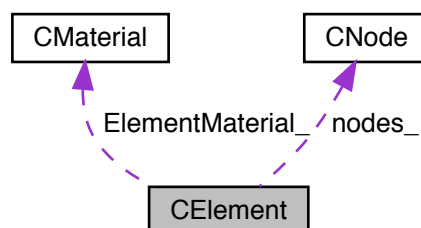
Element base class.

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

Inheritance diagram for CElement:



Collaboration diagram for CElement:



Public Member Functions

- [CElement](#) ()
Constructor.
- virtual [~CElement](#) ()
Virtual destructor.
- virtual bool [Read](#) (ifstream &Input, unsigned int Ele, [CMaterial](#) *MaterialSets, [CNode](#) *NodeList)=0
Read element data from stream Input.
- virtual void [Write](#) ([COutputter](#) &output, unsigned int Ele)=0
Write element data to stream.
- virtual void [GenerateLocationMatrix](#) ()=0
Generate location matrix: the global equation number that corresponding to each DOF of the element.
- virtual void [ElementStiffness](#) (double *stiffness)=0
Calculate element stiffness matrix (Upper triangular matrix, stored as an array column by column)
- 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.
- unsigned int * [GetLocationMatrix](#) ()
Return the Location Matrix of the element.
- unsigned int [GetND](#) ()
Return the dimension of the location matrix.
- 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.

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.

4.4.2.2 ~CElement()

```
virtual CElement::~~CElement ( ) [inline], [virtual]
```

Virtual destructor.

4.4.3 Member Function Documentation

4.4.3.1 ElementStiffness()

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

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

Implemented in [CBar](#).

4.4.3.2 ElementStress()

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

Calculate element stress.

Implemented in [CBar](#).

4.4.3.3 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](#).

4.4.3.4 GetElementMaterial()

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

Return material of the element.

4.4.3.5 GetLocationMatrix()

```
unsigned int* CElement::GetLocationMatrix ( ) [inline]
```

Return the Location Matrix of the element.

4.4.3.6 GetND()

```
unsigned int CElement::GetND ( ) [inline]
```

Return the dimension of the location matrix.

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 (
    COutputter & output,
    unsigned int Ele ) [pure virtual]
```

Write element data to stream.

Implemented in [CBar](#).

4.4.4 Member Data Documentation

4.4.4.1 ElementMaterial_

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

Material of the element.

Pointer to an element of MaterialSetList[]

4.4.4.2 LocationMatrix_

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

Location Matrix of the element.

4.4.4.3 ND_

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

Dimension of the location matrix.

4.4.4.4 NEN_

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

Number of nodes per element.

4.4.4.5 nodes_

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

Nodes of the element.

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

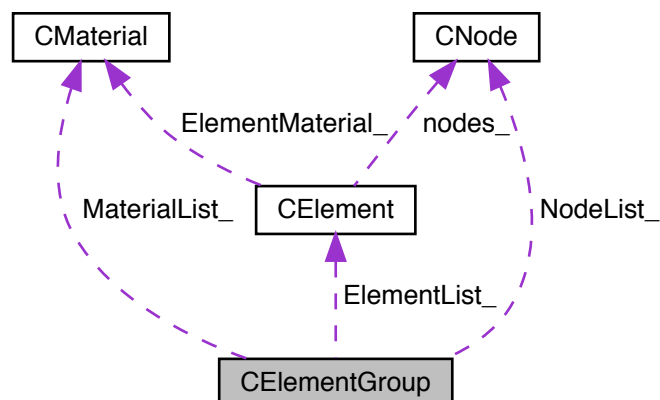
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/[Element.h](#)

4.5 CElementGroup Class Reference

Element group class.

```
#include <ElementGroup.h>
```

Collaboration diagram for CElementGroup:



Public Member Functions

- [CElementGroup](#) ()
Constructor.
- [~CElementGroup](#) ()
Deconstructor.
- bool [Read](#) (ifstream &Input)
Read element group data from stream Input.
- void [CalculateMemberSize](#) ()
Calculate the size of the derived element class and material class.
- void [AllocateElements](#) (std::size_t size)
Allocate array of derived elements.
- void [AllocateMaterials](#) (std::size_t size)
Allocate array of derived materials.
- bool [ReadElementData](#) (ifstream &Input)
Read element data from the input data file.
- [ElementType](#) [GetElementType](#) ()
Return element type of this group.
- unsigned int [GetNUME](#) ()
Return the number of elements in the group.
- [CElement](#) & [operator\[\]](#) (unsigned int i)
- [CMaterial](#) & [GetMaterial](#) (unsigned int index)
Return the index-th material in this group.
- unsigned int [GetNUMMAT](#) ()
Return the number of material/section property setss in this element group.

Private Attributes

- [ElementType](#) [ElementType_](#)
Element type of this group.
- std::size_t [ElementSize_](#)
Size of an Element object in this group.
- unsigned int [NUME_](#)
Number of elements in this group.
- [CElement](#) * [ElementList_](#)
Element List in this group.
- unsigned int [NUMMAT_](#)
Number of material/section property sets in this group.
- [CMaterial](#) * [MaterialList_](#)
Material list in this group.
- std::size_t [MaterialSize_](#)
Size of an Material object in this group.

Static Private Attributes

- static [CNode](#) * [NodeList_](#) = nullptr
List of all nodes in the domain, obtained from [CDomain](#) object.

4.5.1 Detailed Description

Element group class.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 CElementGroup()

```
CElementGroup::CElementGroup ( )
```

Constructor.

4.5.2.2 ~CElementGroup()

```
CElementGroup::~~CElementGroup ( )
```

Deconstructor.

4.5.3 Member Function Documentation

4.5.3.1 AllocateElements()

```
void CElementGroup::AllocateElements (
    std::size_t size )
```

Allocate array of derived elements.

4.5.3.2 AllocateMaterials()

```
void CElementGroup::AllocateMaterials (
    std::size_t size )
```

Allocate array of derived materials.

4.5.3.3 CalculateMemberSize()

```
void CElementGroup::CalculateMemberSize ( )
```

Calculate the size of the derived element class and material class.

Calculate the size of the derived element and material class.

4.5.3.4 GetElementType()

```
ElementTypes CElementGroup::GetElementType ( ) [inline]
```

Return element type of this group.

4.5.3.5 GetMaterial()

```
CMaterial & CElementGroup::GetMaterial (
    unsigned int index )
```

Return the index-th material in this group.

Return index-th material in this element group.

4.5.3.6 GetNUME()

```
unsigned int CElementGroup::GetNUME ( ) [inline]
```

Return the number of elements in the group.

4.5.3.7 GetNUMMAT()

```
unsigned int CElementGroup::GetNUMMAT ( ) [inline]
```

Return the number of material/section property setss in this element group.

4.5.3.8 operator[]()

```
CElement & CElementGroup::operator[] (
    unsigned int i )
```

operator [] For the sake of efficiency, the index bounds are not checked

4.5.3.9 Read()

```
bool CElementGroup::Read (
    ifstream & Input )
```

Read element group data from stream *Input*.

4.5.3.10 ReadElementData()

```
bool CElementGroup::ReadElementData (
    ifstream & Input )
```

Read element data from the input data file.

4.5.4 Member Data Documentation

4.5.4.1 ElementList_

```
CElement* CElementGroup::ElementList_ [private]
```

Element List in this group.

4.5.4.2 ElementSize_

```
std::size_t CElementGroup::ElementSize_ [private]
```

Size of an Element object in this group.

4.5.4.3 ElementType_

```
ElementTypes CElementGroup::ElementType_ [private]
```

Element type of this group.

4.5.4.4 MaterialList_

```
CMaterial* CElementGroup::MaterialList_ [private]
```

Material list in this group.

4.5.4.5 MaterialSize_

```
std::size_t CElementGroup::MaterialSize_ [private]
```

Size of an Material object in this group.

4.5.4.6 NodeList_

```
CNode * CElementGroup::NodeList_ = nullptr [static], [private]
```

List of all nodes in the domain, obtained from [CDomain](#) object.

4.5.4.7 NUME_

```
unsigned int CElementGroup::NUME_ [private]
```

Number of elements in this group.

4.5.4.8 NUMMAT_

```
unsigned int CElementGroup::NUMMAT_ [private]
```

Number of material/section property sets in this group.

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

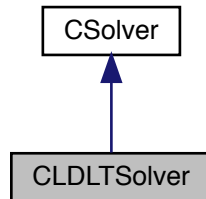
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/h/ElementGroup.h](#)
- [/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/ElementGroup.cpp](#)

4.6 CLDLTSolver Class Reference

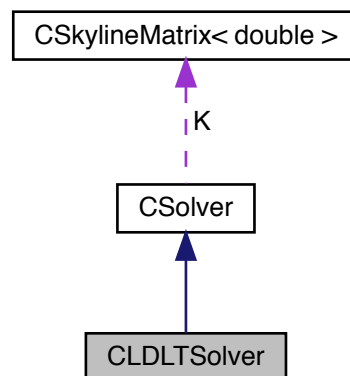
LDLT solver: A in core solver using skyline storage and column reduction scheme.

```
#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

4.6.1 Detailed Description

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

4.6.2 Constructor & Destructor Documentation

4.6.2.1 CLDLTSolver()

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

Constructor.

4.6.3 Member Function Documentation

4.6.3.1 BackSubstitution()

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

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

4.6.3.2 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](#)

4.7 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](#) (COutputter &output, unsigned int lcase)
Write load case data to stream.

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.

4.7.1 Detailed Description

Class LoadData is used to store load data.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 CLoadCaseData()

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

4.7.2.2 ~CLoadCaseData()

```
CLoadCaseData::~~CLoadCaseData ( )
```

4.7.3 Member Function Documentation

4.7.3.1 Allocate()

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

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

4.7.3.2 Read()

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

Read load case data from stream Input.

4.7.3.3 Write()

```
void CLoadCaseData::Write (
    COutputter & output,
    unsigned int lcase )
```

Write load case data to stream.

4.7.4 Member Data Documentation

4.7.4.1 dof

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

Degree of freedom number for this load component.

4.7.4.2 load

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

Magnitude of load.

4.7.4.3 nloads

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

Number of concentrated loads in this load case.

4.7.4.4 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](#)

4.8 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.

4.8.1 Detailed Description

[Clock](#) class for timing.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Clock()

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

Constructor.

4.8.3 Member Function Documentation

4.8.3.1 Clear()

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

Clear the clock.

4.8.3.2 ElapsedTime()

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

Return the elapsed time since the clock started.

4.8.3.3 Resume()

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

Resume the stoped clock.

4.8.3.4 Start()

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

Start the clock.

4.8.3.5 Stop()

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

Stop the clock.

4.8.4 Member Data Documentation

4.8.4.1 ct_

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

4.8.4.2 st0_

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

Flag for Start method.

4.8.4.3 st1_

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

Flag for Stop method.

4.8.4.4 t0_

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

4.8.4.5 t1_

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

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

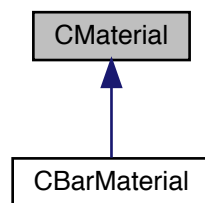
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Clock.cpp

4.9 CMaterial Class Reference

Material base class which only define one data member.

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

Inheritance diagram for CMaterial:



Public Member Functions

- virtual [~CMaterial](#) ()
Virtual destructor.
- virtual bool [Read](#) (ifstream &Input, unsigned int mset)=0
Read material data from stream Input.
- virtual void [Write](#) (COutputter &output, unsigned int mset)=0
Write material data to Stream.

Public Attributes

- unsigned int [nset](#)
Number of set.
- double [E](#)
Young's modulus.

4.9.1 Detailed Description

Material base class which only define one data member.

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

4.9.2 Constructor & Destructor Documentation

4.9.2.1 ~CMaterial()

```
virtual CMaterial::~CMaterial ( ) [inline], [virtual]
```

Virtual destructor.

4.9.3 Member Function Documentation

4.9.3.1 Read()

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

Read material data from stream Input.

Implemented in [CBarMaterial](#).

4.9.3.2 Write()

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

Write material data to Stream.

Implemented in [CBarMaterial](#).

4.9.4 Member Data Documentation

4.9.4.1 E

```
double CMaterial::E
```

Young's modulus.

4.9.4.2 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](#)

4.10 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](#) (COutputter &output, unsigned int np)
Output nodal point data to stream.
- void [WriteEquationNo](#) (COutputter &OutputFile, unsigned int np)
Output equation numbers of nodal point to stream OutputFile.
- void [WriteNodalDisplacement](#) (COutputter &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.

4.10.1 Detailed Description

Node class.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 CNode()

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

Constructor.

4.10.3 Member Function Documentation

4.10.3.1 Read()

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

Read nodal point data from stream Input.

4.10.3.2 Write()

```
void CNode::Write (
    COutputter & output,
    unsigned int np )
```

Output nodal point data to stream.

4.10.3.3 WriteEquationNo()

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

Output equation numbers of nodal point to stream OutputFile.

4.10.3.4 WriteNodalDisplacement()

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

Write nodal displacement.

4.10.4 Member Data Documentation

4.10.4.1 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 is nonactive (not defined)

After call Domain::CalculateEquationNumber(), bcode stores the global equation number corresponding to each degree of freedom of the node

4.10.4.2 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

4.10.4.3 NodeNumber

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

Node number.

4.10.4.4 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](#)

4.11 COutputter Class Reference

Outputter class is used to output results.

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

Collaboration diagram for COutputter:



Public Types

- typedef std::basic_ostream< char, std::char_traits< char > > [CharOstream](#)

Public Member Functions

- ofstream * [GetOutputFile](#) ()
Return pointer to the output file stream.
- void [PrintTime](#) (const struct tm *ptm, [COutputter](#) &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.
- template<typename T >
[COutputter](#) & [operator<<](#) (const T &item)
Overload the operator <<.
- [COutputter](#) & [operator<<](#) ([CharOstream](#) &(*op)([CharOstream](#) &))

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](#) = nullptr
Designed as a single instance class.

Private Attributes

- ofstream [OutputFile](#)
File stream for output.

4.11.1 Detailed Description

Outputter class is used to output results.

4.11.2 Member Typedef Documentation

4.11.2.1 CharOstream

```
typedef std::basic_ostream<char, std::char_traits<char> > COutputter::CharOstream
```

4.11.3 Constructor & Destructor Documentation

4.11.3.1 COutputter()

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

Constructor.

4.11.4 Member Function Documentation

4.11.4.1 GetOutputFile()

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

Return pointer to the output file stream.

4.11.4.2 Instance()

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

Return the single instance of the class.

4.11.4.3 operator<<() [1/2]

```
template<typename T >
COutputter& COutputter::operator<< (
    const T & item ) [inline]
```

Overload the operator <<.

4.11.4.4 operator<<() [2/2]

```
COutputter& COutputter::operator<< (
    CharOstream & (*) (CharOstream &) op ) [inline]
```

4.11.4.5 OutputElementInfo()

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

Output element data.

4.11.4.6 OutputElementStress()

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

Output element stresses.

4.11.4.7 OutputEquationNumber()

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

Output equation numbers.

4.11.4.8 OutputHeading()

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

Output logo and heading.

4.11.4.9 OutputLoadInfo()

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

Output load data.

4.11.4.10 OutputNodalDisplacement()

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

Output displacement data.

4.11.4.11 OutputNodeInfo()

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

Output nodal point data.

4.11.4.12 OutputTotalSystemData()

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

Print total system data.

4.11.4.13 PrintBarElementData()

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

Output bar element data.

4.11.4.14 PrintTime()

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

Output current time and date.

4.11.5 Member Data Documentation

4.11.5.1 _instance

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

Designed as a single instance class.

4.11.5.2 OutputFile

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

File stream for output.

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

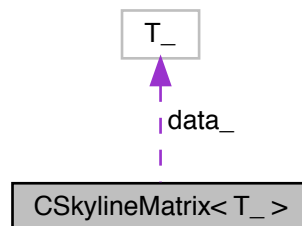
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Outputter.h
- /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Outputter.cpp

4.12 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.
- void [CalculateColumnHeight](#) (unsigned int *LocationMatrix, size_t ND)
Calculate the column height, used with the skyline storage scheme.
- void [CalculateMaximumHalfBandwidth](#) ()

- Calculate the maximum half bandwidth (= max(ColumnHeights) + 1)*
- void [CalculateDiagnoalAddress](#) ()
- void [Assembly](#) (double *Matrix, unsigned int *LocationMatrix, size_t ND)
 - Assemble the element stiffness matrix to the global stiffness matrix.*
- unsigned int * [GetColumnHeights](#) ()
 - Return pointer to the ColumnHeights_.*
- unsigned int [GetMaximumHalfBandwidth](#) () const
 - Return the maximum half bandwidth.*
- 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 [MK_](#)
 - Maximum half bandwith.*
- 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_.*

4.12.1 Detailed Description

```
template<class T_>
class CSkylineMatrix< T_ >
```

[CSkylineMatrix](#) class is used to store the FEM stiffness matrix in skyline storage.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 CSkylineMatrix() [1/2]

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

constructors

constructor functions

4.12.2.2 CSkylineMatrix() [2/2]

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

4.12.2.3 ~CSkylineMatrix()

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

destructor

destructor function

4.12.3 Member Function Documentation

4.12.3.1 Allocate()

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

Allocate storage for the skyline matrix.

Allocate storage for the matrix.

4.12.3.2 Assembly()

```
template<class T_ >
void CSkylineMatrix< T_ >::Assembly (
    double * Matrix,
    unsigned int * LocationMatrix,
    size_t ND )
```

Assemble the element stiffness matrix to the global stiffness matrix.

4.12.3.3 CalculateColumnHeight()

```
template<class T_ >
void CSkylineMatrix< T_ >::CalculateColumnHeight (
    unsigned int * LocationMatrix,
    size_t ND )
```

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

4.12.3.4 CalculateDiagnoalAddress()

```
template<class T_ >
void CSkylineMatrix< T_ >::CalculateDiagnoalAddress ( )
```

Calculate address of diagonal elements in banded matrix Caution: Address is numbered from 1 !

4.12.3.5 CalculateMaximumHalfBandwidth()

```
template<class T_ >
void CSkylineMatrix< T_ >::CalculateMaximumHalfBandwidth ( )
```

Calculate the maximum half bandwidth (= max(ColumnHeights) + 1)

4.12.3.6 dim()

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

Return the dimension of the stiffness matrix.

4.12.3.7 GetColumnHeights()

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

Return pointer to the ColumnHeights_.

4.12.3.8 GetDiagonalAddress()

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

Return pointer to the DiagonalAddress_.

4.12.3.9 GetMaximumHalfBandwidth()

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

Return the maximum half bandwidth.

4.12.3.10 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

4.12.3.11 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

4.12.3.12 size()

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

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

4.12.4 Member Data Documentation**4.12.4.1 ColumnHeights_**

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

Column heights.

4.12.4.2 data_

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

Store the stiffness matrix in skyline storage.

4.12.4.3 DiagonalAddress_

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

Diagonal address of all columns in data_.

4.12.4.4 MK_

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

Maximum half bandwidth.

4.12.4.5 NEQ_

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

Dimension of the stiffness matrix.

4.12.4.6 NWK_

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

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

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

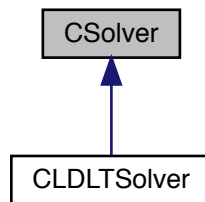
- /Users/xzhang/git/FEM-Projects/STAPpp/src/h/SkylineMatrix.h

4.13 CSolver Class Reference

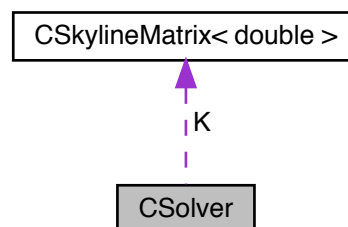
Base class for a solver.

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

Inheritance diagram for CSolver:



Collaboration diagram for CSolver:



Public Member Functions

- [CSolver](#) ([CSkylineMatrix](#)< double > *[K](#))

Protected Attributes

- [CSkylineMatrix](#)< double > * [K](#)

4.13.1 Detailed Description

Base class for a solver.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 CSolver()

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

4.13.3 Member Data Documentation

4.13.3.1 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](#)

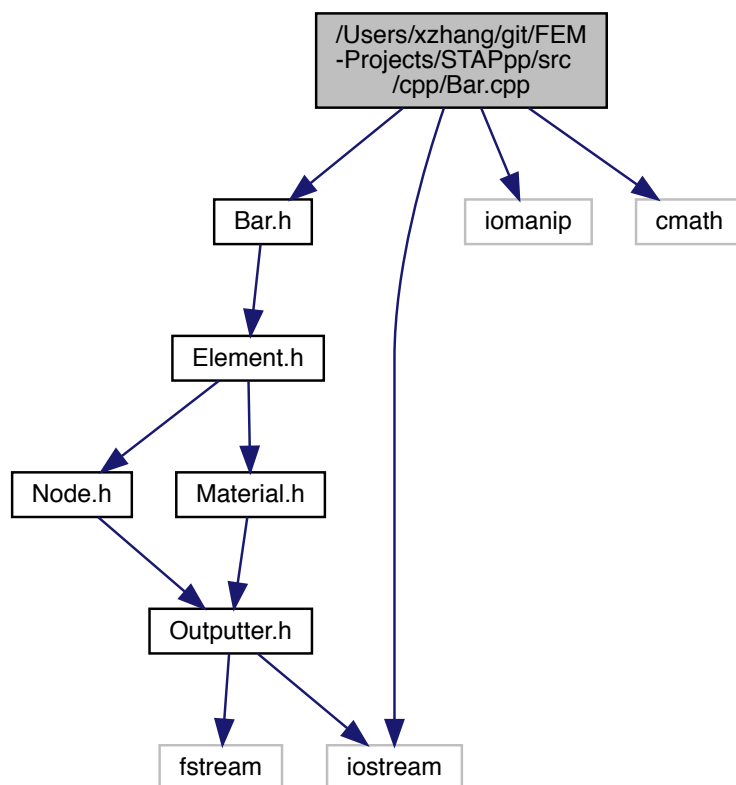
Chapter 5

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>
```

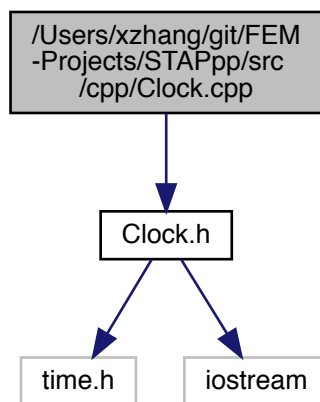
Include dependency graph for Bar.cpp:



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

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

Include dependency graph for Clock.cpp:

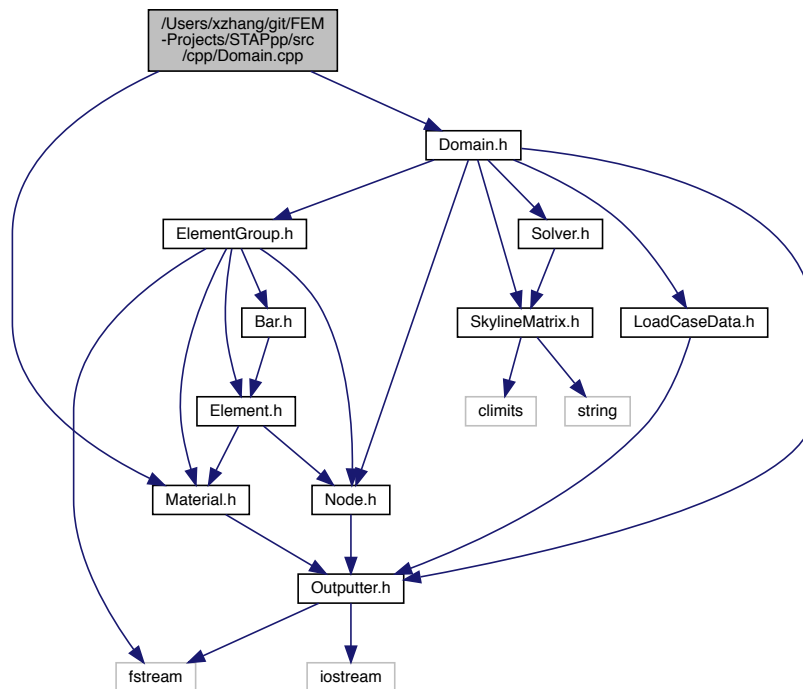


5.3 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Domain.cpp File Reference

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

```
#include "Material.h"
```


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 )

```

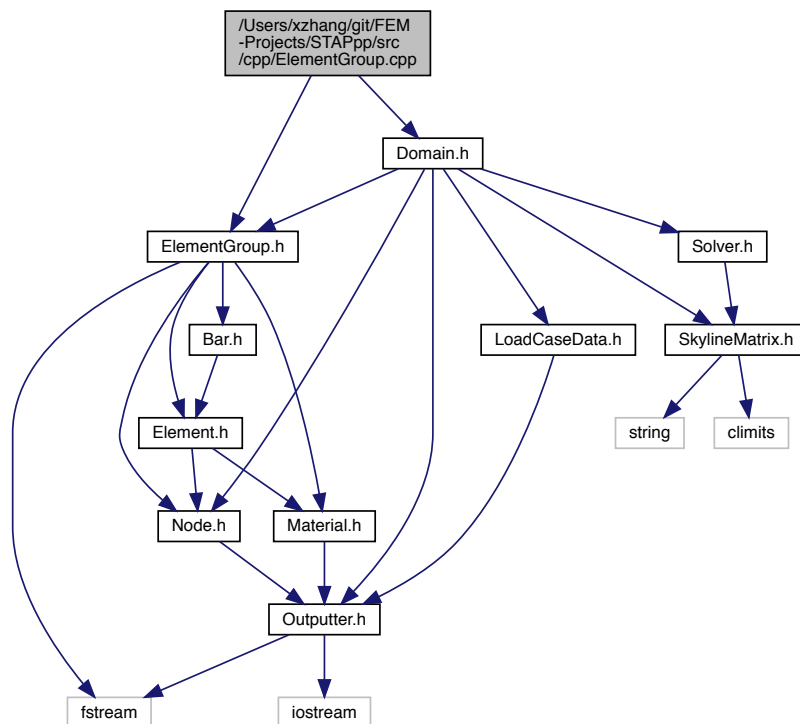
Clear an array.

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

```
#include "ElementGroup.h"
```

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

Include dependency graph for ElementGroup.cpp:



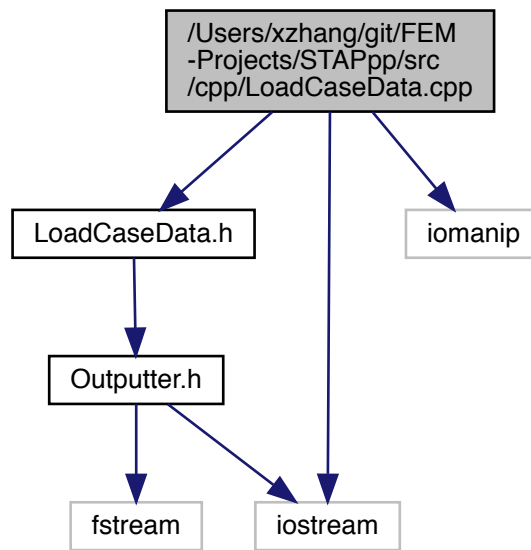
5.5 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/LoadCaseData.cpp File Reference

```
#include "LoadCaseData.h"
```

```
#include <iomanip>
```

```
#include <iostream>
```

Include dependency graph for LoadCaseData.cpp:



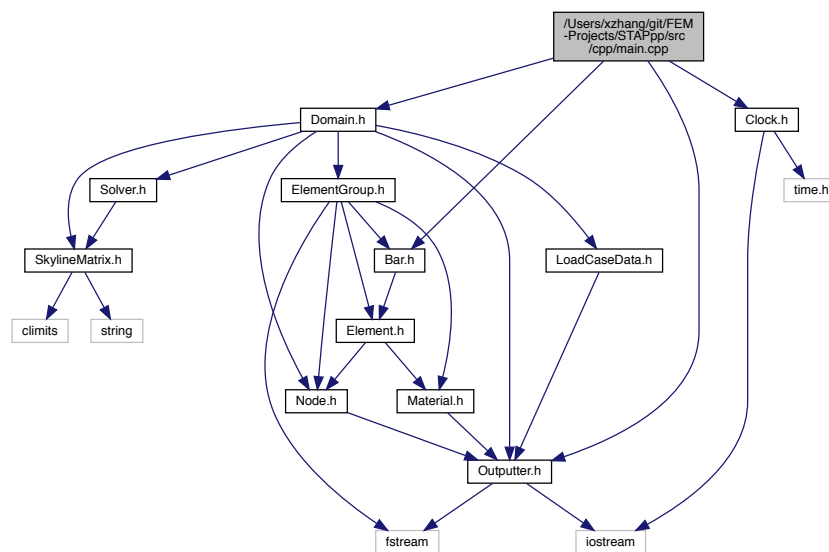
5.6 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.cpp File Reference

```

#include "Domain.h"
#include "Bar.h"
#include "Outputter.h"
#include "Clock.h"

```

Include dependency graph for main.cpp:



Functions

- int [main](#) (int argc, char *argv[])

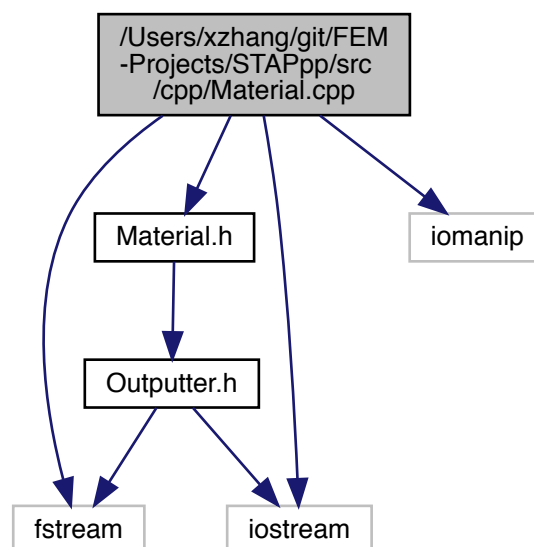
5.6.1 Function Documentation

5.6.1.1 main()

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

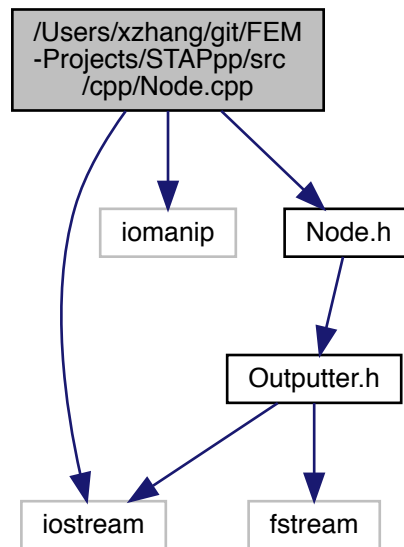
5.7 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Material.cpp File Reference

```
#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

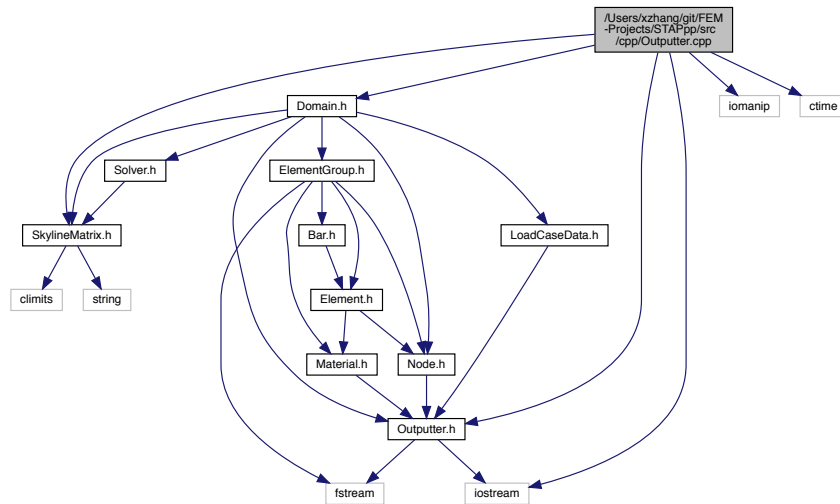
```
#include <iostream>
#include <iomanip>
#include "Node.h"
Include dependency graph for Node.cpp:
```



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:



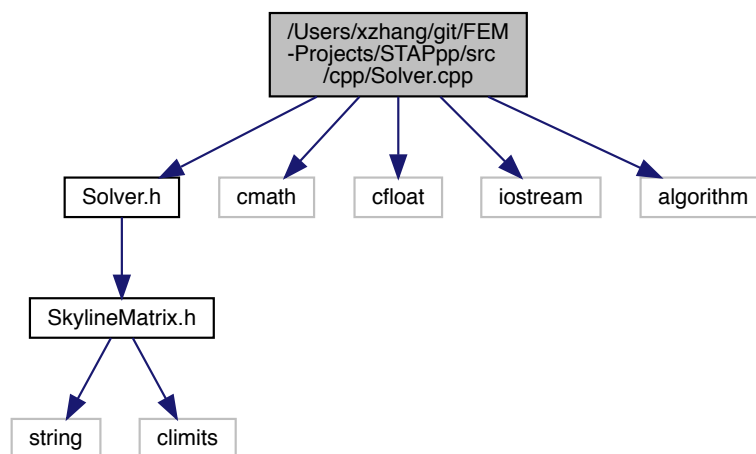
5.10 /Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Solver.cpp File Reference

```

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

```

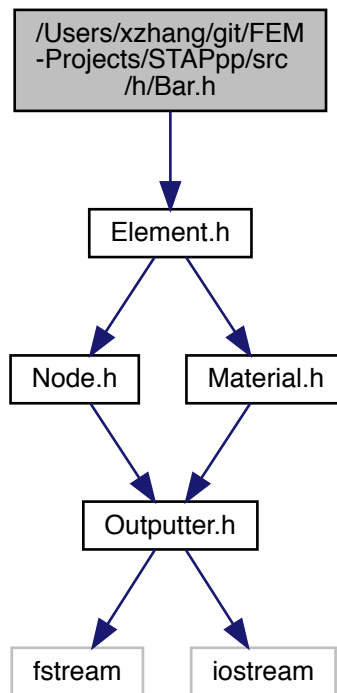
Include dependency graph for Solver.cpp:



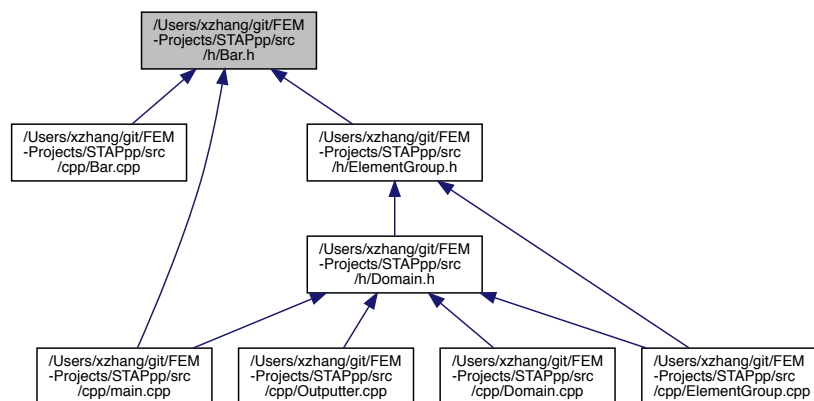
5.11 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h File Reference

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

Include dependency graph for Bar.h:



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



Classes

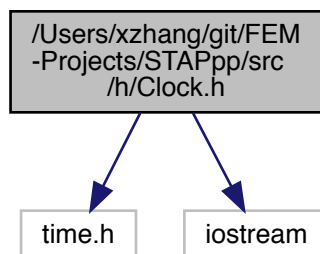
- class [CBar](#)

Bar element class.

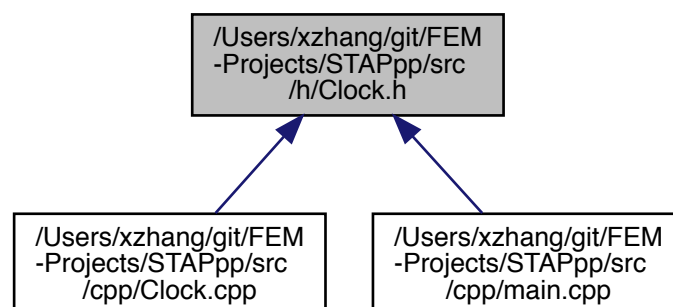
5.12 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h File Reference

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

Include dependency graph for Clock.h:



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



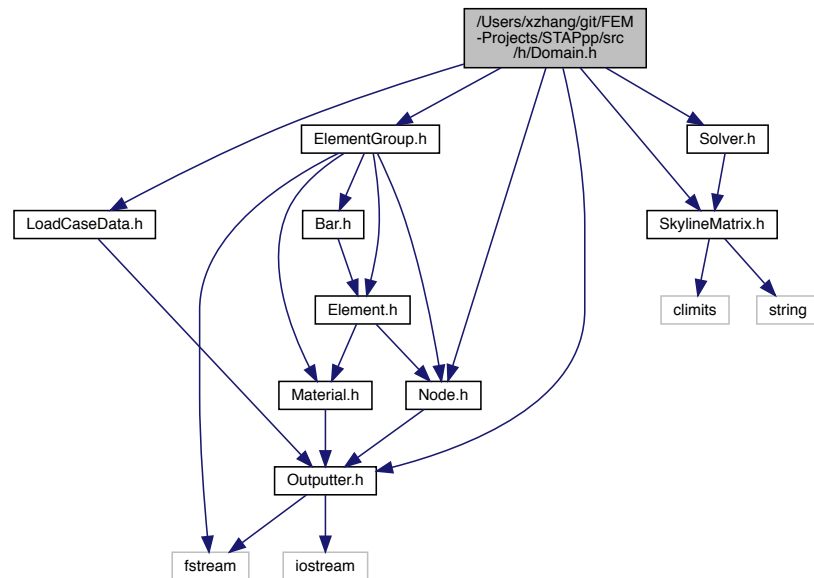
Classes

- class [Clock](#)

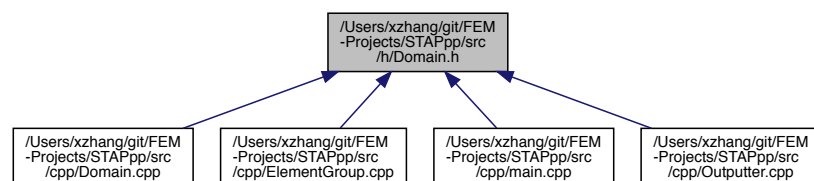
Clock class for timing.

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

```
#include "Node.h"
#include "ElementGroup.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:



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

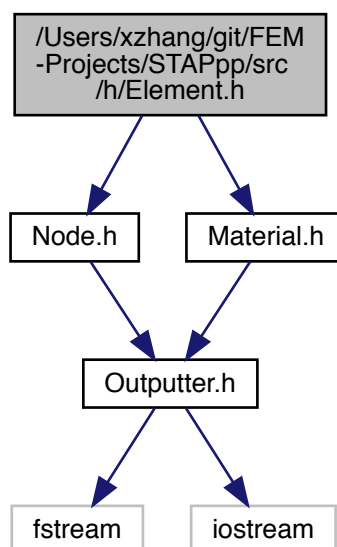
5.13.1.1 clear()

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

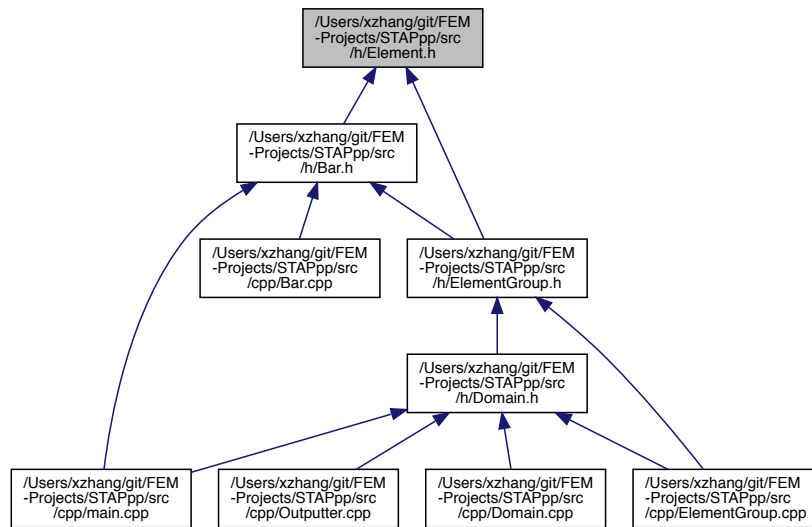
Clear an array.

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

```
#include "Node.h"
#include "Material.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 )

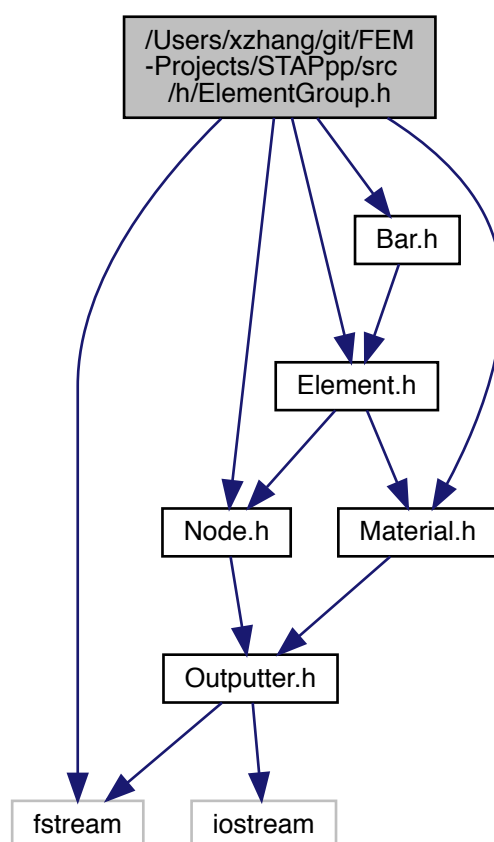
```

Clear an array.

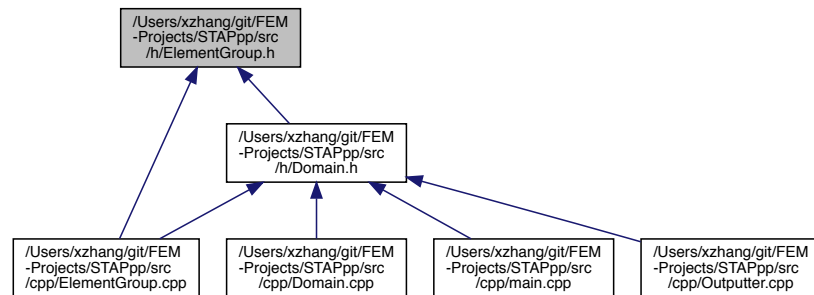
5.15 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/ElementGroup.h File Reference

```
#include <fstream>
#include "Element.h"
#include "Bar.h"
#include "Material.h"
#include "Node.h"
```

Include dependency graph for ElementGroup.h:



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



Classes

- class [CElementGroup](#)
Element group class.

Enumerations

- enum [ElementTypes](#) {
 [UNDEFINED](#) = 0, [Bar](#), [Q4](#), [T3](#),
 [H8](#), [Beam](#), [Plate](#), [Shell](#) }
Define set of element types.

5.15.1 Enumeration Type Documentation

5.15.1.1 ElementTypes

```
enum ElementTypes
```

Define set of element types.

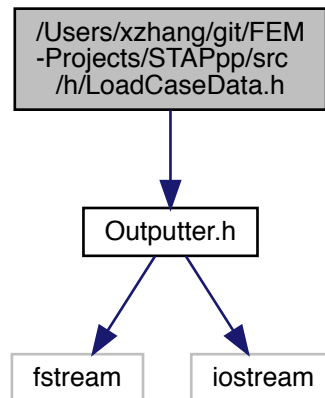
Enumerator

UNDEFINED	
Bar	
Q4	
T3	
H8	
Beam	
Plate	
Shell	

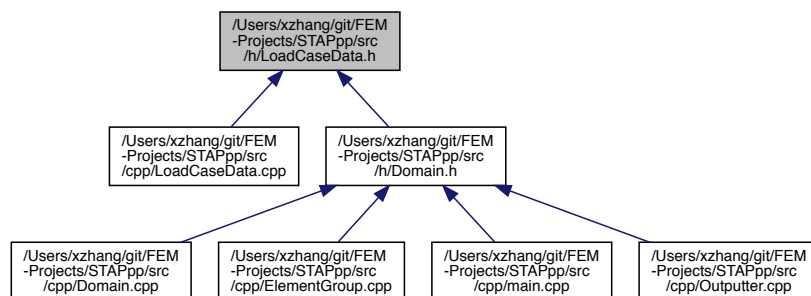
5.16 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/LoadCaseData.h File Reference

```
#include "Outputter.h"
```

Include dependency graph for LoadCaseData.h:



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



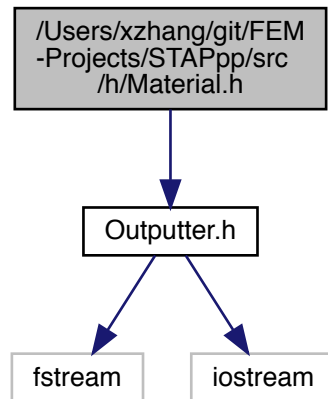
Classes

- class [CLoadCaseData](#)

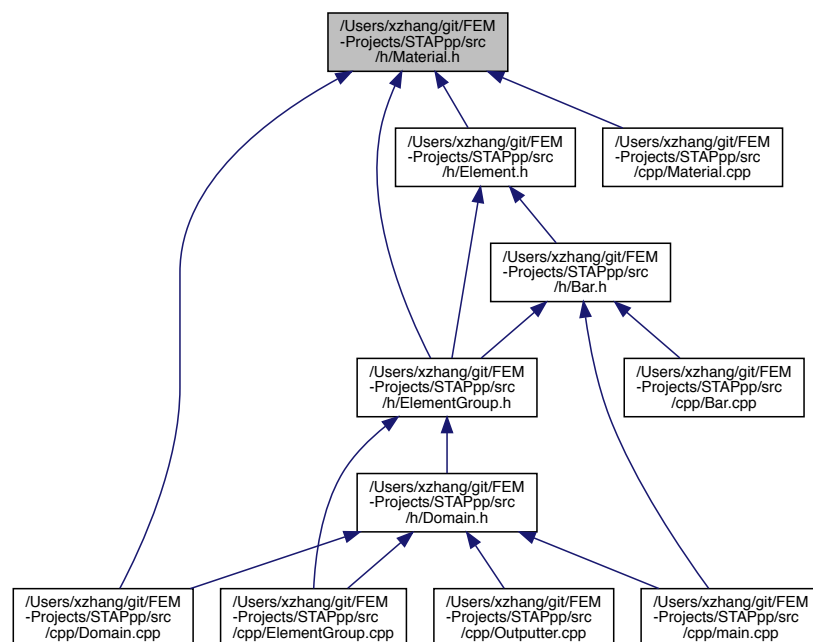
Class LoadData is used to store load data.

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

```
#include "Outputter.h"
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.

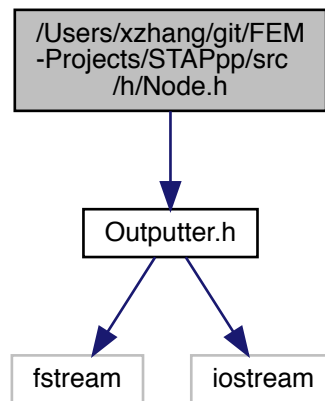
- class [CBarMaterial](#)

Material class for bar element.

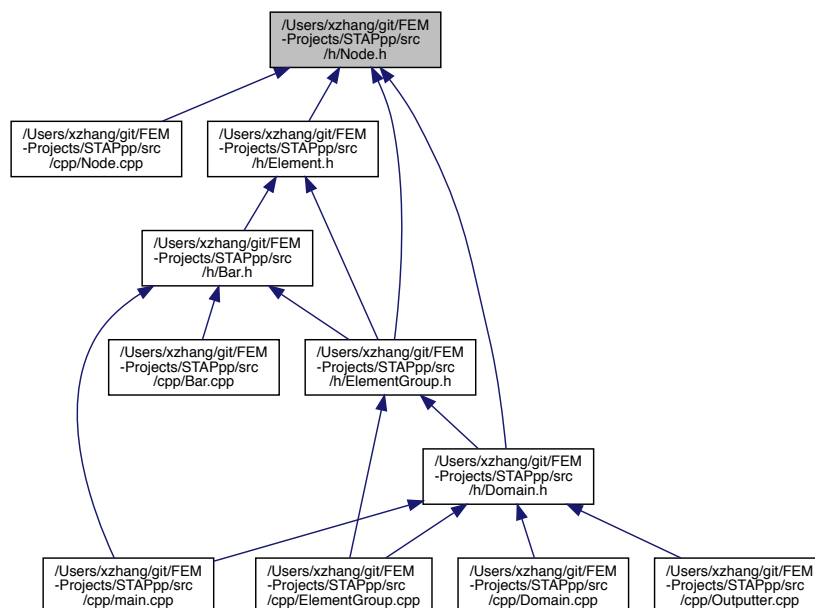
5.18 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Node.h File Reference

```
#include "Outputter.h"
```

Include dependency graph for Node.h:



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



Classes

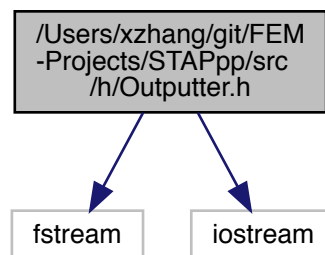
- class [CNode](#)
Node class.

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

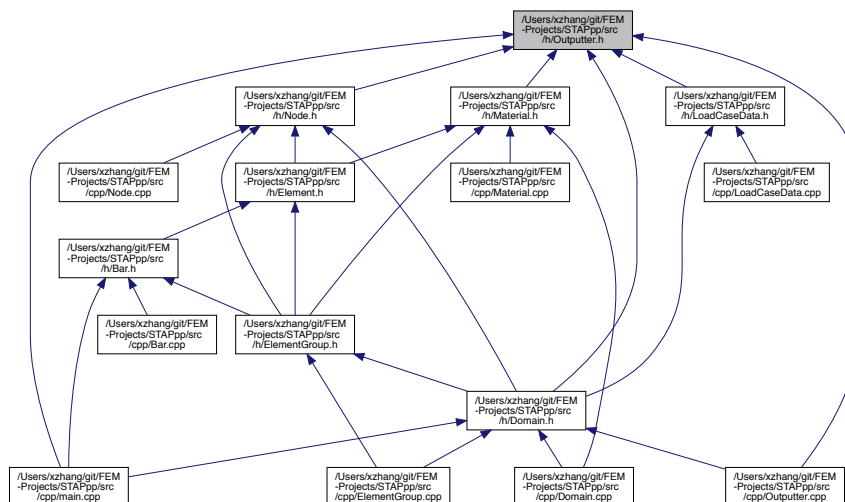
```
#include <fstream>
```

```
#include <iostream>
```

Include dependency graph for Outputter.h:



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



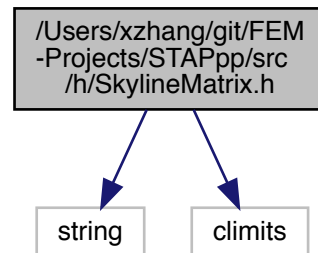
Classes

- class [COutputter](#)
Outputter class is used to output results.

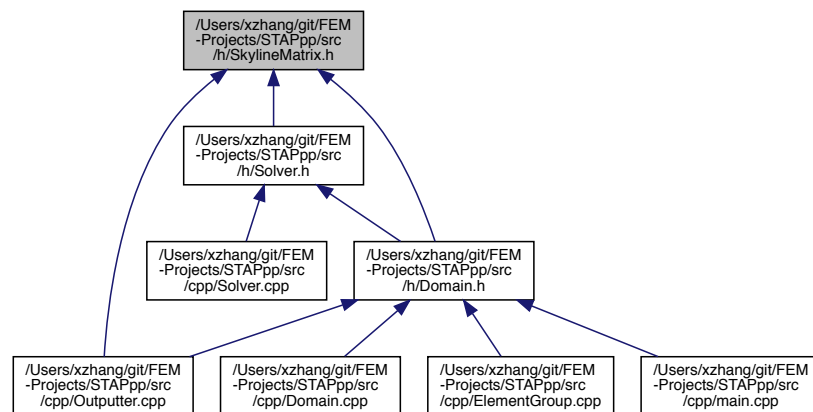
5.20 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/SkylineMatrix.h File Reference

```
#include <string>
#include <climits>
```

Include dependency graph for SkylineMatrix.h:



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



Classes

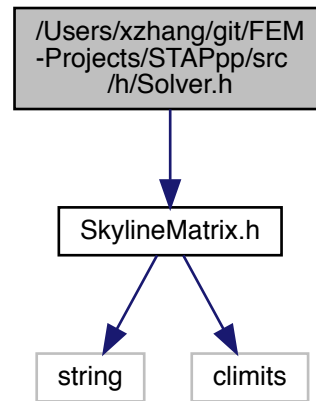
- class [CSkylineMatrix< T_ >](#)

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

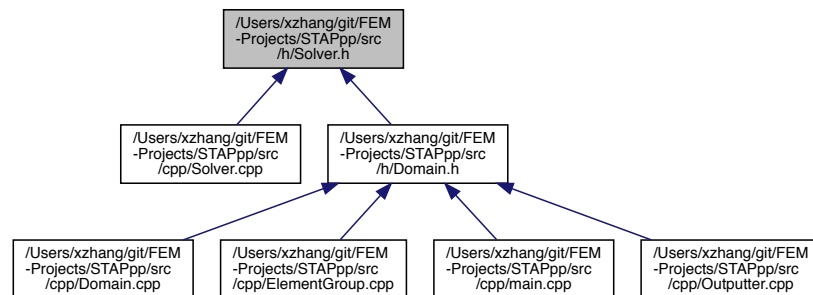
5.21 /Users/xzhang/git/FEM-Projects/STAPpp/src/h/Solver.h File Reference

```
#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.

Index

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/Bar.↔
cpp, 59

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
Clock.cpp, 60

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
Domain.cpp, 60

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
ElementGroup.cpp, 62

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
LoadCaseData.cpp, 62

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
Material.cpp, 64

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
Node.cpp, 65

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
Outputter.cpp, 65

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/↔
Solver.cpp, 66

/Users/xzhang/git/FEM-Projects/STAPpp/src/cpp/main.↔
cpp, 63

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Bar.h,
67

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Clock.h,
68

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
Domain.h, 69

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
Element.h, 70

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
ElementGroup.h, 72

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Load↔
CaseData.h, 74

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
Material.h, 75

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/Node.h,
76

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
Outputter.h, 77

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
SkylineMatrix.h, 78

/Users/xzhang/git/FEM-Projects/STAPpp/src/h/↔
Solver.h, 79

_instance
CDomain, 19
COutputter, 49

~CBar
CBar, 9

~CDomain
CDomain, 15

~CElement
CElement, 24

~CElementGroup
CElementGroup, 29

~CLoadCaseData
CLoadCaseData, 35

~CMaterial
CMaterial, 41

~CSkylineMatrix
CSkylineMatrix, 52

Allocate
CLoadCaseData, 35
CSkylineMatrix, 52

AllocateElements
CElementGroup, 29

AllocateMaterials
CElementGroup, 29

AllocateMatrices
CDomain, 15

Area
CBarMaterial, 12

AssembleForce
CDomain, 16

AssembleStiffnessMatrix
CDomain, 16

Assembly
CSkylineMatrix, 52

BackSubstitution
CLDLTSolver, 34

bcode
CNode, 44

CBar, 7
~CBar, 9
CBar, 8
ElementStiffness, 9
ElementStress, 9
GenerateLocationMatrix, 9
Read, 9
SizeOfStiffnessMatrix, 10
Write, 10

CBarMaterial, 11
Area, 12
Read, 12
Write, 12

CDomain, 13
_instance, 19
~CDomain, 15

- AllocateMatrices, 15
- AssembleForce, 16
- AssembleStiffnessMatrix, 16
- CDomain, 15
- CalculateColumnHeights, 16
- CalculateEquationNumber, 16
- EleGrpList, 20
- Force, 20
- GetDisplacement, 16
- GetEleGrpList, 16
- GetForce, 17
- GetLoadCases, 17
- GetMODEX, 17
- GetNEQ, 17
- GetNLCASE, 17
- GetNLOAD, 17
- GetNUMEG, 18
- GetNUMNP, 18
- GetNodeList, 18
- GetStiffnessMatrix, 18
- GetTitle, 18
- Input, 20
- Instance, 18
- LoadCases, 20
- MODEX, 20
- NEQ, 20
- NLCASE, 21
- NLOAD, 21
- NUMEG, 21
- NUMNP, 21
- NodeList, 21
- ReadData, 19
- ReadElements, 19
- ReadLoadCases, 19
- ReadNodalPoints, 19
- StiffnessMatrix, 21
- Title, 22
- CElement, 22
 - ~CElement, 24
 - CElement, 23
 - ElementMaterial_, 26
 - ElementStiffness, 24
 - ElementStress, 24
 - GenerateLocationMatrix, 24
 - GetElementMaterial, 24
 - GetLocationMatrix, 25
 - GetND, 25
 - GetNodes, 25
 - LocationMatrix_, 26
 - ND_, 26
 - NEN_, 26
 - nodes_, 27
 - Read, 25
 - SizeOfStiffnessMatrix, 25
 - Write, 26
- CElementGroup, 27
 - ~CElementGroup, 29
 - AllocateElements, 29
 - AllocateMaterials, 29
 - CElementGroup, 29
 - CalculateMemberSize, 29
 - ElementList_, 31
 - ElementSize_, 31
 - ElementType_, 31
 - GetElementType, 30
 - GetMaterial, 30
 - GetNUMMAT, 30
 - GetNUME, 30
 - MaterialList_, 31
 - MaterialSize_, 32
 - NUME_, 32
 - NUMMAT_, 32
 - NodeList_, 32
 - operator[], 30
 - Read, 30
 - ReadElementData, 31
- CLDLTSolver, 33
 - BackSubstitution, 34
 - CLDLTSolver, 34
 - LDLT, 34
- CLoadCaseData, 34
 - ~CLoadCaseData, 35
 - Allocate, 35
 - CLoadCaseData, 35
 - dof, 36
 - load, 36
 - nloads, 36
 - node, 37
 - Read, 36
 - Write, 36
- CMaterial, 40
 - ~CMaterial, 41
 - E, 41
 - nset, 42
 - Read, 41
 - Write, 41
- CNode, 42
 - bcode, 44
 - CNode, 43
 - NDF, 44
 - NodeNumber, 44
 - Read, 43
 - Write, 43
 - WriteEquationNo, 43
 - WriteNodalDisplacement, 44
 - XYZ, 44
- COutputter, 45
 - _instance, 49
 - COutputter, 47
 - CharOstream, 46
 - GetOutputFile, 47
 - Instance, 47
 - operator<<, 47
 - OutputElementInfo, 47
 - OutputElementStress, 48
 - OutputEquationNumber, 48

- OutputFile, [49](#)
- OutputHeading, [48](#)
- OutputLoadInfo, [48](#)
- OutputNodalDisplacement, [48](#)
- OutputNodeInfo, [48](#)
- OutputTotalSystemData, [49](#)
- PrintBarElementData, [49](#)
- PrintTime, [49](#)
- CSkylineMatrix
 - ~CSkylineMatrix, [52](#)
 - Allocate, [52](#)
 - Assembly, [52](#)
 - CSkylineMatrix, [51](#)
 - CalculateColumnHeight, [52](#)
 - CalculateDiagnoalAddress, [52](#)
 - CalculateMaximumHalfBandwidth, [53](#)
 - ColumnHeights_, [54](#)
 - data_, [54](#)
 - DiagonalAddress_, [54](#)
 - dim, [53](#)
 - GetColumnHeights, [53](#)
 - GetDiagonalAddress, [53](#)
 - GetMaximumHalfBandwidth, [53](#)
 - MK_, [55](#)
 - NEQ_, [55](#)
 - NWK_, [55](#)
 - operator(), [53](#), [54](#)
 - size, [54](#)
- CSkylineMatrix< T_ >, [50](#)
- CSolver, [56](#)
 - CSolver, [57](#)
 - K, [57](#)
- CalculateColumnHeight
 - CSkylineMatrix, [52](#)
- CalculateColumnHeights
 - CDomain, [16](#)
- CalculateDiagnoalAddress
 - CSkylineMatrix, [52](#)
- CalculateEquationNumber
 - CDomain, [16](#)
- CalculateMaximumHalfBandwidth
 - CSkylineMatrix, [53](#)
- CalculateMemberSize
 - CElementGroup, [29](#)
- CharOstream
 - COutputter, [46](#)
- Clear
 - Clock, [38](#)
- clear
 - Domain.cpp, [61](#)
 - Domain.h, [70](#)
 - Element.h, [71](#)
- Clock, [37](#)
 - Clear, [38](#)
 - Clock, [38](#)
 - ct_, [39](#)
 - ElapsedTime, [38](#)
 - Resume, [38](#)
 - st0_, [39](#)
 - st1_, [39](#)
 - Start, [38](#)
 - Stop, [39](#)
 - t0_, [39](#)
 - t1_, [39](#)
- ColumnHeights_
 - CSkylineMatrix, [54](#)
- ct_
 - Clock, [39](#)
- data_
 - CSkylineMatrix, [54](#)
- DiagonalAddress_
 - CSkylineMatrix, [54](#)
- dim
 - CSkylineMatrix, [53](#)
- dof
 - CLoadCaseData, [36](#)
- Domain.cpp
 - clear, [61](#)
- Domain.h
 - clear, [70](#)
- E
 - CMaterial, [41](#)
- ElapsedTime
 - Clock, [38](#)
- EleGrpList
 - CDomain, [20](#)
- Element.h
 - clear, [71](#)
- ElementGroup.h
 - ElementTypes, [73](#)
- ElementList_
 - CElementGroup, [31](#)
- ElementMaterial_
 - CElement, [26](#)
- ElementSize_
 - CElementGroup, [31](#)
- ElementStiffness
 - CBar, [9](#)
 - CElement, [24](#)
- ElementStress
 - CBar, [9](#)
 - CElement, [24](#)
- ElementType_
 - CElementGroup, [31](#)
- ElementTypes
 - ElementGroup.h, [73](#)
- Force
 - CDomain, [20](#)
- GenerateLocationMatrix
 - CBar, [9](#)
 - CElement, [24](#)
- GetColumnHeights
 - CSkylineMatrix, [53](#)

- GetDiagonalAddress
 - CSkylineMatrix, 53
- GetDisplacement
 - CDomain, 16
- GetEleGrpList
 - CDomain, 16
- GetElementMaterial
 - CElement, 24
- GetElementType
 - CElementGroup, 30
- GetForce
 - CDomain, 17
- GetLoadCases
 - CDomain, 17
- GetLocationMatrix
 - CElement, 25
- GetMODEX
 - CDomain, 17
- GetMaterial
 - CElementGroup, 30
- GetMaximumHalfBandwidth
 - CSkylineMatrix, 53
- GetNEQ
 - CDomain, 17
- GetNLCASE
 - CDomain, 17
- GetNLOAD
 - CDomain, 17
- GetNUMEG
 - CDomain, 18
- GetNUMMAT
 - CElementGroup, 30
- GetNUMNP
 - CDomain, 18
- GetNUME
 - CElementGroup, 30
- GetND
 - CElement, 25
- GetNodeList
 - CDomain, 18
- GetNodes
 - CElement, 25
- GetOutputFile
 - COutputter, 47
- GetStiffnessMatrix
 - CDomain, 18
- GetTitle
 - CDomain, 18
- Input
 - CDomain, 20
- Instance
 - CDomain, 18
 - COutputter, 47
- K
 - CSolver, 57
- LDLT
 - CLDLTSolver, 34
- load
 - CLoadCaseData, 36
- LoadCases
 - CDomain, 20
- LocationMatrix_
 - CElement, 26
- MK_
 - CSkylineMatrix, 55
- MODEX
 - CDomain, 20
- main
 - main.cpp, 64
- main.cpp
 - main, 64
- MaterialList_
 - CElementGroup, 31
- MaterialSize_
 - CElementGroup, 32
- ND_
 - CElement, 26
- NDF
 - CNode, 44
- NEN_
 - CElement, 26
- NEQ_
 - CSkylineMatrix, 55
- NEQ
 - CDomain, 20
- NLCASE
 - CDomain, 21
- NLOAD
 - CDomain, 21
- NUME_
 - CElementGroup, 32
- NUMEG
 - CDomain, 21
- NUMMAT_
 - CElementGroup, 32
- NUMNP
 - CDomain, 21
- NWK_
 - CSkylineMatrix, 55
- nloads
 - CLoadCaseData, 36
- node
 - CLoadCaseData, 37
- NodeList
 - CDomain, 21
- NodeList_
 - CElementGroup, 32
- NodeNumber
 - CNode, 44
- nodes_
 - CElement, 27
- nset
 - CMaterial, 42

- operator<<
 - COutputter, 47
- operator()
 - CSkylineMatrix, 53, 54
- operator[]
 - CElementGroup, 30
- OutputElementInfo
 - COutputter, 47
- OutputElementStress
 - COutputter, 48
- OutputEquationNumber
 - COutputter, 48
- OutputFile
 - COutputter, 49
- OutputHeading
 - COutputter, 48
- OutputLoadInfo
 - COutputter, 48
- OutputNodalDisplacement
 - COutputter, 48
- OutputNodeInfo
 - COutputter, 48
- OutputTotalSystemData
 - COutputter, 49
- PrintBarElementData
 - COutputter, 49
- PrintTime
 - COutputter, 49
- Read
 - CBar, 9
 - CBarMaterial, 12
 - CElement, 25
 - CElementGroup, 30
 - CLoadCaseData, 36
 - CMaterial, 41
 - CNode, 43
- ReadData
 - CDomain, 19
- ReadElementData
 - CElementGroup, 31
- ReadElements
 - CDomain, 19
- ReadLoadCases
 - CDomain, 19
- ReadNodalPoints
 - CDomain, 19
- Resume
 - Clock, 38
- size
 - CSkylineMatrix, 54
- SizeOfStiffnessMatrix
 - CBar, 10
 - CElement, 25
- st0_
 - Clock, 39
- st1_
 - Clock, 39
- Start
 - Clock, 38
- StiffnessMatrix
 - CDomain, 21
- Stop
 - Clock, 39
- t0_
 - Clock, 39
- t1_
 - Clock, 39
- Title
 - CDomain, 22
- Write
 - CBar, 10
 - CBarMaterial, 12
 - CElement, 26
 - CLoadCaseData, 36
 - CMaterial, 41
 - CNode, 43
- WriteEquationNo
 - CNode, 43
- WriteNodalDisplacement
 - CNode, 44
- XYZ
 - CNode, 44