The code for Krylov iterative methods

Introduction

The `scipy/sparse/linalg/isolve/` directory include `krylov.py` (updated and reimplemented iterative.py), `tests/test_krylov_poisson.py`, `tests/test_krylov_conv-diff.py` and required `tests/data/` directory

1. `krylov.py`: Re-implement Krylov iterative methods.
2. `test_krylov_poisson.py`: Test these Krylov methods for Poisson equations (iterations and CPU time).
3. `test_krylov_conv-diff.py`: Test BiCG, BiCGSTAB and GMRES for convection-diffusion equations.
4. `data/`: Stiffness matrix and right-hand side in linear system obtained by linear finite element method [] on structured mesh (for Poisson equations with homogeneous Dirichlet boundary condition and divergence-free convection-diffusion equations)

    > using P1 element for Poisson and [P1, P1] element for convection-diffusion

- `data/poisson_mat_128x128.dat`: stiffness matrix obtained by discrete Poisson equations on 128x128 grid (via change `nn = 128` in line 30 in benchmark_poisson.py)
- `data/poisson_rhs_128x128.dat`: right-hand side obtained by discrete Poisson equations on 128x128 grid
- `data/poisson_mat_256x256.dat`: stiffness matrix obtained by discrete Poisson equations on 256x256 grid
- `data/poisson_rhs_256x256.dat`: right-hand side obtained by discrete Poisson equations on 256x256 grid
- `data/conv-diff_mat_128x128.dat`: stiffness matrix obtained by discrete convection-diffusion equations on 128x128 grid (via change `nn = 128` in line 28 in benchmark_cd.py)
- `data/conv-diff_rhs_128x128.dat`: right-hand side obtained by discrete convection-diffusion equations on 128x128 grid
- `data/conv-diff_mat_256x256.dat`: stiffness matrix obtained by discrete convection-diffusion equations on 256x256 grid
- `data/conv-diff_rhs_256x256.dat`: right-hand side obtained by discrete convection-diffusion equations on 256x256 grid

Running

`python3 test_krylov_poisson.py` (for Poisson example)

`python3 test_krylov_conv-diff.py` (for convection-diffusion example)

Numerical Test

Example 1: `test_krylov_poisson.py`

Poisson equations in $\Omega := [0,1]^2$:

$$\begin{aligned} -\Delta u = f, &\quad \text{in } \Omega, \\ u = 0, &\quad \text{on } \partial\Omega, \end{aligned}$$

where f is determined by constructing the following exact solution sample:

$$\begin{aligned} u &= \sin(2\pi x)\sin(2\pi y) \\ u_x &= 2\pi\cos(2\pi x)\sin(2\pi y) \\ u_y &= 2\pi\sin(2\pi x)\cos(2\pi y) \\ u_{xx} &= -4\pi^2 u \\ u_{yy} &= u_{xx} \\ f &= -u_{xx} - u_{yy} \end{aligned}$$

Example 2: `test_krylov_conv-diff.py`

Convection-diffusion equations in $\Omega := [0,1]^2$:

$$\begin{aligned} -\Delta \boldsymbol{u} + \boldsymbol{b}\cdot\nabla\boldsymbol{u} = \boldsymbol{f}, &\quad \text{in } \Omega \\ \nabla\cdot\boldsymbol{u} = 0, &\quad \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{u}_D, &\quad \text{on } \partial\Omega, \end{aligned}$$

where $\boldsymbol{u} = [u,v]^T$ is unknown, $\boldsymbol{b} = [1,0]^T$. $\boldsymbol{u}_D = [u_D, v_D]^T$, $\boldsymbol{f} = [f,g]^T$ are determined by constructing the following exact solution sample:

$$\begin{aligned} u &= \sin(2\pi x)\sin(2\pi y) \\ v &= \cos(2\pi x)\cos(2\pi y) \\ u_x &= 2\pi\cos(2\pi x)\sin(2\pi y) \\ u_y &= 2\pi\sin(2\pi x)\cos(2\pi y) \\ v_x &= -u_y, \quad v_y = -u_x \\ u_{xx} &= -4\pi^2 u, \quad u_{yy} = u_{xx} \\ v_{xx} &= -4\pi^2 v, \quad v_{yy} = v_{xx} \\ f &= -u_{xx} - u_{yy} + u_x \\ g &= -v_{xx} - v_{yy} + u_y \\ u_D &= u|_{\partial\Omega} \\ v_D &= v|_{\partial\Omega} \end{aligned}$$