Primer for the Matlab Functions:
`GMRES.m`, `ellShiftGMRES.m`, `BGMRES.m`, `ellShiftBGMRES.m`, `glGMRES.m`, & `ellShiftGlGMRES.m`

Alessandro Buccini, Lucas Onisk, Lothar Reichel

1 Introduction

This primer describes six MATLAB functions for the iterative solution of large discrete linear and block linear ill-posed problems. The linear discrete ill-posed problem is given by

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}^{\delta}, \quad \boldsymbol{A} \in \mathbb{R}^{n\times n}, \quad \boldsymbol{x}, \boldsymbol{b}^{\delta} \in \mathbb{R}^{n}, \tag{1}$$

and the block linear ill-posed problem is given by

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{B}^{\delta}, \quad \boldsymbol{X}, \boldsymbol{B}^{\delta} \in \mathbb{R}^{n\times k}, \tag{2}$$

where $\boldsymbol{b}^{\delta}$ and $\boldsymbol{B}^{\delta}$ are contaminated with error. The functions `GMRES.m` and `ellShiftGMRES.m` are for problems of the form (1) and functions `BGMRES.m`, `ellShiftBGMRES.m`, `glGMRES.m`, and `ellShiftGlGMRES.m` are for problems of the form (2). These iterative methods implemented by these functions are described in [1], where several properties are discussed. This primer also describes several auxiliary MATLAB functions.

2 Files and Installation

The software is available from the publisher ETNA (Electronic Transactions on Numerical Analysis) and will be stored as a compressed file entitled `ellShiftedPkg.zip` accompanying paper [1]. Installation details are discussed in this primer as well as in the `README.md` file. The files in `ellShiftedPkg.zip` are listed in Table 1. All files should be extracted and placed in the same directory before use. The code has been developed and tested using MATLAB version R2020b by MathWorks. No other MathWorks products or toolboxes are required. To use the provided demos, specific test problems from the MATLAB package na33 by Neuman et. al [2, 3] should be available. These functions are freely available at `http://www.netlbib.org/numeralgo/` and are summarized in Table 2.

File	**Description**
`codePrimer.pdf`	This document.
`license.md`	Markdown file containing license for package use.
`README.md`	Markdown file with installation instructions and details similar to this primer document.
`GMRES.m`	The GMRES algorithm for linear discrete ill-posed problems with a square nonsymmetric matrix.
`ellShiftGMRES.m`	The ellShiftGMRES algorithm for linear discrete ill-posed problems with a square nonsymmetric matrix. This version allows the user to select the level of range restriction.
`linearSys_demo.m`	A demo to showcase the use of the GMRES and ellShiftGMRES algorithms on a linear discrete ill-posed problem.
`BGMRES.m`	The BGMRES algorithm for block linear discrete ill-posed problems with a square nonsymmetric matrix.
`ellShiftBGMRES.m`	The ellShiftBGMRES algorithm for linear discrete ill-posed problems with a square nonsymmetric matrix. This version allows the user to select the level of range restriction.
`glGMRES.m`	The glGMRES algorithm for block linear discrete ill-posed problems with a square nonsymmetric matrix.
`ellShiftGlGMRES.m`	The ellShiftGlGMRES algorithm for block linear discrete ill-posed problems with a square nonsymmetric matrix. This version allows the user to select the level of range restriction.
`blockLinearSys_demo.m`	A demo to showcase the use of the BGMRES, ellShiftBGMRES, glGMRES, and ellShiftGlGMRES algorithms on a block linear discrete ill-posed problem.

Table 1: Files in *ellShiftedPkg.zip*. All algorithms terminate according to the discrepancy principle.

File	**Description**
`shaw_alt.m`	Discretization of the Fredholm integral equation of the first kind described by Shaw in [4] using a Nyström method based on the composite trapezoidal rule with equidistant nodes. The resulting linear system has a square nonsymmetric matrix.
`phillips_alt.m`	Discretization of the Fredholm integral equation of the first kind described by Phillips in [5] using a Nyström method based on the composite trapezoidal rule with equidistant nodes. The resulting linear system has a square nonsymmetric matrix.

Table 2: Files from package na33 [2, 3] necessary to run the demos.

3 Input and Output Parameters for Functions

The input parameter syntax for the functions as well as their outputs is displayed in Table 3. A more detailed description of each input parameter as well as the outputs is provided in Table 4. All functions have the same output schemes.

File/Syntax
`GMRES.m` `[x,iter,rrnorm] = GMRES(A,b,maxIter,noiseLevel,eta)`
`ellShiftGMRES.m` `[x,iter,rrnorm] = ellShiftGMRES(A,b,ell,maxIter,noiseLevel,eta)`
`BGMRES.m` `[X,iter,rrnorm] = BGMRES(A,B,maxIter,noiseLevel,eta)`
`ellShiftBGMRES.m` `[X,iter,rrnorm] = ellShiftBGMRES(A,B,ell,maxIter,noiseLevel,eta)`
`glGMRES.m`

`[X,iter,rrnorm] = glGMRES(A,B,maxIter,noiseLevel,eta)`
`ellShiftGlGMRES.m` `[X,iter,rrnorm] = ellShiftGlGMRES(A,B,ell,maxIter,noiseLevel,eta)`

Table 3: Syntax for `GMRES`, `ellShiftGMRES`, `BGMRES`, `ellShiftBGMRES`, `glGMRES`, & `ellShiftGlGMRES` functions.

Input Parameters	**Description**
`A`	The $n \times n$ matrix of the linear discrete ill-posed problem (1) or block linear discrete ill-posed problem (2).
`b`	The right-hand side $n \times 1$ vector of (1).
`B`	The right-hand side $n \times k$ matrix of (2).
`ell`	A positive integer that defines the shift of the Krylov subspace where a solution is sought.
`maxIter`	The maximum number of iterations the algorithm will attempt to carryout if the discrepancy principle is not satisfied.
`noiseLevel`	The scaled noise level contaminating the right-hand side vector of (1) or each column of the right-hand side matrix of (2). The scaled noise level is given by $$100\left(\frac{\|e\|}{\|b\|}\right)$$ where $\|e\|$ is the norm of the error contaminating the right-hand side vector (or each right-hand side vector).
`eta`	A constant greater than 1 (i.e. `eta` > 1) used in the discrepancy principle that acts as a safety parameter to protect against over-minimizing the residual.
Outputs	**Description**
`x`	The approximate solution of (1) according to the discrepancy principle.
`X`	The approximate solution of (2) according to the discrepancy principle..

`iter`	This parameter shows the number of iterations that have been carried out by the function when the discrepancy principle is satisfied. To compute the number of matrix-vector products with `A` for any of the six methods, compute $$(\texttt{iter} + \texttt{ell})k$$ where k is the number of right-hand sides in the linear or block linear system.
`rrnorm`	A $1 \times \texttt{iter}$ vector containing the norms of the relative residual errors of the computed iterates of the linear system, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\texttt{iter}}$, or block linear system, $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{\texttt{iter}}$, respectively: $$\texttt{rrnorm} = \left[\|\boldsymbol{A}\boldsymbol{x}_1 - \boldsymbol{b}^\delta\|/\|\boldsymbol{b}^\delta\|, \ldots, \|\boldsymbol{A}\boldsymbol{x}_{\texttt{iter}} - \boldsymbol{b}^\delta\|/\|\boldsymbol{b}^\delta\|\right]$$ or $$\texttt{rrnorm} = \left[\|\boldsymbol{A}\boldsymbol{X}_1 - \boldsymbol{B}^\delta\|/\|\boldsymbol{B}^\delta\|, \ldots, \|\boldsymbol{A}\boldsymbol{X}_{\texttt{iter}} - \boldsymbol{B}^\delta\|/\|\boldsymbol{B}^\delta\|\right].$$ The Euclidean and spectral matrix norm are utilized above where appropriate.

Table 4: Input parameters for and outputs from `GMRES`, `ellShiftGMRES`, `BGMRES`, `ellShiftBGMRES`, `glGMRES`, & `ellShiftGlGMRES` functions.

4 Demos and Further Resources

Two demos for the six algorithms are included in the ***ellShiftedPkg.zip*** package. The demo `linearSys_demo.m` partially replicates the results and figures of the *Shaw* example in [1]. The noise level can be adjusted manually in the script (default is 0.01, corresponding to 1% standard normal noise). The script file `blockLinearSys_demo.m` provides a basic example for the block case using two right-hand sides. In this demo, $\boldsymbol{A}$ and $\boldsymbol{b}$ come from the `phillips_alt.m` function from [2]. The right-hand sides are individually contaminated with noise. No parameters are required to be entered for either script. As long as all necessary files are in the working directory, the demos may be executed by selecting *Run*.

Providing demos and the necessary code for the *Satellite* and *Board* examples from the accompanying paper is beyond the scope of this code package. The major limiting factor here involves producing a matrix function handle which can efficiently compute matrix-vector products. An existing package which contains this functionality is the IR-Tools package by Gazzola et. al [6]. The aforementioned contains demos for image deblurring as well as various other ill-posed problems. We invite those interested in the underpinnings of image deblurring to see the work by Hansen et. al in [7].

References

[1] A. Buccini, L. Onisk, and L. Reichel. "Range Restricted Iterative Methods for Linear Discrete Ill-posed Problems". In: *Electronic Transactions on Numerical Analysis* XX (20XX), pp. XXX–XXX.

[2] A. Neuman, L. Reichel, and H. Sadok. "Algorithms for range restricted iterative methods for linear discrete ill-posed problems". In: *Numer. Algorithms* 59 (2012), pp. 325–331.

[3] A. Neuman, L. Reichel, and H. Sadok. "Implementations of range restricted iterative methods for linear discrete ill-posed problems". In: *Linear Algebra Appl.* 436 (2012), pp. 3974–3990.

[4] C.B. Shaw Jr. "Improvements of the resolution of an instrument by numerical solution of an integral equation". In: *J. Math. Anal. Appl.* 37 (1972), pp. 83–112.

[5] D. L. Phillips. "A technique for the numerical solution of certain integral equations of the first kind". In: *J. ACM* 9 (1962), pp. 84–97.

[6] S. Gazzola, P. C. Hansen, and J. G. Nagy. "IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems". In: *Numerical Algorithms* 81 (2019), pp. 773–811.

[7] P. C. Hansen, J. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering.* SIAM, Philadelphia, 2006.