THE UNIVERSITY *of* TENNESSEE *at* CHATTANOOGA
COLLEGE *of* ENGINEERING *and* COMPUTER SCIENCE
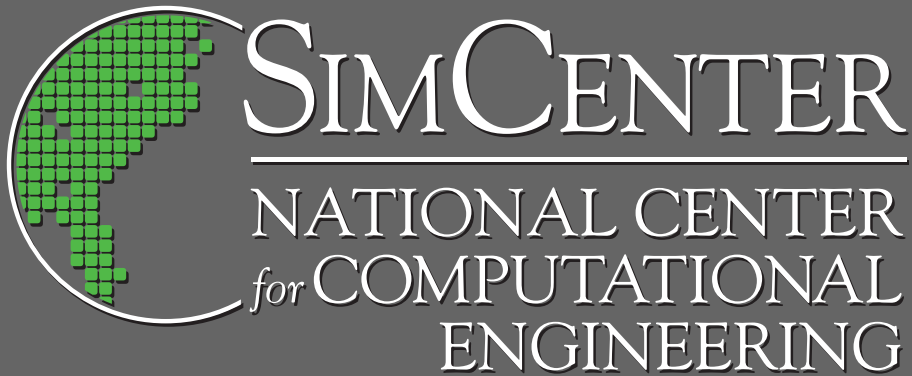
SIMCENTER
NATIONAL CENTER *for* COMPUTATIONAL ENGINEERING

A Progressive Statistical Method for Preconditioning Matrix-Free Solution of High-Order Discretization of Linear Time-Dependent Problems

A Technical Report by

A. Ghasemi
and L. K. Taylor

GRADUATE SCHOOL OF COMPUTATIONAL ENGINEERING

701 East M.L. King Boulevard • Chattanooga, TN 37403

A Progressive Statistical Method for Preconditioning Matrix-Free Solution of High-Order Discretization of Linear Time-Dependent Problems

A. Ghasemi and *L. K. Taylor*
SimCenter: National Center for Comp. Eng.,
701 East M L King Blvd, Chattanooga, TN 37403

Introduction and Motivations

Consider the general form of linear time-dependent problems

$$\frac{\partial u}{\partial t} = R = \mathcal{L}\left(t, u\right) = \sum_k \sum_i \alpha_{ki} \frac{\partial^k u}{\partial x_i^{\ k}}, \qquad (1)$$

where the residual vector R contains high-order derivatives of the dependent variable $u(t, x_1, x_2, \ldots, x_n)$ with respect to independent variables. Equation (1) is an abstract representation of a generic Partial Differential Equation (PDE) in the residual form where the linear operator $\mathcal{L}$ accounts for a linear combination of the kth partial derivative in the ith direction, i.e. $\partial^k / \partial x_i^{\ k}$. When discretized using a suitable discretization method like Finite-Differences, Finite Elements or Finite Volume [5], the resulting equations can be written in the following semi-discrete form

$$\frac{d}{dt} \mathbf{u} = \mathbf{L}\ \mathbf{u}. \qquad (2)$$

The fully discrete from is obtained after eq.(2) is discretized in time. There are many choices available for time-discretization [6]. However a common step in these algorithms, which is the most costly part of the solution, is simply an Euler implicit scheme presented below

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \mathbf{L}\mathbf{u}^{n+1}, \qquad (3)$$

which leads to the solution of the following *system of linear equations*

$$\underbrace{\left(\mathbf{I} - \Delta t \mathbf{L}\right)}_{\mathbf{A}} \mathbf{u}^{n+1} = \mathbf{u}^n, \qquad (4)$$

where $\mathbf{A} = \left(\mathbf{I} - \Delta t \mathbf{L}\right)$ is typically a huge sparse matrix for practical problems. For this reason, a direct solution of eq.(4) is impossible and thus an iterative algorithm is usually used for the solution. These algorithms range from stationary methods including Jacobi, Gauss-Seidel, SOR to more sophisticated Krylov subspace methods CG (Conjugate Gradient) and GMRES (Generalized Minimal Residual). The reader should refer to [4] and references therein for more detail. In

this work, the GMRES algorithm [1][1] is used as the base algorithm however the statistical preconditioning proposed here can be readily applied to all Krylov subspace methods because they are based on matrix-vector multiplication.

The important point is that for many practical problems the matrix $\mathbf{A}$ in eq.(4) is so huge that it does not fit in the physical memory of the current machines. This situation typically happens in Large Eddy Simulation (LES) or Direct Numerical Simulations (DNS) of a turbulent fluid flow [8]. In this situation a full implicit solution procedure is avoided to prevent the problem of storing the full Jacobian matrix. Yet still another possibility is available by using a matrix-free implementation [7]. In this approach the matrix $\mathbf{A}$ is never computed/stored but eq.(4) is solved by replacing the matrix-vector product in the GMRES algorithm with residual computation. However the only disadvantage of this approach is that preconditioning is almost impossible since entries a_{ij} of $\mathbf{A}$ are not available. This is the *motivation* for current work where a simple statistical method is used to estimate $\mathbf{A}$ by least-squaring the history of matrix-vector product obtained in the GMRES algorithm.

The PSP-GMRES Algorithm

The details of the Progressive Statistical Preconditioning for GMRES is covered in algorithms (1) and (2) below. To solve the linear system eq.(4), the original GMRES algorithm starts as usual in algorithm (1). However when a matrix-vector product is performed in lines 4 and 11 of (1), for the given vector $\mathbf{P}_x(:, i)$, the result is stored in the ith column of matrix $\mathbf{P}_y$, i.e. $\mathbf{P}_y(:, i)$. Therefore the *statistical* dataset $\mathbf{P}_y$ versus $\mathbf{P}_x$ is obtained *progressively* in a sense that the more matrix-vector product is performed the better dataset is obtained. Now, the primary goal is to find a relation between $\mathbf{P}_y$ and $\mathbf{P}_x$ such that it has minimum error compared to the exact relation $\mathbf{P}_y = \mathbf{A}\mathbf{P}_x$. This relation, which is estimated using a banded diagonal matrix $\mathbf{N}$ such that $\mathbf{P}_y \approx \mathbf{N}\mathbf{P}_x$ will be further used as a preconditioner in the GMRES algorithm (1) when eq.(4) is solved for the next time step Δt. The approximation of $\mathbf{A}$ with the banded diagonal matrix $\mathbf{N}$ is illustrated in fig.(1). As shown, all off-diagonal entries on a given rows of the original huge matrix $\mathbf{A}$ are reduced to a banded matrix $\mathbf{N}$ where the matrix-vector product would be cheap. This reduction procedure is completely described in algorithm (2) using a multi-regression estimation of matrix $\mathbf{P}_y$ versus matrix $\mathbf{P}_x$. As shown, the output of the GMRES algorithm, i.e. $\mathbf{P}_x$ and $\mathbf{P}_y$ are sent to MREP (Multi regressor Preconditioner) where for $d = 1$, a tridiagonal estimation (three regressors) is enforced. The emphasis on three regressor model is based on the fact that this approach leads to a tridiagonal matrix $\mathbf{N}$ which can be solved efficiently in $O(n)$ FLOPs using Thomas algorithm. Thus it is expected that this three-regressor model would save a lot when the preconditioned equations need to be solved in lines 10 and 42 of alg.(1).

As a summary, the output of the matrix-vector product in the GMRES alg.(1) is stored in two matrices $\mathbf{P}_x$ and $\mathbf{P}_y$ which are used as dataset for further modeling of the unavail-

---

[1]The least-squares is used in Ref.[1] to solve a system containing an upper Hessenberg matrix. However in this work it is used to estimate the preconditioner from a regression point of view.

able matrix $\mathbf{A}$. Subsequently, in the next immediate stage, the matrices $\mathbf{P}_x$ and $\mathbf{P}_y$ are used in alg. (2) where a three regressors least square procedure (four unknowns) is used (d=1) to obtain the tridiagonal entries of the approximate matrix $\mathbf{N}$. This matrix is guaranteed to approximate the exact matrix $\mathbf{A}$ with minimal residual. Finally the computed matrix $\mathbf{N}$ is used as a preconditioner in the GMRES algorithm (1) to improve the convergence. Obviously, the more convergence of GMRES is improved the closer $\mathbf{N}$ to $\mathbf{A}$ is and vice versa. Now, it is interesting to see the results of this method in the implementation.

1 **subroutine** (X, R, $\mathbf{P}_x$, $\mathbf{P}_y$) $\leftarrow$ **PSPGMRES** ($\mathbf{A}$, B, $\mathbf{N}$, ϵ, n_A, X_0, n_r);
2 $i \leftarrow 1$;
3 **for** $l \leftarrow 1$ **to** n_r **do**
4 $\quad$ $\mathbf{P}_x(:,i) \leftarrow X_0$, $\mathbf{P}_y(:,i) \leftarrow \mathbf{A}\,X_0$;
5 $\quad$ $\bar{R} \leftarrow B - \mathbf{P}_y(:,i)$, $i \leftarrow i + 1$;
6 $\quad$ $\mathbf{V}(:,1) \leftarrow \bar{R}/\|\bar{R}\|$;
7 $\quad$ $G(1) \leftarrow \|\bar{R}\|$;
8 $\quad$ **for** $k \leftarrow 1$ **to** n_A **do**
9 $\quad\quad$ $G(k+1) \leftarrow 0$;
10 $\quad\quad$ $Y_k \leftarrow \mathbf{N}^{-1}\mathbf{V}(:,k)$;
11 $\quad\quad$ $\mathbf{P}_x(:,i) \leftarrow Y_k$, $\mathbf{P}_y(:,i) \leftarrow \mathbf{A}\,Y_k$;
12 $\quad\quad$ $U_k \leftarrow \mathbf{P}_y(:,i)$, $i \leftarrow i + 1$;
13 $\quad\quad$ **for** $j \leftarrow 1$ **to** k **do**
14 $\quad\quad\quad$ $\mathbf{H}(j,k) \leftarrow \mathbf{V}(:,j)^{'}\,U_k$;
15 $\quad\quad\quad$ $U_k \leftarrow U_k - \mathbf{H}(j,k)\,\mathbf{V}(:,j)$;
16 $\quad\quad$ **end**
17 $\quad\quad$ $\mathbf{H}(k+1,k) \leftarrow \|U_k\|$;
18 $\quad\quad$ $\mathbf{V}(:,k+1) \leftarrow U_k/\mathbf{H}(k+1,k)$;
19 $\quad\quad$ **for** $j \leftarrow 1$ **to** $k-1$ **do**
20 $\quad\quad\quad$ $\delta \leftarrow \mathbf{H}(j,k)$;
21 $\quad\quad\quad$ $\mathbf{H}(j,k) \leftarrow \delta C(j) + S(j)\mathbf{H}(j+1,k)$;
22 $\quad\quad\quad$ $\mathbf{H}(j+1,k) \leftarrow -\delta S(j) + C(j)\mathbf{H}(j+1,k)$;
23 $\quad\quad$ **end**
24 $\quad\quad$ $\gamma \leftarrow \sqrt{\mathbf{H}(k,k)^2 + \mathbf{H}(k+1,k)^2}$;
25 $\quad\quad$ $C(k) \leftarrow \mathbf{H}(k,k)/\gamma$;
26 $\quad\quad$ $S(k) \leftarrow \mathbf{H}(k+1,k)/\gamma$;
27 $\quad\quad$ $\mathbf{H}(k,k) \leftarrow \gamma$;
28 $\quad\quad$ $\mathbf{H}(k+1,k) \leftarrow 0$;
29 $\quad\quad$ $\delta \leftarrow G(k)$;
30 $\quad\quad$ $G(k) \leftarrow C(k)\delta + S(k)G(k+1)$;
31 $\quad\quad$ $G(k+1) \leftarrow -S(k)\delta + C(k)G(k+1)$;
32 $\quad\quad$ $R(k) \leftarrow \|G(k+1)\|$;
33 $\quad\quad$ **if** $R(k) \le \epsilon$ **then**
34 $\quad\quad\quad$ *finish-flag gets* 1;
35 $\quad\quad\quad$ **break**;
36 $\quad\quad$ **end**
37 $\quad$ **end**
38 $\quad$ $Q \leftarrow \mathbf{H}^{-1}G$;
39 $\quad$ **for** $j \leftarrow 1$ **to** k **do**
40 $\quad\quad$ $Z_k \leftarrow Z_k + Q(j)\mathbf{V}(:,j)$;
41 $\quad$ **end**
42 $\quad$ $X \leftarrow X_0 + \mathbf{N}^{-1}Z_k$;
43 $\quad$ $X_0 \leftarrow X$;
44 $\quad$ **clean** $G, \mathbf{V}, \mathbf{H}, C, S$;
45 $\quad$ **if** *finish-flag is set* **then**
46 $\quad\quad$ **break**;
47 $\quad$ **end**
48 **end**

Algorithm 1: The Progressive Statistical Preconditioned GMRES algorithm (PSPGMRES) with possible restarting.

1 **function** $\mathbf{N} \leftarrow$ **MREP** ($\mathbf{P}_x$, $\mathbf{P}_y$, d);
2 n $\leftarrow$ number of rows of $\mathbf{P}_x$;
3 m $\leftarrow$ number of columns of $\mathbf{P}_x$;
4 **for** $i \leftarrow 1$ **to** d **do**
5 $\quad$ $(\beta_0, \beta_1) \leftarrow$ **linear fit**($\mathbf{P}_x(i,:)$, $\mathbf{P}_y(i,:)$);
6 $\quad$ $\mathbf{N}(i,i) \leftarrow \beta_1$;
7 **end**
8 **for** $i \leftarrow (d+1)$ **to** $(n-d)$ **do**
9 $\quad$ $\mathbf{P}_x^* \leftarrow \begin{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}_{1\times m} \\ \mathbf{P}_x(i-d,:) \\ \vdots \\ \mathbf{P}_x(i-1,:) \\ \mathbf{P}_x(i,:) \\ \mathbf{P}_x(i+1,:) \\ \vdots \\ \mathbf{P}_x(i+d,:) \end{bmatrix}$;
10 $\quad$ $\mathbf{P}_y^* \leftarrow \mathbf{P}_y(i,:)$;
11 $\quad$ $N^* \leftarrow \left(\mathbf{P}_x^*(\mathbf{P}_x^*)'\right)^{-1}\mathbf{P}_x^*\left(\mathbf{P}_y^{*'}\right)$;
12 $\quad$ **for** $j \leftarrow 2$ **to** $2(d+1)$ **do**
13 $\quad\quad$ $\mathbf{N}(i, i-j-d+1) \leftarrow N^*(j)$;
14 $\quad$ **end**
15 **end**
16 **for** $i \leftarrow (n-d+1)$ **to** n **do**
17 $\quad$ $(\beta_0, \beta_1) \leftarrow$ **linear fit**($\mathbf{P}_x(i,:)$, $\mathbf{P}_y(i,:)$);
18 $\quad$ $\mathbf{N}(i,i) \leftarrow \beta_1$;
19 **end**

Algorithm 2: The algorithm for Multi-Regressor Estimator of the Preconditioner (MREP).

Results

The algorithms are implemented in a computer program. In a mock test case, the matrix $\mathbf{A}_{n\times n}$ in eq.(4) is a seven diagonal random matrix which is generated on fly in a way that diagonal dominance is preserved. This vector $\mathbf{u}^n$ is just a vector from 1 to n. The size of the system of equation, i.e. n is varied to $n = \{20, 80, 150, 350, 700\}$. The structure of the original matrix $\mathbf{A}$ for the case 20x20 and the preconditioner $\mathbf{N}$ is shown in fig.(3). As shown for $d = 1$, a tridiagonal matrix is generated. Figure (2) compares the convergence of the PSP-GMRES method with the original GMRES algorithm.

The original matrix of discretization $\qquad$ The statistically estimated preconditioner
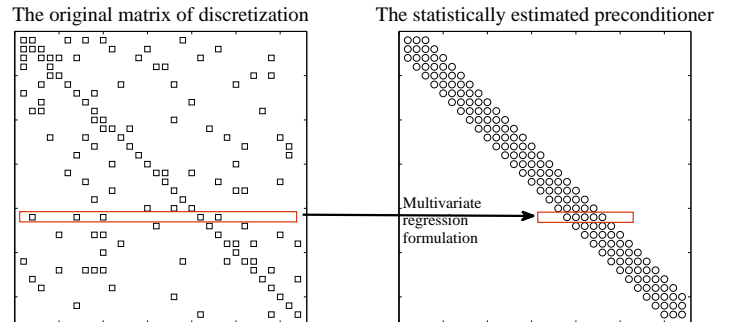
Multivariate regression formulation

Figure 1: The schematic of the statistical reduction of the sparse matrix $\mathbf{A}$ into the banded diagonal matrix $\mathbf{N}$.

As shown for case 20x20, the convergence is decreased from 40 iterations to 30 iterations. This indicates a factor

4/3 in the speed-up of original algorithm. Interestingly more impressive results would be obtained for larger systems. According to the same figure, as the size of the linear system is increased, the speed-up genrated by the PSP algorithm is also increased. As shown in fig.(2), the new statistical preconditioning approach reduces the convergence of the original GMRES from approximately 400 iterations to approximately 200 iterations when **A** is 700x700 matrix! This is a numerically validated indication of the robustness of the statistical approach.
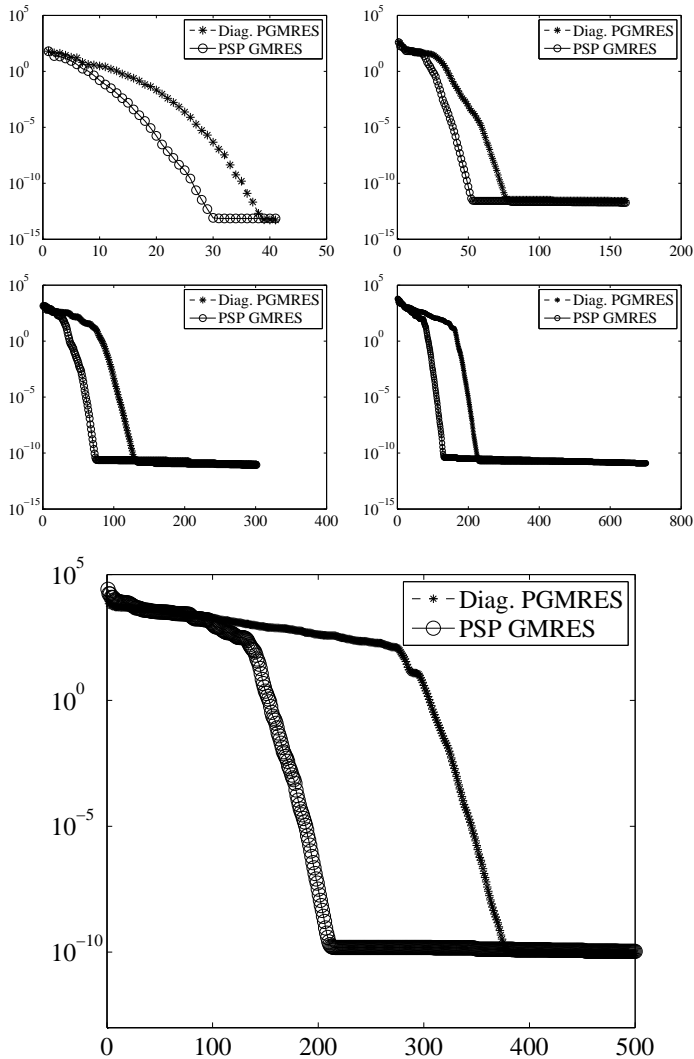
Figure 2: The residual norm versus number of GMRES iterations. From top-left to bottom-right: 20x20, 80x80, 150x150, 350x350 and 700x700.

More Regressors and Conclusions

It should be noted that the number of regressors used in alg.(2) was chosen to be three ($d = 1$) in the presented implementation in order to obtain a tridiagonal preconditioner. However other alternatives might be available for $d > 1$ for future works. For example, all off-diagonal entries might be selected as regressors in the initial model. Therefore the initial model might have n regressors. As the GMRES progress

and more matrix-vector multiplications are performed, a standard model selection procedure, like backward elimination, might be used to eliminate inappropriate regressors[3, 2]. This would result in optimum model which might result in optimum speed-up of the convergence of the GMRES algorithm. However this approach has two *expensive* parts that should be analyzed mathematically in a separate paper.

The first disadvanage is that elimination of inappropriate regressors requires to generate some models at startup. This would require *at least* the cost of $O(n^3)$ for generating the first model using all regressors which is not acceptable because the cost is even higher than solving the system *without* preconditioning! In addition, more models are needed to be created and compared to each other during this procedure which makes it even more expensive.

The second disadvangate of increasing the number of regressors is that the final preconditioner matrix would not be tridiagonal therefore the cost of solution of preconditioned system in lines 10 and 42 of alg.(1) would increase dramatically in this case!
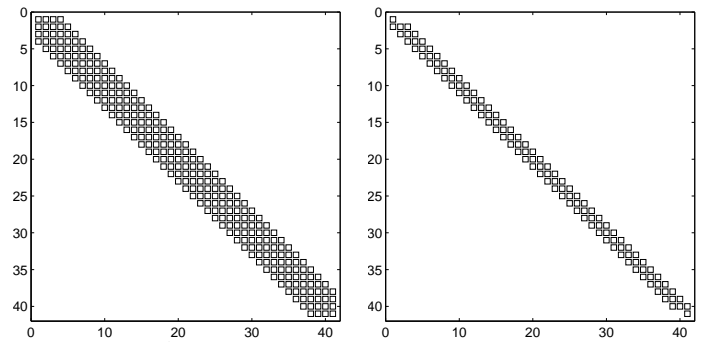

Figure 3: Left) The original pattern of the matrix. Right) The statistically reduced matrix used as preconditioner. The size of the matrix is $n = 20$.

As a final conclusion, the three regressor model presented here has already generated impressive results in improving the convergence of the original GMRES algorithm according to fig(2). However, there might be situations that adding more regressors, i.e. $d > 1$ would result in outstanding improvements in the convergence rate. It was mentioned that a backward elimination method is not practical in this case thus a forward selection is appropriate. This situation should be considered more rigorously in the future works.

References

[1] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM J. Sci. Stat. Comput.*, July 1986, **7**,3,Jul,1986, pp. 856-869.

[2] C. Gao, "Lecture Note Applied Statistical Methods", University of Tennessee at Chattanooga, Spring 2013, http://web2.utc.edu/~thc544/Math4120Spring2013/Lecture%20Notes/.

[3] D. C. Montgomery, E. A. Peck and G. G. Vining, "Introduction to Linear Regression Analysis 4th Ed.", Wiley-Interscience, 2006.

[4] R. Barrett et. al., "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods", 2nd Ed., SIAM, 1994.

[5] C. Johnson, "Numerical Solution of Partial Differential Equations by the Finite Element Method", Dover Pub,. Jan. 2009.

[6] J. C. Butcher, "Numerical Methods for Ordinary Differential Equations", 2nd Ed., Wiley, 2008.

[7] , D.A. Knoll and D.E. Keyes, "Jacobian-free NewtonKrylov methods: a survey of approaches and applications", J. Comp. Physics, **193**,pp. 357-397, 2007.

[8] M. Lesieur, O. Mtais and P. Comte, "Large-Eddy Simulations of Turbulence", Cambridge Univ. Press, 2005.