

Test of the linear systems from the convection-diffusion equations

This package provides implementations of comparing BICGSTAB, GMRES, DQGMRES, FOM, DIOM, SCG, SWI, REGMRES, REFOM, RESCG, and DYNOSWIWP for solving the linear systems from the convection-diffusion equations:

$$Ax = b,$$

where $A \in \mathbb{R}^{n \times n}$ is unsymmetric positive definite.

Contents

- BICGSTAB
- REGMRES
- DQGMRES
- REFOM
- DIOM(m)
- RESCG
- SWI(m)
- DYNOSWIWP
- Test

BICGSTAB

Using Matlabs function “bicgstab” directly.

REGMRES

REGMRES (Restarted GMRES): Algorithm 6.11 in Yousef Saad’s “Iterative Methods for Sparse Linear System (2nd Edition)”

```
function [x, k, res, resvec] = regmres(A, b, restart, tol, x0, maxit)
```

- $[x] = \text{regmres}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .
- $[x] = \text{regmres}(A, b, \text{restart})$ specifies the restarted number. If restart is $[]$ then regmres uses the default, n
- $[x] = \text{regmres}(A, b, \text{restart}, \text{tol})$ specifies the tolerance of the method. If tol is $[]$ then regmres uses the default, $1e-6$.
- $[x] = \text{regmres}(A, b, \text{restart}, \text{tol}, x0)$ specifies the initial guess. If $x0$ is $[]$ then regmres uses the default, an all zero vector.
- $[x] = \text{regmres}(A, b, \text{restart}, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is $[]$ then regmres uses the default, 10000.

- $[x, k] = \text{regmres}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$.
- $[x, k, \text{res}] = \text{regmres}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{regmres}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

GMRES: set `restart = n` in the function “`regmres`”.

DQGMRES

DQGMRES (Direct Quasi-GMRES): Algorithms 6.6 and 6.13 in Yousef Saad’s “Iterative Methods for Sparse Linear System (2nd Edition)”

```
function [x, k, res, resvec] = dqgmres(A, b, m, tol, x0, maxit)
```

- $[x] = \text{dqgmres}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .
- $[x] = \text{dqgmres}(A, b, m)$ specifies the number of the sliding window. If m is `[]` then `dqgmres` uses the default, n .
- $[x] = \text{dqgmres}(A, b, m, \text{tol})$ specifies the tolerance of the method. If tol is `[]` then `dqgmres` uses the default, $1e-6$.
- $[x] = \text{dqgmres}(A, b, m, \text{tol}, x0)$ specifies the initial guess. If $x0$ is `[]` then `dqgmres` uses the default, an all zero vector.
- $[x] = \text{dqgmres}(A, b, m, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is `[]` then `dqgmres` uses the default, 10000.
- $[x, k] = \text{dqgmres}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$.
- $[x, k, \text{res}] = \text{dqgmres}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{dqgmres}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

REFOM

REFOM (Restarted FOM): Algorithm 6.5 in Yousef Saad’s “Iterative Methods for Sparse Linear System (2nd Edition)”

```
function [x, k, res, resvec] = refom(A, b, restart, tol, x0, maxit)
```

- $[x] = \text{refom}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .

- $[x] = \text{refom}(A, b, \text{restart})$ specifies the restarted number. If restart is $[]$ then refom uses the default, n
- $[x] = \text{refom}(A, b, \text{restart}, \text{tol})$ specifies the tolerance of the method. If tol is $[]$ then refom uses the default, 1e-6.
- $[x] = \text{refom}(A, b, \text{restart}, \text{tol}, x0)$ specifies the initial guess. If x0 is $[]$ then refom uses the default, an all zero vector.
- $[x] = \text{refom}(A, b, \text{restart}, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is $[]$ then refom uses the default, 10000.
- $[x, k] = \text{refom}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$.
- $[x, k, \text{res}] = \text{refom}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{refom}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

FOM: set restart= n in the function “refom”.

DIOM(m)

DIOM (Direct Incomplete Orthogonalization Method): Algorithms 6.6 and 6.8 in Yousef Saad’s “Iterative Methods for Sparse Linear System (2nd Edition)”

```
function [x, k, res, resvec] = diom(A, b, mk, tol, x0, maxit)
```

- $[x] = \text{diom}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .
- $[x] = \text{diom}(A, b, m)$ specifies the number of the sliding window. If m is $[]$ then diom uses the default, n.
- $[x] = \text{diom}(A, b, m, \text{tol})$ specifies the tolerance of the method. If tol is $[]$ then diom uses the default, 1e-6.
- $[x] = \text{diom}(A, b, m, \text{tol}, x0)$ specifies the initial guess. If x0 is $[]$ then diom uses the default, an all zero vector.
- $[x] = \text{diom}(A, b, m, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is $[]$ then diom uses the default, 10000.
- $[x, k] = \text{diom}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$.
- $[x, k, \text{res}] = \text{diom}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{diom}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

RESCG

RESCG (Restarted SCG): Restarted semi-conjugate gradient method

```
function [x, k, res, resvec] = rescg(A, b, restart, tol, x0, maxit)
```

- $[x] = \text{rescg}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .
- $[x] = \text{rescg}(A, b, \text{restart})$ specifies the restarted number. If restart is $[]$ then rescg uses the default, n .
- $[x] = \text{rescg}(A, b, \text{restart}, \text{tol})$ specifies the tolerance of the method. If tol is $[]$ then rescg uses the default, $1e-6$.
- $[x] = \text{rescg}(A, b, \text{restart}, \text{tol}, x0)$ specifies the initial guess. If $x0$ is $[]$ then rescg uses the default, an all zero vector.
- $[x] = \text{rescg}(A, b, \text{restart}, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is $[]$ then rescg uses the default, 10000.
- $[x, k] = \text{rescg}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$.
- $[x, k, \text{res}] = \text{rescg}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{rescg}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

SCG: set $\text{restart}=n$ in the function “ rescg ”.

SWI(m)

SWI: Sliding window implementation with pre-allocated memory

```
[x, k, res, resvec] = swi(A, b, m, tol, x0, maxit)
```

- $[x] = \text{swi}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .
- $[x] = \text{swi}(A, b, m)$ specifies the number of the sliding window. If m is $[]$ then swi uses the default, n .
- $[x] = \text{swi}(A, b, m, \text{tol})$ specifies the tolerance of the method. If tol is $[]$ then swi uses the default, $1e-6$.
- $[x] = \text{swi}(A, b, m, \text{tol}, x0)$ specifies the initial guess. If $x0$ is $[]$ then swi uses the default, an all zero vector.
- $[x] = \text{swi}(A, b, m, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is $[]$ then swi uses the default, 10000.
- $[x, k] = \text{swi}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$.
- $[x, k, \text{res}] = \text{swi}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{swi}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

DYNSWIWP

DYNSWIWP: Sliding window implementation with choosing m dynamically

The approach to choose m dynamically is as follows:

$$m = \begin{cases} \max\{[0.8m], 2\}, & \text{if } \|r_k\| < 0.9\|r_{k-1}\|; \\ \min\{2m, n\}, & \text{if } \|r_k\| > 2\|r_{k-1}\|; \\ m, & \text{otherwise.} \end{cases}$$

```
[x, k, res, resvec] = dynswiwp(A, b, m, tol, x0, maxit)
```

- $[x] = \text{dynswiwp}(A, b)$ attempts to find a solution x to the system of linear equations $Ax = b$. The n -by- n coefficient matrix A must be positive definite but need not be symmetric. The right hand side column vector b must have length n .
- $[x] = \text{dynswiwp}(A, b, m)$ specifies the number of the sliding window. If m is $[]$ then `dynswiwp` uses the default, n .
- $[x] = \text{dynswiwp}(A, b, m, \text{tol})$ specifies the tolerance of the method. If tol is $[]$ then `dynswiwp` uses the default, $1e-6$.
- $[x] = \text{dynswiwp}(A, b, m, \text{tol}, x0)$ specifies the initial guess. If $x0$ is $[]$ then `dynswiwp` uses the default, an all zero vector.
- $[x] = \text{dynswiwp}(A, b, m, \text{tol}, x0, \text{maxit})$ specifies the maximum number of iterations. If maxit is $[]$ then `dynswiwp` uses the default, 10000.
- $[x, k] = \text{dynswiwp}(A, b, \dots)$ returns the iteration number at which x was computed: $1 \leq k \leq \text{maxit}$
- $[x, k, \text{res}] = \text{dynswiwp}(A, b, \dots)$ also returns the last relative residual norm $\|b - Ax\|/\|b\|$.
- $[x, k, \text{res}, \text{resvec}] = \text{dynswiwp}(A, b, \dots)$ also returns a vector of estimates of the residual norms at each iteration, including $\|b - Ax\|$.

BICGSTAB

Using Matlabs function “bicgstab” directly.

Test

1. Download the files: `regmres.m`, `refom.m`, `rescg.m`, `dqgmres.m`, `diom.m`, `swi.m`, `dynswiwp.m`, `test_condiff.m`, `A.mat` and `b.mat`;
2. Run the function “`test_condiff`” directly, i.e.,

```
>> test_condiff
```

BICGSTAB	130	0.0045	5.36e-08
GMRES	43	0.0114	4.17e-07
DQGMRES(36)	43	0.0132	5.81e-07
FOM	43	0.0232	4.47e-07
DIOM(2)	5751	0.2528	NaN
DIOM(5)	49	0.0119	8.25e-07
DIOM(10)	60	0.0226	2.33e-07
SCG	43	0.0052	4.47e-07
SWI(2)	71	0.0116	8.28e-07
SWI(5)	52	0.0090	1.58e-07
SWI(10)	63	0.0075	3.50e-07
RGMRES	123	0.0147	7.18e-07
RFOM	85	0.0190	9.29e-07
RSCG	84	0.0103	9.91e-07
DSWI	69	0.0158	6.37e-07