Documentation for module scipy.linsolve.umfpack

gendocs.py

November 28, 2007

Contents

1 Module scipy.linsolve.umfpack

1.1 Interface to the umfpack library

Contains: UmfpackContext class

1.1.1 Description

Routines for symbolic and numeric LU factorization of sparse matrices and for solving systems of linear equations with sparse matrices.
Tested with UMFPACK V4.4 (Jan. 28, 2005), V5.0 (May 5, 2006) Copyright (c) 2005 by Timothy A. Davis. All Rights Reserved. UMFPACK homepage: http://www.cise.ufl.edu/research/sparse/umfpack
Use 'print UmfpackContext().funs' to see all UMFPACK library functions the module exposes, if you need something not covered by the examples below.

1.1.2 Installation

Example site.cfg entry:

```
UMFPACK v4.4 in <dir>:

[amd]
library_dirs = <dir>/UMFPACK/AMD/Lib
include_dirs = <dir>/UMFPACK/AMD/Include
amd_libs = amd

[umfpack]
library_dirs = <dir>/UMFPACK/UMFPACK/Lib
include_dirs = <dir>/UMFPACK/UMFPACK/Include
umfpack_libs = umfpack

UMFPACK v5.0 (as part of UFsparse package) in <dir>:

[amd]
library_dirs = <dir>/UFsparse/AMD/Lib
include_dirs = <dir>/UFsparse/AMD/Include, <dir>/UFsparse/UFconfig
amd_libs = amd

[umfpack]
library_dirs = <dir>/UFsparse/UMFPACK/Lib
include_dirs = <dir>/UFsparse/UMFPACK/Include, <dir>/UFsparse/UFconfig
umfpack_libs = umfpack
```

1.1.3 Examples

Assuming this module imported as um (import scipy.linsolve.umfpack as um)
Sparse matrix in CSR or CSC format: mtx Righthand-side: rhs Solution: sol

```
# Contruct the solver.
umfpack = um.UmfpackContext() # Use default 'di' family of UMFPACK routines.

# One-shot solution.
sol = umfpack( um.UMFPACK_A, mtx, rhs, autoTranspose = True )
# same as:
sol = umfpack.linsolve( um.UMFPACK_A, mtx, rhs, autoTranspose = True )
```

-or-

```
# Make LU decomposition.
umfpack.numeric( mtx )
...
# Use already LU-decomposed matrix.
sol1 = umfpack( um.UMFPACK_A, mtx, rhs1, autoTranspose = True )
sol2 = umfpack( um.UMFPACK_A, mtx, rhs2, autoTranspose = True )
# same as:
sol1 = umfpack.solve( um.UMFPACK_A, mtx, rhs1, autoTranspose = True )
sol2 = umfpack.solve( um.UMFPACK_A, mtx, rhs2, autoTranspose = True )
```

-or-

```
# Make symbolic decomposition.
umfpack.symbolic( mtx0 )
# Print statistics.
umfpack.report_symbolic()

# ...

# Make LU decomposition of mtx1 which has same structure as mtx0.
umfpack.numeric( mtx1 )
# Print statistics.
umfpack.report_numeric()

# Use already LU-decomposed matrix.
sol1 = umfpack( um.UMFPACK_A, mtx1, rhs1, autoTranspose = True )

# ...

# Make LU decomposition of mtx2 which has same structure as mtx0.
umfpack.numeric( mtx2 )
sol2 = umfpack.solve( um.UMFPACK_A, mtx2, rhs2, autoTranspose = True )

# Print all statistics.
umfpack.report_info()
```

-or-

```
# Get LU factors and permutation matrices of a matrix.
L, U, P, Q, R, do_recip = umfpack.lu( mtx )
```

Returns:

'L'	...	Lower triangular m-by-min(m,n) CSR matrix
'U'	...	Upper triangular min(m,n)-by-n CSC matrix
'P'	...	Vector of row permuations
'Q'	...	Vector of column permuations
'R'	...	Vector of diagonal row scalings
'do_recip'	...	boolean

Note:
For a given matrix A, the decomposition satisfies: $LU = PRAQ$ when do_recip is true, $LU = P(R^{-1})AQ$ when do_recip is false

1.1.4 Umfpackcontext solution methods

umfpack(), umfpack.linsolve(), umfpack.solve()
Parameters:

'sys'	...	constant, one of UMFPACK system description constants, like UMFPACK_A, UMFPACK_At, see umfSys list and UMFPACK docs
'mtx'	...	sparse matrix (CSR or CSC)
'rhs'	...	right hand side vector
'autoTranspose'	...	bool automatically changes 'sys' to the transposed type, if 'mtx' is in CSR, since UMFPACK assumes CSC internally

1.1.5 Setting control parameters

Assuming this module imported as um:
List of control parameter names is accessible as 'um.umfControls' - their meaning and possible values are described in the UMFPACK documentation. To each name corresponds an attribute of the 'um' module, such as, for example 'um.UMFPACK_PRL' (controlling the verbosity of umfpack report functions). These attributes are in fact indices into the control array - to set the corresponding control array value, just do the following:

```
umfpack = um.UmfpackContext()
umfpack.control[um.UMFPACK_PRL] = 4 # Let's be more verbose.
```

–
Other contributors:
Nathan Bell (lu() method wrappers)

1.2 Functions