Trilinos/Petra & Aztec

R. Tuminaro
Sandia National Labs

< http://www.cs.sandia.gov/CRF/aztec1.html>

Trilinos: M. Heroux & R. Lehoucq

Solver Team: M. Adams, D. Day, J. Hu,
S. Hutchinson, N. Nachtigal,
J. Shadid, A. Williams,
C. Tong (LLNL)

Aztec

Parallel Iterative Methods for Solving

$$Ax = b$$

where A is an $n \times n$ matrix via iteration

$$x^{(i+1)} = x^{(i)} + G_i(b - Ax^i)$$

Preconditioning

Replace with

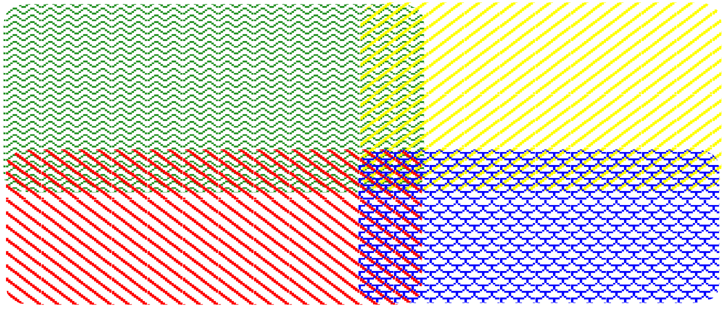
$$AM^{-1}y = b$$

where

$$x = M^{-1}y$$

and $M^{-1}y$ is relatively easy to compute.

domain
decomposition:

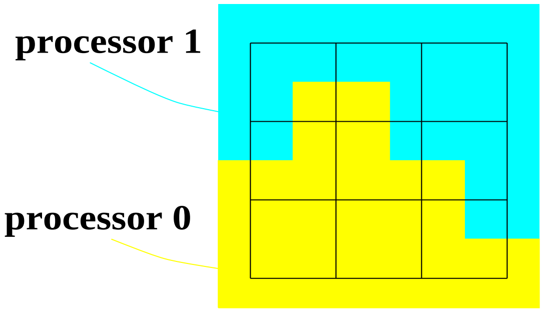


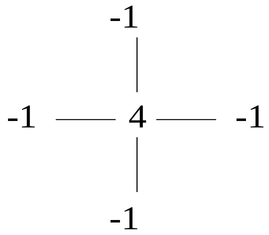
Brief Aztec/Trilinos History

- 1993 Aztec within Sandia
  - 1 matrix format
- 1995 Aztec released externally
  - 2 matrix formats
- 1997 Aztec 2.0 released externally
  - matrix-free interface
- 1998 ML 1.0 within Sandia
- 1999 Aztec 2.1 released externally
- 2000 Trilinos within Sandia

Using Aztec

- processor information
- initialize matrix and right hand side





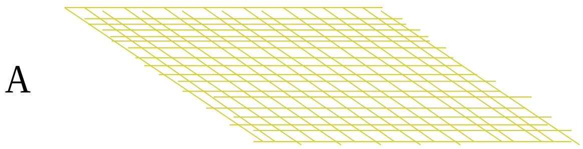
Stencil

```
MSRadd_row(loc, row, n) {
  next_nonzero(row, 4, ...);
  if ( row%n != n-1) next_nonzero(row+1,-1, ...);
  if ( row%n != 0  ) next_nonzero(row-1,-1, ...);
  if ( row   >= n  ) next_nonzero(row-n,-1, ...);
  if ( row+n < n*n ) next_nonzero(row+n,-1, ...);
}
next_nonzero(col, value, ...) {
   if (col == row) a[loc] = value;
   else {
      ija[ija[loc+1]  ] = col;
        a[ija[loc+1]++] = value;
   }
}
```

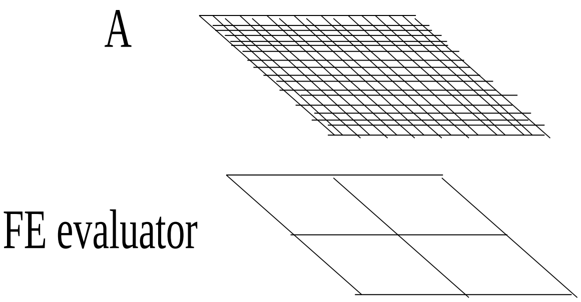
- choose solver/preconditioner options
- solve

ML Overview

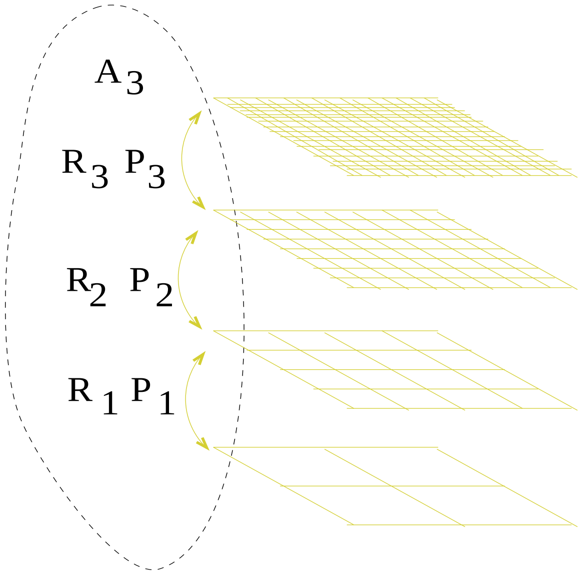
- Smoothed Aggregation Algebraic Multigrid:



- Finite Element basis domain decomposition:



- User Defined:



- Smoothers: Jacobi, Gauss-Seidel, block Gauss-Seidel, symmetric Gauss-Seidel, any Aztec solver or preconditioner ...
- Can use as preconditioner to Aztec solvers

Sample Program

```
      .
      .
Petra_Comm& Comm = *new Petra_Comm(MPI_COMM_WORLD);
      .
      .
Petra_Map& Map = *new Petra_Map(NumGlobalElements, 0, Comm);
Petra_RDP_Vector& q = *new Petra_RDP_Vector(Map);
      .
      .
Map.MyGlobalElements(MyGlobalElements);

// NumNz is used to build Petra Matrix.  NumNz[i] is the
// Number of OFF-DIAGONAL terms for the ith global equation
// on this processor

// We are building a tridiagonal matrix where each row has
// (-1 2 -1) So we need 2 off-diagonal terms (except for
// the first and last equation)

for (i=0; i<NumMyElements; i++) {
  if (MyGlobalElements[i] == 0 ||
      MyGlobalElements[i] == NumGlobalElements-1)
      NumNz[i] = 1;
  else NumNz[i] = 2;
}
      .
      .
```

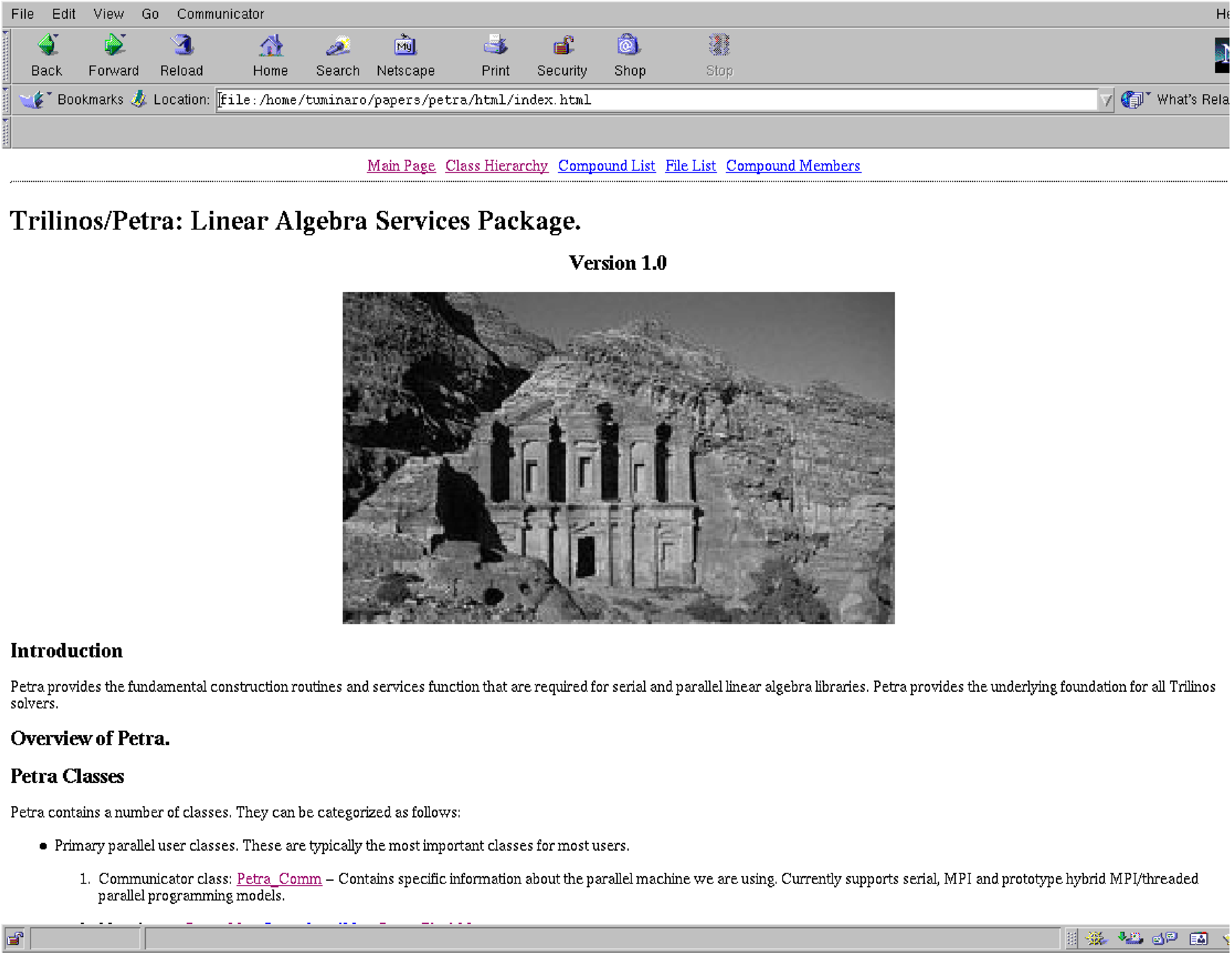
```
// Create Petra_Matrix

Petra_RDP_DCRS_Matrix& A= *new Petra_RDP_DCRS_Matrix(Map,NumNz);

// Add  rows one-at-a-time
Values[0] = -1.0; Values[1] = -1.0;

for (i=0; i<NumMyElements; i++) {
   if (MyGlobalElements[i]==0) {
        Indices[0] = 1;
        NumEntries = 1;
   }
   else if (MyGlobalElements[i] == NumGlobalElements-1) {
        Indices[0] = NumGlobalElements-2;
        NumEntries = 1;
   }
   else {
        Indices[0] = MyGlobalElements[i]-1;
        Indices[1] = MyGlobalElements[i]+1;
        NumEntries = 2;
   }
   A.InsertGlobalValues(MyGlobalElements[i], NumEntries, Values,
                        Indices);
   // Put in diagonal entry
   A.InsertGlobalValues(MyGlobalElements[i], 1, &two,
                        MyGlobalElements+i);
}
A.FillComplete();

  .
  .
```



Main Page Class Hierarchy Compound List File List Compound Members

Trilinos/Petra: Linear Algebra Services Package.

Version 1.0

Introduction

Petra provides the fundamental construction routines and services function that are required for serial and parallel linear algebra libraries. Petra provides the underlying foundation for all Trilinos solvers.

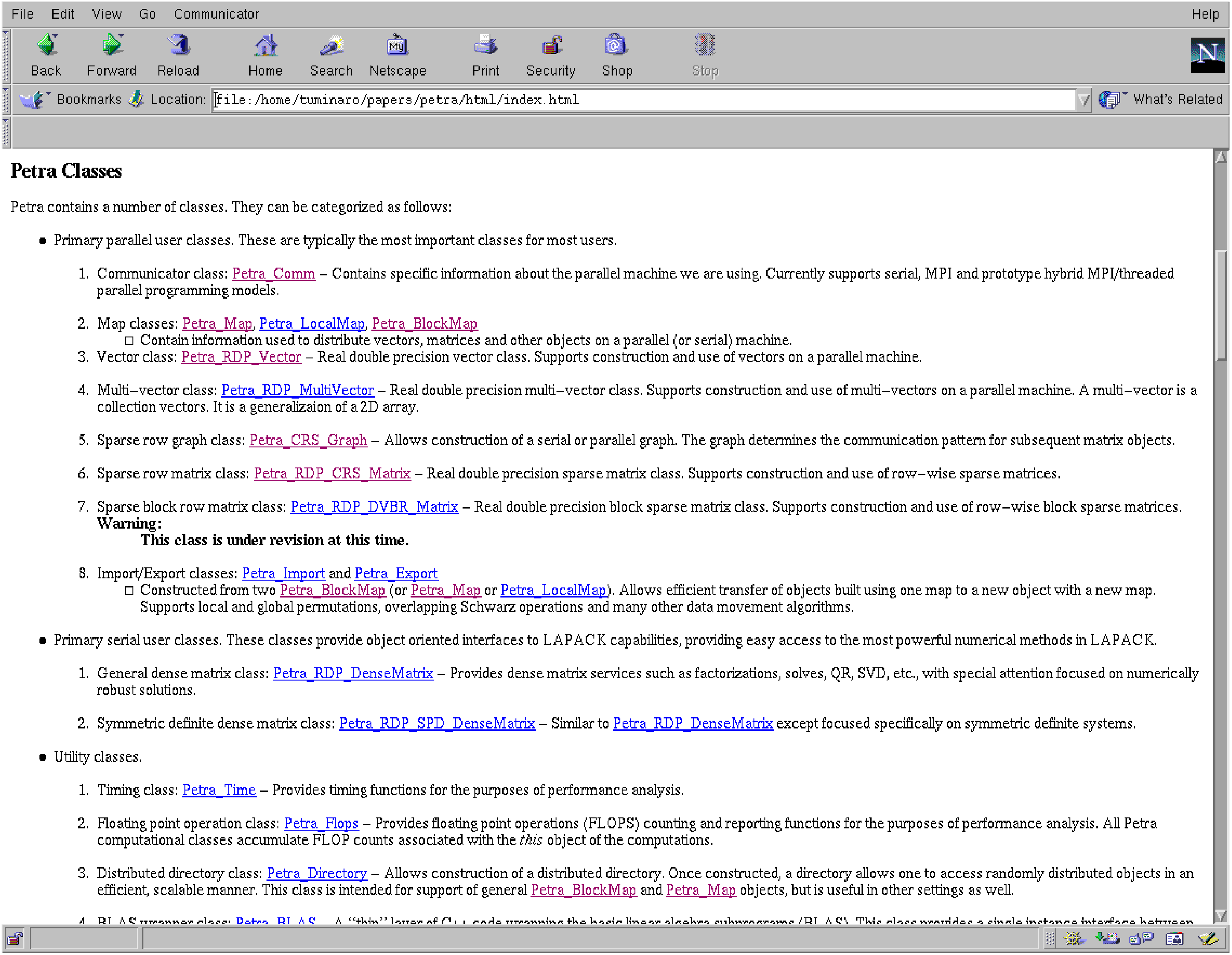
Overview of Petra.

Petra Classes

Petra contains a number of classes. They can be categorized as follows:

- Primary parallel user classes. These are typically the most important classes for most users.

  1. Communicator class: Petra_Comm – Contains specific information about the parallel machine we are using. Currently supports serial, MPI and prototype hybrid MPI/threaded parallel programming models.



Petra Classes

Petra contains a number of classes. They can be categorized as follows:

- Primary parallel user classes. These are typically the most important classes for most users.

  1. Communicator class: Petra_Comm – Contains specific information about the parallel machine we are using. Currently supports serial, MPI and prototype hybrid MPI/threaded parallel programming models.

  2. Map classes: Petra_Map, Petra_LocalMap, Petra_BlockMap
     - □ Contain information used to distribute vectors, matrices and other objects on a parallel (or serial) machine.
  3. Vector class: Petra_RDP_Vector – Real double precision vector class. Supports construction and use of vectors on a parallel machine.

  4. Multi–vector class: Petra_RDP_MultiVector – Real double precision multi–vector class. Supports construction and use of multi–vectors on a parallel machine. A multi–vector is a collection vectors. It is a generalizaion of a 2D array.

  5. Sparse row graph class: Petra_CRS_Graph – Allows construction of a serial or parallel graph. The graph determines the communication pattern for subsequent matrix objects.

  6. Sparse row matrix class: Petra_RDP_CRS_Matrix – Real double precision sparse matrix class. Supports construction and use of row–wise sparse matrices.

  7. Sparse block row matrix class: Petra_RDP_DVBR_Matrix – Real double precision block sparse matrix class. Supports construction and use of row–wise block sparse matrices.
     Warning:
     This class is under revision at this time.

  8. Import/Export classes: Petra_Import and Petra_Export
     - □ Constructed from two Petra_BlockMap (or Petra_Map or Petra_LocalMap). Allows efficient transfer of objects built using one map to a new object with a new map. Supports local and global permutations, overlapping Schwarz operations and many other data movement algorithms.

- Primary serial user classes. These classes provide object oriented interfaces to LAPACK capabilities, providing easy access to the most powerful numerical methods in LAPACK.

  1. General dense matrix class: Petra_RDP_DenseMatrix – Provides dense matrix services such as factorizations, solves, QR, SVD, etc., with special attention focused on numerically robust solutions.

  2. Symmetric definite dense matrix class: Petra_RDP_SPD_DenseMatrix – Similar to Petra_RDP_DenseMatrix except focused specifically on symmetric definite systems.

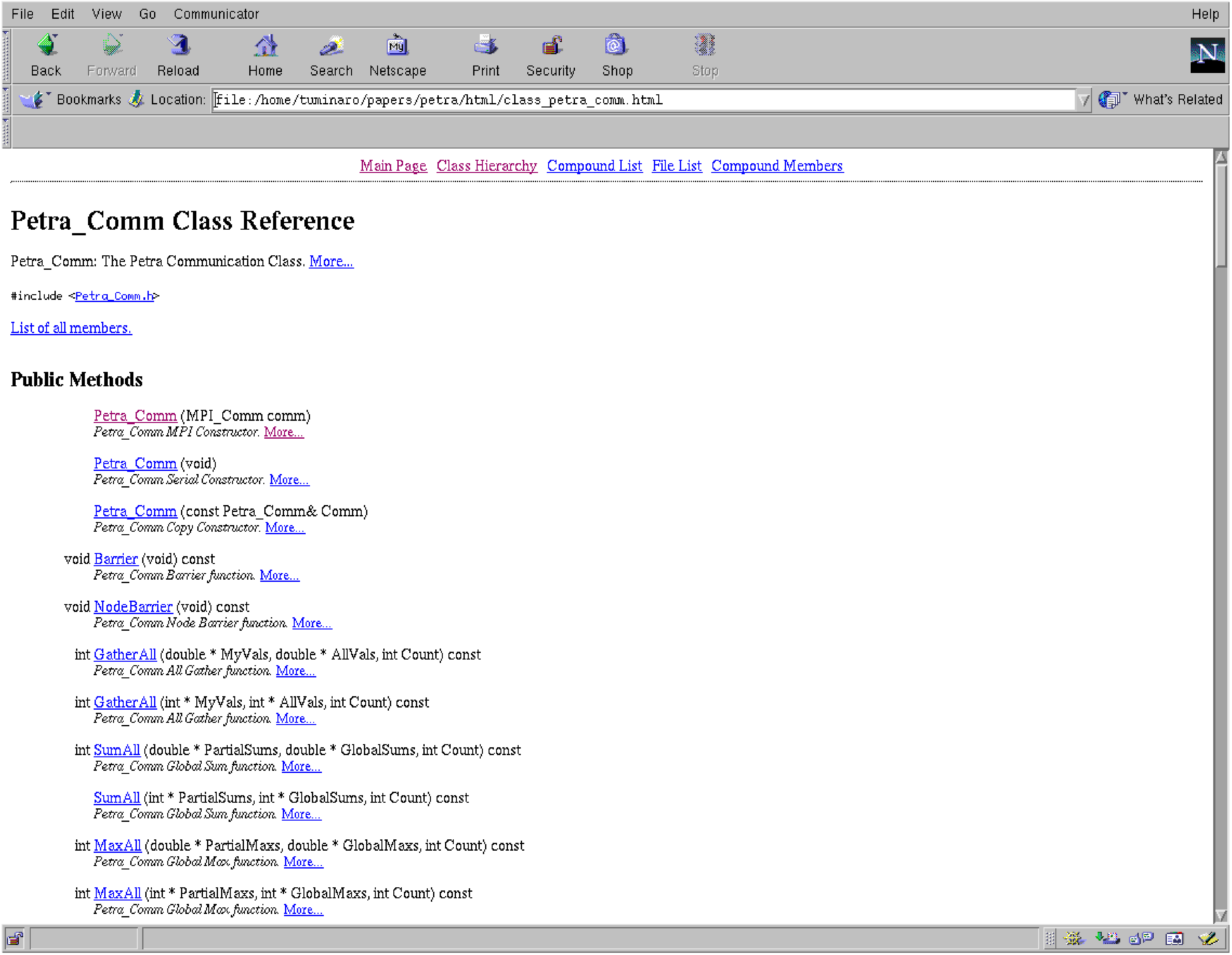
- Utility classes.

  1. Timing class: Petra_Time – Provides timing functions for the purposes of performance analysis.

  2. Floating point operation class: Petra_Flops – Provides floating point operations (FLOPS) counting and reporting functions for the purposes of performance analysis. All Petra computational classes accumulate FLOP counts associated with the *this* object of the computations.

  3. Distributed directory class: Petra_Directory – Allows construction of a distributed directory. Once constructed, a directory allows one to access randomly distributed objects in an efficient, scalable manner. This class is intended for support of general Petra_BlockMap and Petra_Map objects, but is useful in other settings as well.

  4. BLAS wrapper class: Petra_BLAS – A "thin" layer of C++ code wrapping the basic linear algebra subprograms (BLAS). This class provides a single instance interface between



File:/home/tuminaro/papers/petra/html/class_petra_comm.html

Main Page Class Hierarchy Compound List File List Compound Members

Petra_Comm Class Reference

Petra_Comm: The Petra Communication Class. More...

```
#include <Petra_Comm.h>
```

List of all members.

Public Methods

Petra_Comm (MPI_Comm comm)
Petra_Comm MPI Constructor. More...

Petra_Comm (void)
Petra_Comm Serial Constructor. More...

Petra_Comm (const Petra_Comm& Comm)
Petra_Comm Copy Constructor. More...

void Barrier (void) const
Petra_Comm Barrier function. More...

void NodeBarrier (void) const
Petra_Comm Node Barrier function. More...

int GatherAll (double * MyVals, double * AllVals, int Count) const
Petra_Comm All Gather function. More...

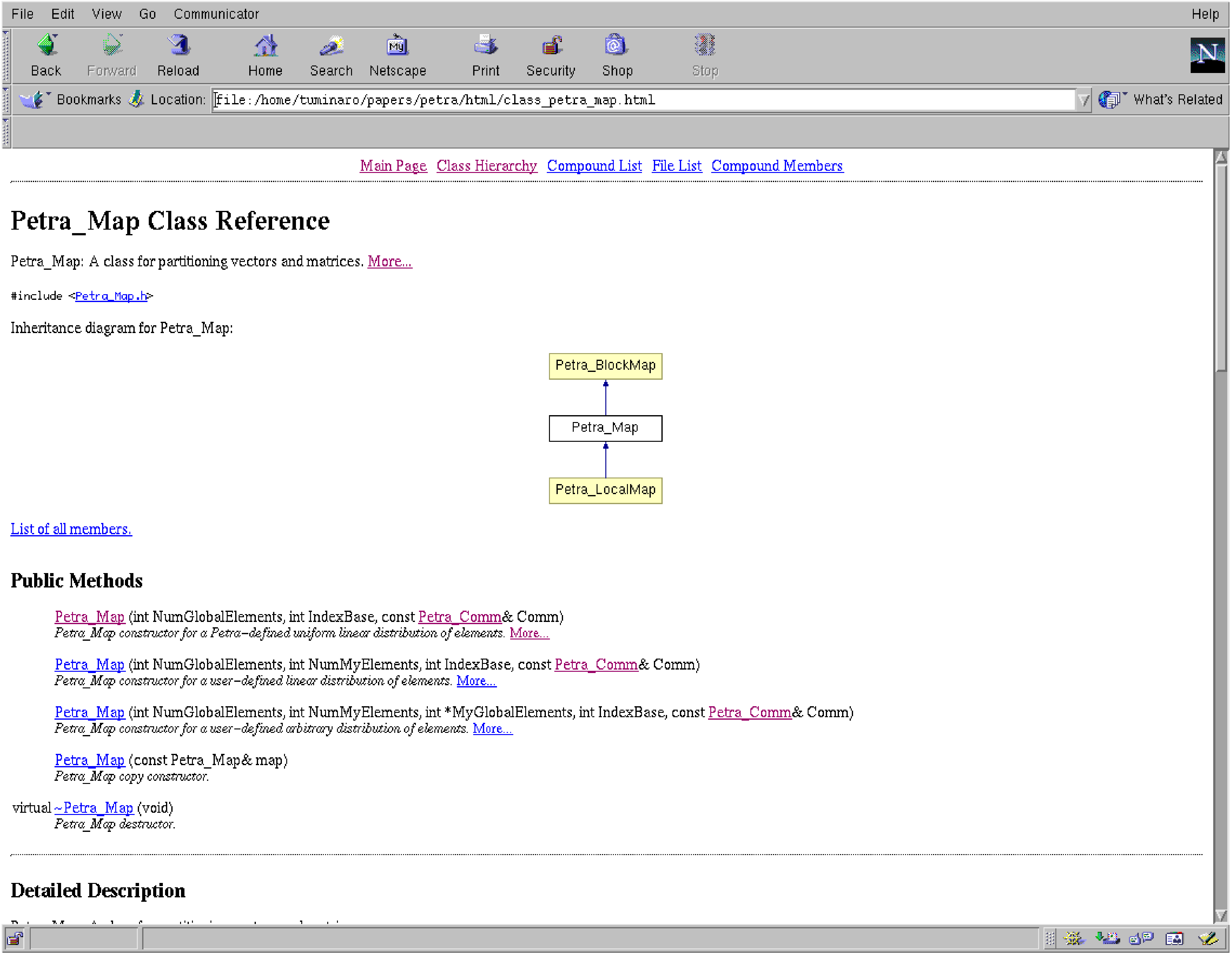
int GatherAll (int * MyVals, int * AllVals, int Count) const
Petra_Comm All Gather function. More...

int SumAll (double * PartialSums, double * GlobalSums, int Count) const
Petra_Comm Global Sum function. More...

SumAll (int * PartialSums, int * GlobalSums, int Count) const
Petra_Comm Global Sum function. More...

int MaxAll (double * PartialMaxs, double * GlobalMaxs, int Count) const
Petra_Comm Global Max function. More...

int MaxAll (int * PartialMaxs, int * GlobalMaxs, int Count) const
Petra_Comm Global Max function. More...



File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: file:/home/tuminaro/papers/petra/html/class_petra_map.html
What's Related

Main Page Class Hierarchy Compound List File List Compound Members

Petra_Map Class Reference

Petra_Map: A class for partitioning vectors and matrices. More...

```
#include <Petra_Map.h>
```

Inheritance diagram for Petra_Map:

Petra_BlockMap
Petra_Map
Petra_LocalMap

List of all members.

Public Methods

Petra_Map (int NumGlobalElements, int IndexBase, const Petra_Comm& Comm)
Petra_Map constructor for a Petra–defined uniform linear distribution of elements. More...

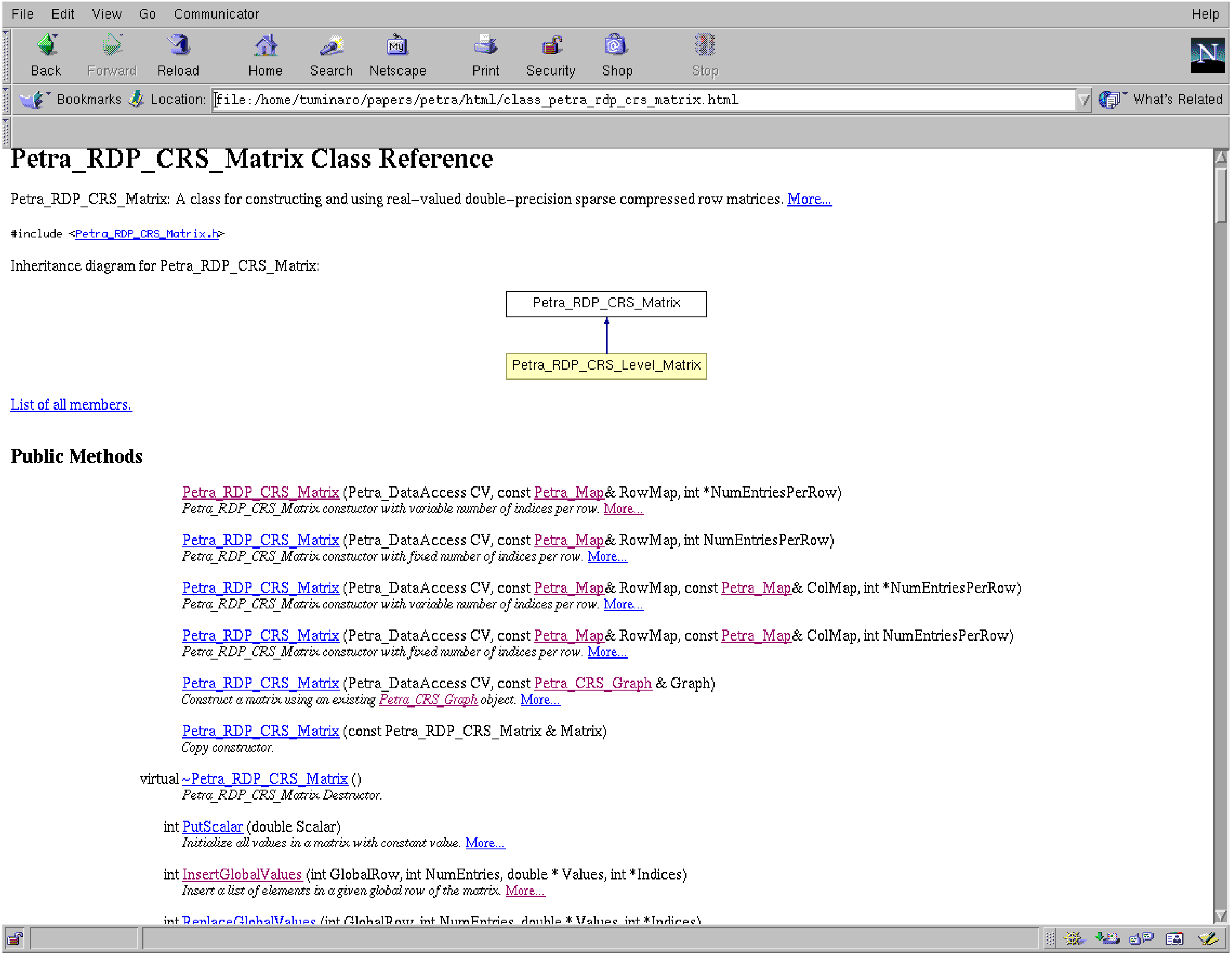
Petra_Map (int NumGlobalElements, int NumMyElements, int IndexBase, const Petra_Comm& Comm)
Petra_Map constructor for a user–defined linear distribution of elements. More...

Petra_Map (int NumGlobalElements, int NumMyElements, int *MyGlobalElements, int IndexBase, const Petra_Comm& Comm)
Petra_Map constructor for a user–defined arbitrary distribution of elements. More...

Petra_Map (const Petra_Map& map)
Petra_Map copy constructor.

virtual ~Petra_Map (void)
Petra_Map destructor.

Detailed Description


File Edit View Go Communicator Help
Back
Forward
Reload
Home
Search
Netscape
Print
Security
Shop
Stop
Bookmarks
Location: file:/home/tuminaro/papers/petra/html/class_petra_rdp_crs_matrix.html
What's Related
Petra_RDP_CRS_Matrix
Petra_RDP_CRS_Level_Matrix

Petra_RDP_CRS_Matrix Class Reference

Petra_RDP_CRS_Matrix: A class for constructing and using real–valued double–precision sparse compressed row matrices. More...

`#include <Petra_RDP_CRS_Matrix.h>`

Inheritance diagram for Petra_RDP_CRS_Matrix:

List of all members.

Public Methods

Petra_RDP_CRS_Matrix (Petra_DataAccess CV, const Petra_Map& RowMap, int *NumEntriesPerRow)
Petra_RDP_CRS_Matrix constuctor with variable number of indices per row. More...

Petra_RDP_CRS_Matrix (Petra_DataAccess CV, const Petra_Map& RowMap, int NumEntriesPerRow)
Petra_RDP_CRS_Matrix constuctor with fixed number of indices per row. More...

Petra_RDP_CRS_Matrix (Petra_DataAccess CV, const Petra_Map& RowMap, const Petra_Map& ColMap, int *NumEntriesPerRow)
Petra_RDP_CRS_Matrix constuctor with variable number of indices per row. More...

Petra_RDP_CRS_Matrix (Petra_DataAccess CV, const Petra_Map& RowMap, const Petra_Map& ColMap, int NumEntriesPerRow)
Petra_RDP_CRS_Matrix constuctor with fixed number of indices per row. More...

Petra_RDP_CRS_Matrix (Petra_DataAccess CV, const Petra_CRS_Graph & Graph)
Construct a matrix using an existing Petra_CRS_Graph object. More...

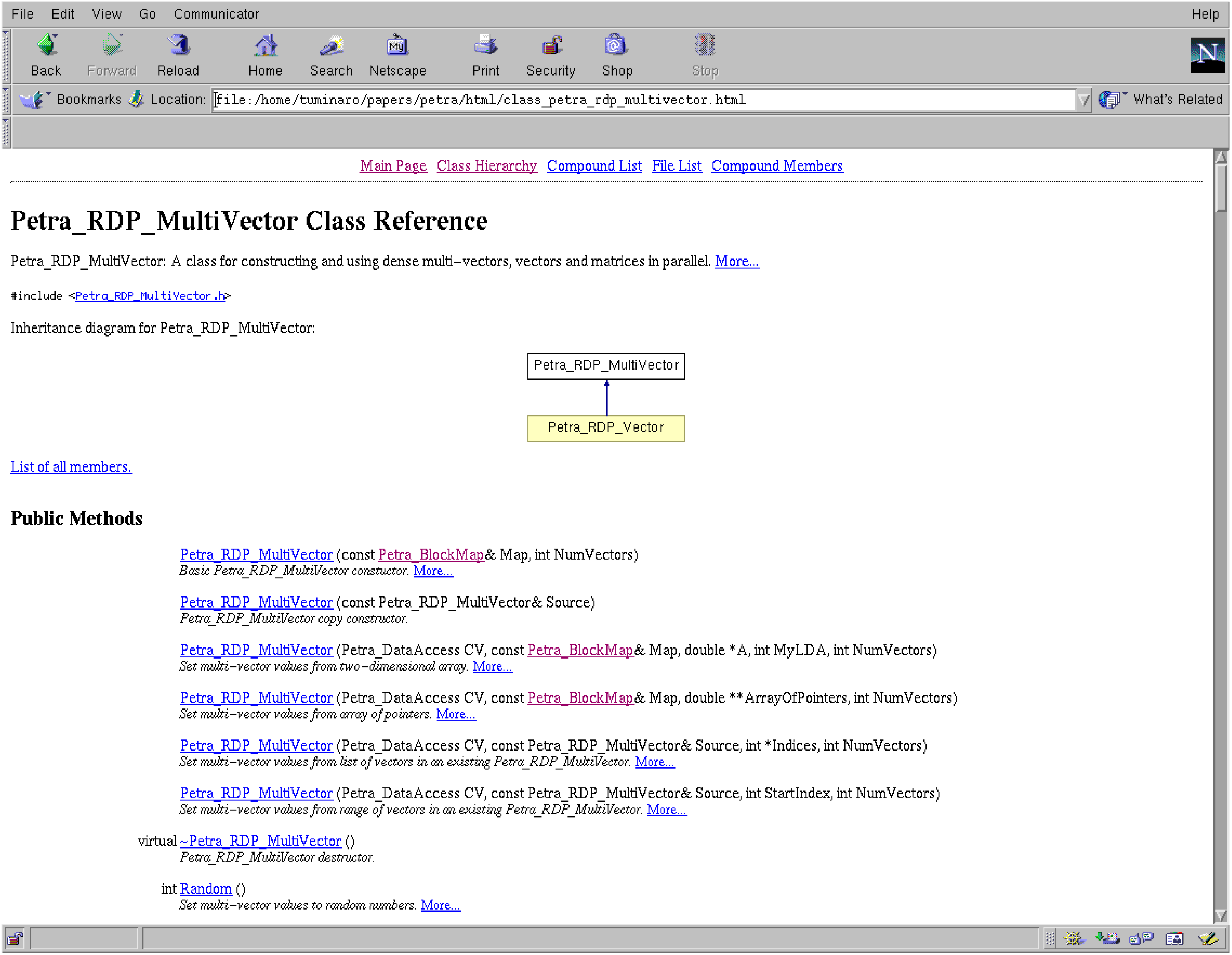
Petra_RDP_CRS_Matrix (const Petra_RDP_CRS_Matrix & Matrix)
Copy constructor.

virtual ~Petra_RDP_CRS_Matrix ()
Petra_RDP_CRS_Matrix Destructor.

int PutScalar (double Scalar)
Initialize all values in a matrix with constant value. More...

int InsertGlobalValues (int GlobalRow, int NumEntries, double * Values, int *Indices)
Insert a list of elements in a given global row of the matrix. More...

int ReplaceGlobalValues (int GlobalRow, int NumEntries, double * Values, int *Indices)


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: file:/home/tuminaro/papers/petra/html/class_petra_rdp_multivector.html
What's Related

Main Page Class Hierarchy Compound List File List Compound Members

Petra_RDP_MultiVector Class Reference

Petra_RDP_MultiVector: A class for constructing and using dense multi–vectors, vectors and matrices in parallel. More...

`#include <Petra_RDP_MultiVector.h>`

Inheritance diagram for Petra_RDP_MultiVector:

Petra_RDP_MultiVector
Petra_RDP_Vector

List of all members.

Public Methods

Petra_RDP_MultiVector (const Petra_BlockMap& Map, int NumVectors)
Basic Petra_RDP_MultiVector constuctor. More...

Petra_RDP_MultiVector (const Petra_RDP_MultiVector& Source)
Petra_RDP_MultiVector copy constructor.

Petra_RDP_MultiVector (Petra_DataAccess CV, const Petra_BlockMap& Map, double *A, int MyLDA, int NumVectors)
Set multi–vector values from two–dimensional array. More...

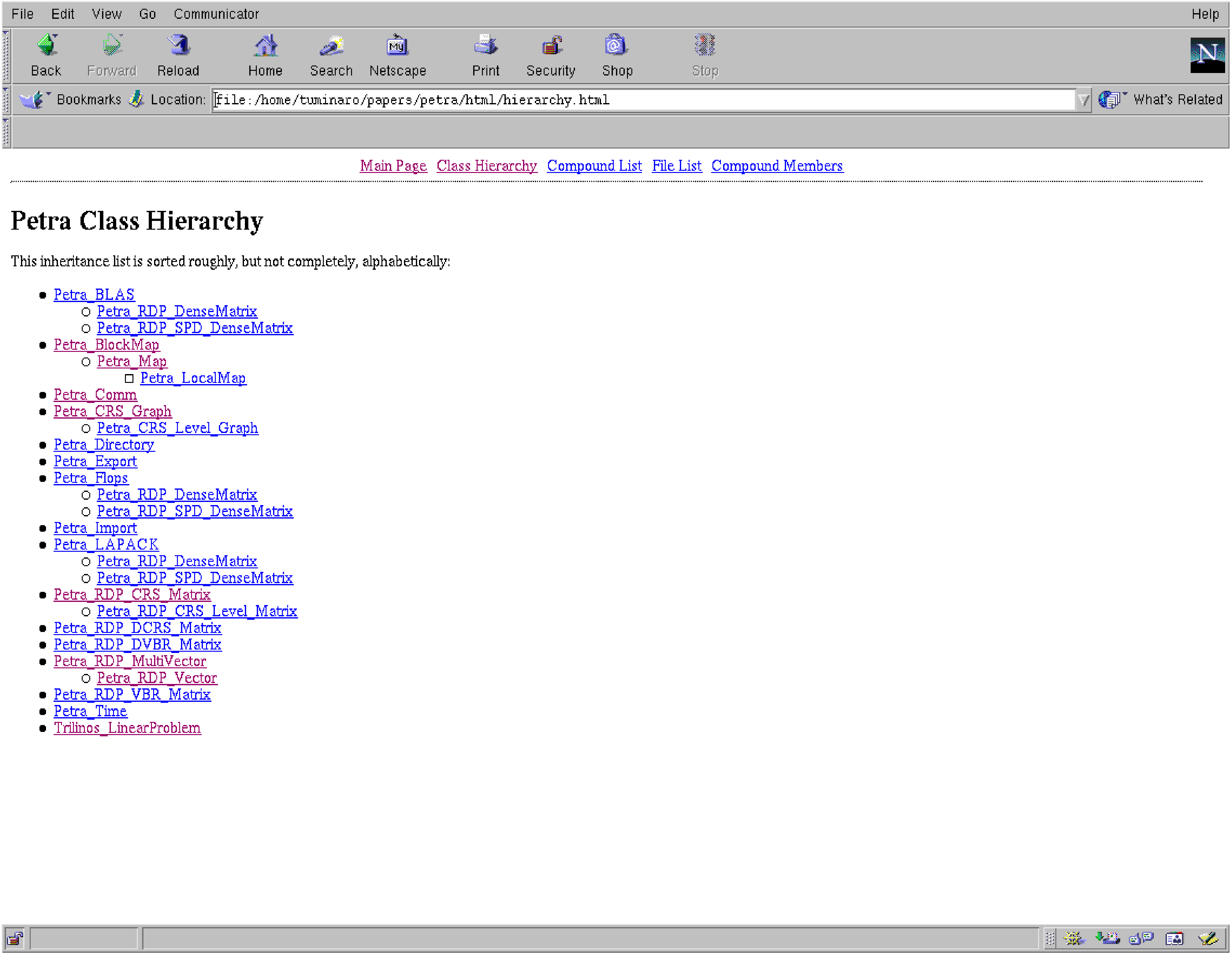
Petra_RDP_MultiVector (Petra_DataAccess CV, const Petra_BlockMap& Map, double **ArrayOfPointers, int NumVectors)
Set multi–vector values from array of pointers. More...

Petra_RDP_MultiVector (Petra_DataAccess CV, const Petra_RDP_MultiVector& Source, int *Indices, int NumVectors)
Set multi–vector values from list of vectors in an existing Petra_RDP_MultiVector. More...

Petra_RDP_MultiVector (Petra_DataAccess CV, const Petra_RDP_MultiVector& Source, int StartIndex, int NumVectors)
Set multi–vector values from range of vectors in an existing Petra_RDP_MultiVector. More...

virtual ~Petra_RDP_MultiVector ()
Petra_RDP_MultiVector destructor.

int Random ()
Set multi–vector values to random numbers. More...

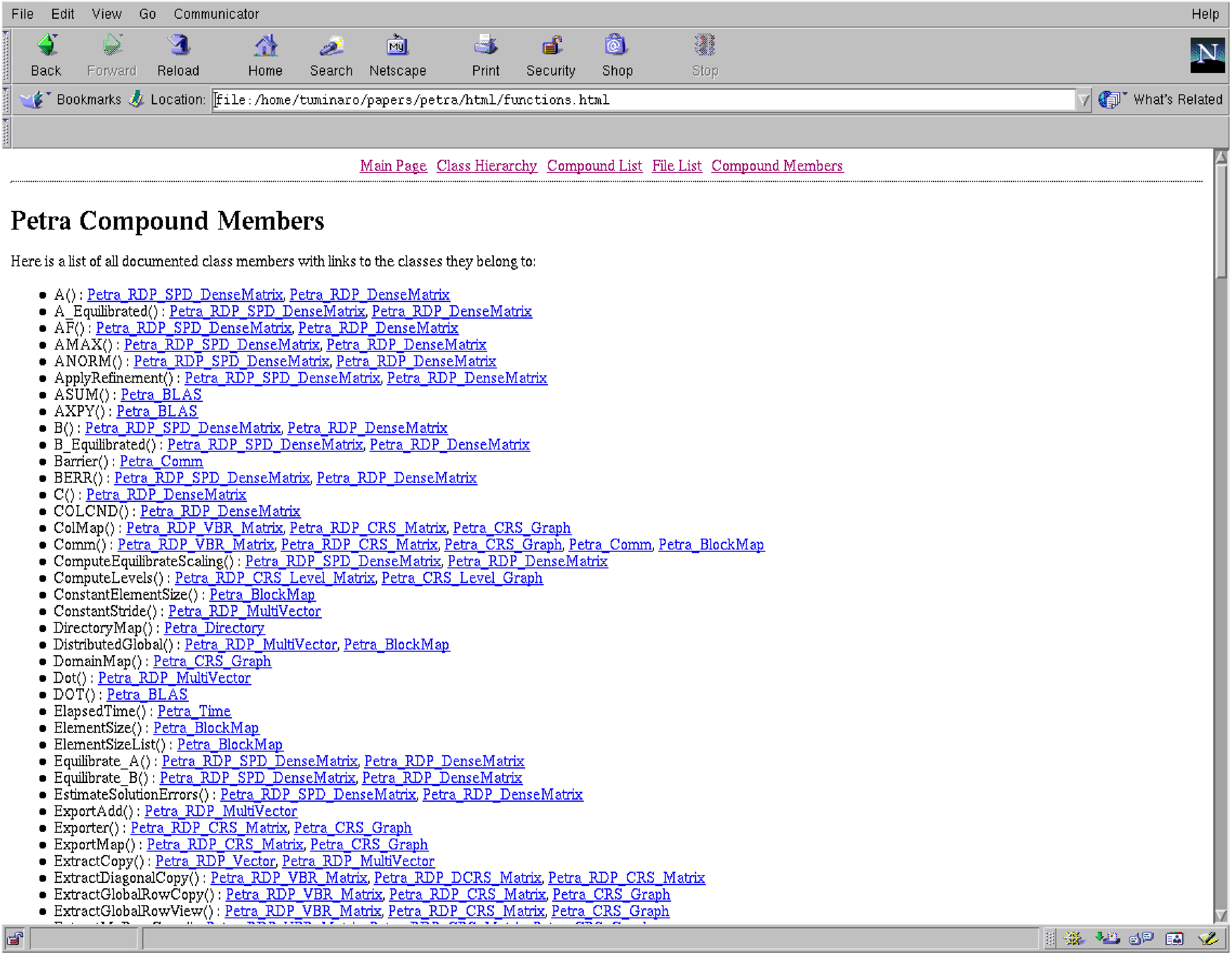


Main Page Class Hierarchy Compound List File List Compound Members

Petra Class Hierarchy

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

- Petra_BLAS
    - Petra_RDP_DenseMatrix
    - Petra_RDP_SPD_DenseMatrix
- Petra_BlockMap
    - Petra_Map
        - Petra_LocalMap
- Petra_Comm
- Petra_CRS_Graph
    - Petra_CRS_Level_Graph
- Petra_Directory
- Petra_Export
- Petra_Flops
    - Petra_RDP_DenseMatrix
    - Petra_RDP_SPD_DenseMatrix
- Petra_Import
- Petra_LAPACK
    - Petra_RDP_DenseMatrix
    - Petra_RDP_SPD_DenseMatrix
- Petra_RDP_CRS_Matrix
    - Petra_RDP_CRS_Level_Matrix
- Petra_RDP_DCRS_Matrix
- Petra_RDP_DVBR_Matrix
- Petra_RDP_MultiVector
    - Petra_RDP_Vector
- Petra_RDP_VBR_Matrix
- Petra_Time
- Trilinos_LinearProblem


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: File:/home/tuminaro/papers/petra/html/functions.html
What's Related

Main Page Class Hierarchy Compound List File List Compound Members

Petra Compound Members

Here is a list of all documented class members with links to the classes they belong to:

- A() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- A_Equilibrated() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- AF() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- AMAX() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- ANORM() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- ApplyRefinement() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- ASUM() : Petra_BLAS
- AXPY() : Petra_BLAS
- B() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- B_Equilibrated() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- Barrier() : Petra_Comm
- BERR() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- C() : Petra_RDP_DenseMatrix
- COLCND() : Petra_RDP_DenseMatrix
- ColMap() : Petra_RDP_VBR_Matrix, Petra_RDP_CRS_Matrix, Petra_CRS_Graph
- Comm() : Petra_RDP_VBR_Matrix, Petra_RDP_CRS_Matrix, Petra_CRS_Graph, Petra_Comm, Petra_BlockMap
- ComputeEquilibrateScaling() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- ComputeLevels() : Petra_RDP_CRS_Level_Matrix, Petra_CRS_Level_Graph
- ConstantElementSize() : Petra_BlockMap
- ConstantStride() : Petra_RDP_MultiVector
- DirectoryMap() : Petra_Directory
- DistributedGlobal() : Petra_RDP_MultiVector, Petra_BlockMap
- DomainMap() : Petra_CRS_Graph
- Dot() : Petra_RDP_MultiVector
- DOT() : Petra_BLAS
- ElapsedTime() : Petra_Time
- ElementSize() : Petra_BlockMap
- ElementSizeList() : Petra_BlockMap
- Equilibrate_A() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- Equilibrate_B() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- EstimateSolutionErrors() : Petra_RDP_SPD_DenseMatrix, Petra_RDP_DenseMatrix
- ExportAdd() : Petra_RDP_MultiVector
- Exporter() : Petra_RDP_CRS_Matrix, Petra_CRS_Graph
- ExportMap() : Petra_RDP_CRS_Matrix, Petra_CRS_Graph
- ExtractCopy() : Petra_RDP_Vector, Petra_RDP_MultiVector
- ExtractDiagonalCopy() : Petra_RDP_VBR_Matrix, Petra_RDP_DCRS_Matrix, Petra_RDP_CRS_Matrix
- ExtractGlobalRowCopy() : Petra_RDP_VBR_Matrix, Petra_RDP_CRS_Matrix, Petra_CRS_Graph
- ExtractGlobalRowView() : Petra_RDP_VBR_Matrix, Petra_RDP_CRS_Matrix, Petra_CRS_Graph

Summary

- Aztec 2.1
  - < http://www.cs.sandia.gov/CRF/aztec1.html>
  - double & complex
  - several iterative solvers
  - several (mostly algebraic) preconditioners

- ML within Sandia
  - algebraic multigrid

- Trilinos/Petra within Sandia
  - improved interface
  - several new features
    - ∗ block iterative solvers
    - ∗ eigenvalues