Call Fortran From Python Demo

Xiaozhou Li

2014/12/16

How it works

- compile a fortran source code to a python module

```
> f2py --fcompiler=gfortran -m python_module -c fortran_code.f90
```

Note: fwrap.

Example - Lapack

- compile the entire LAPACK library:

```
> f2py --fcompiler=gfortran -m lapack -c lapack-3.5.0/SRC/*.f -lblas
```

- example: dgesv function

```
> f2py --fcompiler=gfortran -m lapack -c dgesv.f -lblas
> python lapack_dgesv.py
dgesv(n,nrhs,a,ipiv,b,info,[lda,ldb])

Wrapper for ``dgesv``.

Parameters
----------
n : input int
nrhs : input int
a : input rank-2 array('d') with bounds (lda,*)
ipiv : input rank-1 array('i') with bounds (*)
b : input rank-2 array('d') with bounds (ldb,*)
info : input int

Other Parameters
----------------
lda : input int, optional
    Default: shape(a,0)
ldb : input int, optional
    Default: shape(b,0)

[[-4. ]
 [ 4.5]]
```

Example - Timing

Consider a heat equation:

$$\Delta u = 0.$$

A simple discretization:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = 0.$$

Solving this iteratively via the Gauss-Seidel method:

$$u_{i,j} \leftarrow \frac{(u_{i+1,j} + u_{i-1,j})\Delta y^2 + (u_{i,j+1} + u_{i,j-1})\Delta x^2}{2(\Delta x^2 + \Delta y^2)}$$

Size 80×80, iteration 10000, comparing python, fortran (f2py), C (ctypes) and C (cffi).

```
> python laplace.py
 python: 90.76527094841003 s
 f2py: 0.6033449172973633 s
 ctypes: 0.6915409564971924 s
 cffi: 0.6900129318237305 s
```

Example - DG animation

- Animation of DG approximation for a linear equation:

```
>f2py -m DG_linear -c dg.f90
>python DG_animation.py
```

Fortran 200x

- check about ISO_C_BINDING;
- treat it as C code: using ctypes or cffi.