haasad / **PyPardisoProject** Public

Python interface to the Intel MKL Pardiso library to solve large sparse linear systems of equations

BSD-3-Clause license

71 stars **12** forks

Star Watch

Code Issues 4 Pull requests Actions Projects Wiki Secu

master

haasad Bump to v0.4.1 ···

on Apr 7 **171**

View code

README.md

Releases 5

pypardiso-tests passing

v0.4.1 Latest

on Apr 7

+ 4 releases

Used by 16

+ 8

Languages

- **Python** 100.0%

PyPardiso

PyPardiso is a python package to solve large sparse linear systems of equations with the Intel oneAPI Math Kernel Library PARDISO solver, a shared-memory multiprocessing parallel direct sparse solver.

PyPardiso provides the same functionality as SciPy's scipy.sparse.linalg.spsolve for solving the sparse linear system `Ax=b`. However in many cases it is significantly faster than SciPy's built-in single-threaded SuperLU solver.

PyPardiso is not a python interface to the PARDISO Solver from the PARDISO 7.2 Solver Project and it also doesn't currently support complex numbers. Check out JuliaSparse/Pardiso.jl for these more advanced use cases.

Installation

PyPardiso runs on Linux, Windows and MacOS. It can be installed with **conda** or **pip**. It is recommended to install PyPardiso using a virtual environment.

conda-forge	PyPI

conda-forge	PyPI
Anaconda.org 0.4.1	pypi package 0.4.1
`conda install -c conda-forge pypardiso`	`pip install pypardiso`

Basic usage

How to solve the sparse linear system `Ax=b` for `x`, where `A` is a square, sparse matrix in CSR (or CSC) format and `b` is a vector (or matrix):

```
In [1]: import pypardiso

In [2]: import numpy as np

In [3]: import scipy.sparse as sp

In [4]: A = sp.rand(10, 10, density=0.5, format='csr')

In [5]: A
Out[5]:
<10x10 sparse matrix of type '<class 'numpy.float64'>'
        with 50 stored elements in Compressed Sparse Row format>

In [6]: b = np.random.rand(10)

In [7]: x = pypardiso.spsolve(A, b)

In [8]: x
Out[8]:
array([ 0.02918389,  0.59629935,  0.33407289, -0.48788966,  3.44508841,
        0.52565687, -0.48420646,  0.22136413, -0.95464127,  0.58297397])
```