KELDYSH INSTITUTE OF APPLIED MATHEMATICS (RUSSIAN ACADEMY OF SCIENCES), 2012

GMRES(m) algorithms implementation

Nikita Kokonkov
kknkoff@gmail.com

Introduction

To solve the linear equations system

$$\mathbf{A}\boldsymbol{x} = \mathbf{F} \tag{1}$$

of the size N the algorithms of the generalized minimal residual over Krylov subspace methods with restarting [1-2] have been realized. The algorithms are implemented in FORTRAN 77 with OpenMP. For the sake of the efficiency the code is not idiot-proof and so it does not check its dummy arguments for correctness and permute rows of the $\mathbf{A}$ matrix. Hence user has to look after his input by himself.

Subroutines

There are two kinds of subroutines for two data types and two matrix compression schemes. The subroutines names according to their kinds, data types, and compression schemes are tabulated below.

Storage scheme		ELLPACK		CSC	
Preconditioned		Yes	No	Yes	No
Subroutine	`DOUBLE PRECISION`	`KDPJ`	`KDP`	`KCDPJ`	`KCDP`
	`DOUBLE COMPLEX`	`KDCJ`	`KDC`	`KCDCJ`	`KCDC`

Two kinds of subroutines are preconditioned and non-preconditioned ones. The preconditioned subroutines are somewhat faster than non-preconditioned ones over Krylov subspaces of the same sizes but requires non-zero left hand side matrix diagonal. Subroutines of `DOUBLE PRECISION` or `DOUBLE COMPLEX` data types are designed for solving the system (1) with matrices and vectors with real or complex elements respectively. They dummy arguments have the corresponding data types. For the $\mathbf{A}$ matrix compression is used storage scheme that resembles the ELLPACK format or the resembling CSC format one.

Storage schemes

The resembling ELLPACK and CSC formats storage schemes are based on applying that formats to the $\grave{\mathbf{A}}$ matrix that is the left hand side $\mathbf{A}$ matrix without its diagonal elements:

$$\grave{\mathbf{A}} = \mathbf{A} - \operatorname{diag}\left(a_{11}, \ldots, a_{NN}\right). \tag{2}$$

The left hand side matrix diagonal $\operatorname{diag}\left(a_{11}, \ldots, a_{NN}\right)$, solution $\boldsymbol{x}$ and right hand side $\mathbf{F}$ vectors are stored in three one-dimensional `DOUBLE PRECISION` (`DOUBLE COMPLEX`) N elements. The $\grave{\mathbf{A}}$ matrix is stored in notorious ELLPACK or CSC format in three arrays: one-dimensional `INTEGER` array of the N or the $N+1$ elements for ELLPACK or CSC formats respectively, two-dimensional `INTEGER` and `DOUBLE PRECISION` (`DOUBLE COMPLEX`) arrays of the `L` (the maximum number of the row non-zero elements in the $\grave{\mathbf{A}}$ matrix) rows and N columns or one-dimensional `INTEGER` and `DOUBLE PRECISION` (`DOUBLE COMPLEX`) arrays of the `L` (the number of non-zero elements in the $\grave{\mathbf{A}}$ matrix) elements each for the ELLPACK or CSC formats respectively.

For example sparse matrix

$$\mathbf{A} = \begin{pmatrix} 1.1 & 0.0 & 2.3 & 0.0 & 0.0 \\ 3.1 & 4.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 5.2 & 6.3 & 0.0 & 7.5 \\ 0.0 & 0.0 & 0.0 & 8.4 & 0.0 \\ 9.1 & 0.0 & 0.0 & 0.4 & 1.5 \end{pmatrix}. \quad (3)$$

could be presented by four arrays in

ELLPACK-like format

$$\mathtt{P} = \begin{bmatrix} \mathtt{1.1D0} \\ \mathtt{4.2D0} \\ \mathtt{6.3D0} \\ \mathtt{8.4D0} \\ \mathtt{1.5D0} \end{bmatrix} \quad \mathtt{LA} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 2 \end{bmatrix} \quad \mathtt{IA} = \begin{bmatrix} 3 & 1 & 5 & 0 & 1 \\ 0 & 0 & 2 & 0 & 4 \end{bmatrix} \quad \mathtt{A} = \begin{bmatrix} \mathtt{2.3D0} & \mathtt{3.1D0} & \mathtt{7.5D0} & \mathtt{0.0D0} & \mathtt{9.1D0} \\ \mathtt{0.0D0} & \mathtt{0.0D0} & \mathtt{5.2D0} & \mathtt{0.0D0} & \mathtt{0.4D0} \end{bmatrix} \quad (4)$$

CSC-like format

$$\mathtt{P} = \begin{bmatrix} \mathtt{1.1D0} \\ \mathtt{4.2D0} \\ \mathtt{6.3D0} \\ \mathtt{8.4D0} \\ \mathtt{1.5D0} \end{bmatrix} \quad \mathtt{LA} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 5 \\ 5 \\ 7 \end{bmatrix} \quad \mathtt{IA} = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 5 \\ 4 \\ 1 \end{bmatrix} \quad \mathtt{A} = \begin{bmatrix} \mathtt{2.3D0} \\ \mathtt{3.1D0} \\ \mathtt{5.2D0} \\ \mathtt{7.5D0} \\ \mathtt{0.4D0} \\ \mathtt{9.1D0} \end{bmatrix} \quad (5)$$

Interface

All subroutines despite difference in compression format have a similar dummy arguments list

`(M,N,L,LA,IA,A,F,P,U,T,V,H,G,E)`

where all arguments except the last four (that are temporary arrays) are the subroutine input:

`M` – the `INTEGER` maximum Krylov subspace size (m);

`N` – the `INTEGER` size of the equation system (1), that is number of left hand side matrix, unknowns, and right hand side vector rows;

`L` – the `INTEGER` maximum number of non-zero row elements in the left hand side matrix minus one or the number of non-zero elements in the left hand side matrix minus N for the ELLPACK-like or CSC-like formats respectively;

`LA` – the `INTEGER` N elements array of the non-zero row elements number in the packed left hand side matrix or the $N+1$ elements array of the packed left hand side matrix rows offsets for the ELLPACK-like or CSC-like formats respectively;

`IA` – the `INTEGER` `[L]` -size array of the elements positions in the packed left hand side matrix rows or rectangular `[L,N]` -size array of the elements positions in the packed left hand side matrix rows for the ELLPACK-like or CSC-like formats respectively;

`A` – the `DOUBLE PRECISION` (`DOUBLE COMPLEX`) `[L]` -size array of the packed left hand side matrix rows elements or rectangular `[L,N]` -size array of the packed left hand side matrix rows elements for the ELLPACK-like or CSC-like formats respectively;

`F` – the `DOUBLE PRECISION` (`DOUBLE COMPLEX`) N -size right hand size vector;

`P` – the `DOUBLE PRECISION` (`DOUBLE COMPLEX`) N elements array of the left hand side matrix diagonal elements;

`U` – the `DOUBLE PRECISION` (`DOUBLE COMPLEX`) N -size unknowns vector where on input should be placed the solution vector $\boldsymbol{x}$ initial guess;

`T` – the `DOUBLE PRECISION` numerical solution error tolerance;

`V` – the `DOUBLE PRECISION` (`DOUBLE COMPLEX`) `[N,M+1]` -size temporary array;

`H` – the `DOUBLE PRECISION` (`DOUBLE COMPLEX`) `[M*(M+1)/2]` -size temporary array;

G – the DOUBLE PRECISION (DOUBLE COMPLEX) [2,M] -size temporary array;
E – the DOUBLE PRECISION (DOUBLE COMPLEX) [M] -size temporary array.
The equation system (1) solution vector $\boldsymbol{x}$ would be placed in array U.

Practical recommendations

All arrays sizes should be greater than zero and match the arrays actual sizes. The maximum Krylov subspace size INTEGER M should be greater than zero and is majorized only by the memory available for the temporary arrays V, H, G, E storage. The error tolerance DOUBLE PRECISION T should be greater than machine precision that are depend on the bytes number in the DOUBLE PRECISION data type for the machine. The solution initial guess DOUBLE PRECISION (DOUBLE COMPLEX) U array is advisable to be close to the equation system (1) solution vector $\boldsymbol{x}$. Temporary arrays V, H, G, E could have any values before and after the subroutine calling.

Realization

Each subroutine code is placed in its own standalone *.F file that does not require any externals in contrast to the [3]. The subroutines could be downloaded by the links listed below:

KDP	http://cpt.imamod.ru/alg/krylov/KDP.F
KDC	http://cpt.imamod.ru/alg/krylov/KDC.F
KDPJ	http://cpt.imamod.ru/alg/krylov/KDPJ.F
KDCJ	http://cpt.imamod.ru/alg/krylov/KDCJ.F
KCDP	http://cpt.imamod.ru/alg/krylov/KCDP.F
KCDC	http://cpt.imamod.ru/alg/krylov/KCDC.F
KCDPJ	http://cpt.imamod.ru/alg/krylov/KCDPJ.F
KCDCJ	http://cpt.imamod.ru/alg/krylov/KCDCJ.F

Reference

1. Y. Saad, M. H. Schultz "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems" SIAM Journal on Scientific and Statistical Computing, Volume 7, Issue 3, 1986, Pages 856-869. DOI:10.1137/0907058
2. Y. Saad "A flexible inner-outer preconditioned GMRES algorithm" SIAM Journal on Scientific Computing, Volume 14, Issue 2, 1993, Pages 461-469. DOI:10.1137/0914028
3. R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods" Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994, 135 pages. DOI:10.1137/1.9781611971538

Document history

December 2012	Produced
November 2013	Appended
March 2015	Emendated
November 2015	Appended

Citation example

N. I. Kokonkov "GMRES(m) algorithms implementation", KIAM (RAS), 2012. http://cpt.imamod.ru/alg/krylov/cpt-2012_geminres.pdf