TACC Technical Report TR-13-03

FLAME Derivation of the General Minimal Residual Method

Paolo Bientinesi*, Victor Eijkhout†

This work was sponsored by NSF through awards CCF 0917096 and OCI-0850750, and by grant GSC 111 of the Deutsche Forschungsgemeinschaft (German Research Association). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

* RWTH Aachen, Germany
† Texas Advanced Computing Center, The University of Texas at Austin

1 Introduction

The Conjugate Gradient method is found by constructing a sequence of residuals R that spans the Krylov space K and that is orthogonal. The GMRES method constructs another basis of the Krylov space, but the residuals S now satisfy

- the residual is of minimum L_2 length in each iteration, or equivalent,
- the residuals are A-orthogonal.

While it would be easy to construct the A-orthogonal basis of K, and in fact several methods do so, this construction can break down. For this reason, Saad and Schultz [2] proposed using a breakdown-free process, such as Arnoldi iteration, and constructing the basis by appropriate linear combinations.

2 Theory

Let us inventory the available facts.

We can start from a CG method

$$AM^{-1}R = RH, \quad H = (I-J)D^{-1}(I+U), \quad R^tM^{-1}R = \Omega^2$$

which constructs R, Ω, and H (or equivalently D and U), such that U is strictly upper triangular and Ω diagonal. Here the operator A is given, as well as the first column of R.

The Arnoldi iteration for computation of eigenvalues computes a very similar set:

$$AM^{-1}N = NG, \quad N^tM^{-1}N = I$$

where G is an upper Hessenberg matrix; the operator A is given, as well as the first column of N. The main difference between these procedures is that the Arnoldi method generates an orthonomal basis N of the Krylov space, while CG methods construct only an orthogonal basis; it is uniquely determined by the form $H = (I-J)D^{-1}(I+U)$ of the Hessenberg matrix.

It is not hard to prove that the two bases differ only by a diagonal scaling:

$$R = N\Omega, \quad H = \Omega^{-1}G\Omega. \tag{1}$$

Any sequence of residuals S with the same starting vector is now related to V by taking linear combinations $S = RV$ where V is upper triangular with column sums $\equiv 1$. (Such linear combinations are called ‘affine’; note that they are not convex since the coefficients are allowed to be negative.)

The GMRES derivation problem is now:

> Given the residuals R of a CG method or the basis N of a Lanczos method, which differ by a scaling $R = N\Omega$, find new residuals
>
> $$S = RV$$
>
> where
>
> $$V \in \mathbf{U}^e = \{U : U \text{ upper triangular and } e^t U = e^t\}$$
>
> such that the columns of S have minimal length over all possible choices $V = \mathbf{U}^e$.

We introduce a notation for the above type of problem where each column of a matrix is the solution of a minimization problem:

Definition 1 *We define a vector sequence X to be a 'multiple minimizer' with respect to a sequence R and a predicate $P(\cdot)$*

$$X = \underset{X \in S}{\text{mmin}}\{R\colon P(R)\} \quad \text{where } P(R) \text{ is a predicate on } R$$

iff each column of X minimizes a problem:

$$\forall_i\colon x_i = \text{argmin}\{\|r_i\|\colon P(r_i)\}.$$

In the specific case that the individual problems are minimization problems, we introduce a shorthand notation:

$$X = \underset{X \in S}{\text{mmin}}(A, F) = \underset{X \in S}{\text{mmin}}\{R\colon R = AX - F\}$$

which means that

$$\forall_i\colon x_i = \text{argmin}\{\|r_i\|\colon r_i = Ax_i - f_i\}.$$

With this notation, we can state that the GMRES derivation problem is that of finding V such that

$$V = \underset{V \in \mathbf{U}^e}{\text{mmin}}\{S\colon S = RV\},$$

This is also the post-condition on the GMRES worksheet we are constructing.

In order to arrive at a construction for S, we now perform a number of manipulations on this minimization problem.

- First of all, we transform the equation $S = RV$ into an update equation, by multiplying by a square matrix $J-E$:

$$\begin{cases} S = RV \text{ where} \\ V = \text{mmin}_{\mathbf{U}^e}\{S\colon S(J-E) = RV(J-E)\} \end{cases}$$

- Next we observe that, with $V \in \mathbf{U}^e$, $e^tV(J-E) = 0^t$, so (up to the last element) $V(J-E)$ is an upper Hessenberg matrix with zero column sums. If we call the set of such Hessenberg matrices $\mathbf{H}^0$, then

$$\begin{cases} S = RV \text{ where} \\ V \text{ follows from } \tilde{H} = V(J-E), \text{ and} \\ \tilde{H} = \text{mmin}_{\mathbf{H}^0}\{S\colon S(J-E) = R\tilde{H}\} \end{cases}$$

- Now we observe that any $H \in \mathbf{H}^0$ can be written (by lemma 23 of [1]) as $H = (I-J)U$ with U upper triangular. Therefore, the $\tilde{H}$ matrix above is related to the Hessenberg matrix H from the CG algorithm by $\tilde{H} = H\tilde{U}$ for some upper triangular matrix $\tilde{U}$. The minimization problem now becomes:

$$\begin{cases} S = RV \text{ where} \\ V \text{ follows from } \tilde{H} = V(J-E), \text{ and} \\ \tilde{H} \text{ follows from } \tilde{H} = H\tilde{U}, \text{ and} \\ \tilde{U} = \text{mmin}_{\mathbf{U}}\{S\colon S(J-E) = RH\tilde{U}\} \end{cases}$$

  where $\mathbf{U}$ is the set of all upper triangular matrices.
- Next, we note that $s_1 = r_1$, so we can write $S = [r_1, \tilde{S}]$ with $\tilde{S}$ the tail of the sequence S. This also makes $\tilde{S} = SJ$ and $RE = SE$. We also write $\bar{H} = \Omega H$ and $\bar{E} = \Omega E$. With this we rewrite (silently introducing a minus sign)

$$\begin{aligned} \tilde{U} &= \text{mmin}_{\mathbf{U}}\{S\colon S(J-E) = RH\tilde{U}\} \\ &= \text{mmin}_{\mathbf{U}}\{\tilde{S}\colon \tilde{S} = RH\tilde{U} - RE\} \\ &= \text{mmin}_{\mathbf{U}}(RH, RE) \\ &= \text{mmin}_{\mathbf{U}}(N\bar{H}, N\bar{E}) \\ &= \text{mmin}_{\mathbf{U}}(\bar{H}, \bar{E}) \end{aligned}$$

  making the final problem

$$\begin{cases} S = RV \text{ where} \\ V \text{ follows from } \tilde{H} = V(J-E), \text{ and} \\ \tilde{H} \text{ follows from } \tilde{H} = H\tilde{U}, \text{ and} \\ \tilde{U} = \text{mmin}_{\mathbf{U}}(\bar{H}, \bar{E}) \end{cases}$$

Collapsing this whole story, We now have the situation that the s_n vectors are computed as

$$s_{n+1} = r_0 - Rv_n, \qquad v_n = Hu_n, \qquad u_n = \operatorname*{argmin}_u \left\| \|r_1\| e_1 - \Omega_{n+1} H_{[n+1,n]} u \right\| \quad (2)$$

Bearing in mind the fact that the Hessenberg matrices of the Arnoldi and CG methods differ by a scaling (equation (1)), we can also write

$$u_n = \operatorname*{argmin}_w \left\| \|r_1\| e_1 - G_{[n+1,n]} u \right\| \quad (3)$$

We compute this by making a QR factorization of H. Since H gets extended by a column in every iteration, we can also update the QR factorization.

3 Worksheets

The algorithm for GMRES, as described above is an interleaving of two algorithms: the creation of the basic R residuals, and the derivation of the minimized combinations S from them. Therefore we will present this as two separate worksheets.

3.1 Residual sequence

Easy things first: the equations for generating R. We have a PME

$$\begin{cases} AM^{-1}\left(R_L \,\middle|\, r_M \,\middle|\, R_R \right) = \left(R_L \,\middle|\, r_M \,\middle|\, R_R \right) \left(\begin{array}{c|c|c} H_{TL} & h_{TM} & H_{TR} \\ \hline h^t_{ML} & h_{MM} & h^t_{MR} \\ \hline \emptyset & h_{BM} & H_{BR} \end{array} \right) \\ \left(\begin{array}{c} R^t_L \\ \hline r^t_M \\ \hline R^t_R \end{array} \right) M^{-1} \left(R_L \,\middle|\, r_M \,\middle|\, R_R \right) = \left(\begin{array}{c|c|c} \Omega^2_{TL} & 0 & 0 \\ \hline 0 & \omega^2_{MM} & 0 \\ \hline 0 & 0 & \Omega^2_{BR} \end{array} \right) \\ \left(e^t \,\middle|\, 1 \,\middle|\, e^t \right) \left(\begin{array}{c|c|c} H_{TL} & h_{TM} & H_{TR} \\ \hline h^t_{ML} & h_{MM} & h^t_{MR} \\ \hline \emptyset & h_{BM} & H_{BR} \end{array} \right) = \left(0^t \,\middle|\, 0 \,\middle|\, 0^t \right) \end{cases} \quad (4)$$

where h_{ML} is nonzero only in its last component, and h_{BM} only in its first. From this we pick the following invariant:

$$\begin{cases} (AM^{-1}R_L) = \begin{pmatrix} R_L & r_M \end{pmatrix} \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix} \\ \begin{pmatrix} R^t_L \\ r^t_M \end{pmatrix} \begin{pmatrix} R_L & r_M \end{pmatrix} = \begin{pmatrix} \Omega^2_{TL} & 0 \\ 0 & \omega^2_{MM} \end{pmatrix} \\ \begin{pmatrix} e^t & 1 \end{pmatrix} \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix} = (0^t) \end{cases}$$

The before equations are

$$AM^{-1}R_0 = (R_0\, r_1) \begin{pmatrix} H_{00} \\ h^t_{01} \end{pmatrix}, \qquad e^t H_{00} + h^t_{01} = 0^t, \qquad \begin{pmatrix} R^t_0 \\ r_1 \end{pmatrix} M^{-1} \begin{pmatrix} R_0 & r_1 \end{pmatrix} = \begin{pmatrix} \Omega^2_{00} & \\ & \omega^2_1 \end{pmatrix}$$

The after equations are

$$(AM^{-1}R_0 AM^{-1}r_1) = (R_0\, r_1\, r_2) \begin{pmatrix} H_{00} & h_{01} \\ h^t_{01} & h_{11} \\ 0 & h_{21} \end{pmatrix}, \qquad (e^t\, 1\, 1) \begin{pmatrix} H_{00} & h_{01} \\ h^t_{01} & h_{11} \\ 0 & h_{21} \end{pmatrix} = (0^t\, 0)$$

$$\begin{pmatrix} R^t_0 \\ r^t_1 \\ r^t_2 \end{pmatrix} M^{-1} \begin{pmatrix} R_0 & r_1 & r_2 \end{pmatrix} = \begin{pmatrix} \Omega^2_{00} & & \\ & \omega^2_1 & \\ & & \omega^2_2 \end{pmatrix}$$

The extra equations to satisfy after the update are

$$\begin{cases} AM^{-1}r_1 = R_0 h_{01} + r_1 h_{11} + r_2 h_{21} \\ R^t_0 M^{-1} r_2 = 0, \quad r^t_1 M^{-1} r_2 = 0 \\ e^t h_{01} + h_{11} + h_{21} = 0 \end{cases}$$

The computation now proceeds as follows:

- Clearly, we need to choose

  $$r_2 h_{21} = AM^{-1}r_1 - R_0 h_{01} - r_1 h_{11}$$

  where all h_{*1} coefficients are to be determined.
- For $0 = R^t_0 M^{-1} r_2$ we note that

  $$R^t_0 M^{-1} r_2 h_{21} = R^t_0 AM^{-1} r_1 - R^t_0 M^{-1} R_0 h_{01}$$

  so choosing $h_{01} = \Omega_0^{-2} R^t_0 AM^{-1} r_1$ suffices.

- Similarly, inspection of

$$r_1^t M^{-1} r_2 = r_1^t A M^{-1} r_1 - r_1^t M^{-1} r_1 h_{11}$$

shows that $h_{11} = \omega_1^{-2} r_1^t A M^{-1} r_1$ makes $r_1^t M^{-1} r_2 = 0$ be satisfied.
- Finally, $h_{21} = -(h_{11} + e^t h_{01})$.

Formally:

0. Description Our target operation is to compute: $AM^{-1}R = RH$, where $e^t H = 0^t$, given A and r_0.

1. Precondition and postcondition The precondition is that the first column of R is known: $Re_0 = r_0$. At the end of the algorithm, R and H are fully computed to satisfy

$$AM^{-1}R = RH, \quad e^t H = 0^t, \quad R^t M^{-1} R = \Omega, \quad \Omega \text{ diagonal.}$$

3. Partitioning We use a three-way partition: $R \rightarrow \left(\begin{array}{c|c|c} R_L & r_M & R_R \end{array} \right), H \rightarrow \left(\begin{array}{c|c|c} H_{TL} & h_{TM} & H_{TR} \\ \hline h_{ML}^t & \eta_{MM} & h_{MR}^t \\ \hline 0 & h_{BM} & H_{BR} \end{array} \right)$, where initially R_L is $N \times 1$.

This gives a Partitioned Matrix Expression (PME):

$$AM^{-1} \left(\begin{array}{c|c|c} R_L & r_M & R_R \end{array} \right) = \left(\begin{array}{c|c|c} R_L & r_M & R_R \end{array} \right) \left(\begin{array}{c|c|c} H_{TL} & h_{TM} & H_{TR} \\ \hline h_{ML}^t & \eta_{MM} & h_{MR} \\ \hline \emptyset & h_{BM} & H_{BR} \end{array} \right)$$

where H is a Hessenberg matrix, so H_{TL}, H_{BR} are themselves of Hessenberg form, and

$$h_{ML}^t = (0, \ldots, 0, \star), \quad h_{BM}^t = (\star, 0, \ldots, 0).$$

2. Loop invariant We choose the following set:

$$AM^{-1}R_L = \left(\begin{array}{c|c} R_L & r_M \end{array} \right) \left(\frac{H_{TL}}{h_{ML}^t} \right), \quad \left(\begin{array}{c|c} e_L^t & 1 \end{array} \right) \left(\frac{H_{TL}}{h_{ML}^t} \right) = 0^t$$

$$\left(\begin{array}{c|c} R_L & r_M \end{array} \right)^t M^{-1} \left(\begin{array}{c|c} R_L & r_M \end{array} \right) = \left(\begin{array}{c|c} \Omega_{TL} & 0 \\ \hline 0 & \omega_{MM} \end{array} \right).$$

4. Loop guard The loop guard is, as usual, $R_R <> ()$.

5. Repartition The usual:
$$\begin{array}{c}\left(\begin{array}{c|c|c} R_L & r_M & R_R \end{array}\right) \rightarrow \\ \left(\begin{array}{c|c|c|c} R_0 & r_1 & r_2 & R_3 \end{array}\right)\end{array}, \quad \left(\begin{array}{c|c|c} H_{TL} & h_{TM} & H_{TR} \\ \hline h_{ML}^t & \eta_{MM} & h_{MR}^t \\ \hline 0 & h_{BM} & H_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c|c} H_{00} & h_{01} & h_{02} & H_{03} \\ \hline h_{10}^t & \eta_{11} & \eta_{12} & h_{13}^t \\ \hline 0 & \eta_{21} & \eta_{22} & h_{23}^t \\ \hline 0 & 0 & h_{32} & H_{33} \end{array}\right).$$

6. Before update The algorithm state is

$$AM^{-1}R_0 = \left(\begin{array}{c|c} R_0 & r_1 \end{array}\right)\left(\begin{array}{c} H_{00} \\ \hline h_{10}^t \end{array}\right), \quad \left(\begin{array}{c|c} e_L^t & 1 \end{array}\right)\left(\begin{array}{c} H_{00} \\ \hline h_{10}^t \end{array}\right) = 0^t$$
$$\left(\begin{array}{c|c} R_0 & r_1 \end{array}\right)^t M^{-1}\left(\begin{array}{c|c} R_0 & r_1 \end{array}\right) = \left(\begin{array}{c|c} \Omega_{00} & 0 \\ \hline 0 & \omega_{11} \end{array}\right).$$

7. After update After the update, the algorithm state is

$$AM^{-1}\left(\begin{array}{c|c} R_0 & r_1 \end{array}\right) = \left(\begin{array}{c|c|c} R_0 & r_1 & r_2 \end{array}\right)\left(\begin{array}{c|c} H_{00} & h_{01} \\ \hline h_{10}^t & \eta_{11} \\ \hline 0 & h_{21} \end{array}\right), \quad \left(\begin{array}{c|c|c} e_L^t & 1 & 1 \end{array}\right)\left(\begin{array}{c|c} H_{00} & h_{01} \\ \hline h_{10}^t & \eta_{11} \\ \hline 0 & h_{21} \end{array}\right) = 0^t$$
$$\left(\begin{array}{c|c|c} R_0 & r_1 & r_2 \end{array}\right)^t M^{-1}\left(\begin{array}{c|c|c} R_0 & r_1 & r_2 \end{array}\right) = \left(\begin{array}{c|c|c} \Omega_0 & 0 & 0 \\ \hline 0 & \omega_1 & 0 \\ \hline 0 & 0 & \omega_2 \end{array}\right).$$

8. Update This leaves us with the following set to satisfy in the update:

$$\begin{cases} AM^{-1}r_1 = R_0h_{01} + r_1\eta_{11} + r_2\eta_{21} \\ e^t h_{01} + \eta_{11} + \eta_{21} = 0 \\ r_2^tM^{-1}r_2 = \omega_2, \, r_2^tM^{-1}R_0 = 0, \, r_2^tM^{-1}r_1 = 0 \end{cases}$$

which leads to the obvious update:
$$\begin{cases} t \leftarrow AM^{-1}r_1 \\ h_{01} = \Omega_0^{-1}R_0^t t, \quad t \leftarrow t - R_0h_{01} \\ h_{11} = \omega_1^{-1}r_1^t t, \quad t \leftarrow t - r_1h_{11} \\ h_{21} = -h_{11} - e^t h_{01}, \quad t \leftarrow t/h_{21}, \quad r_2 = t \end{cases}$$

The full worksheet is in Figure 1.

3.2 Updated QR factorization

The crucial part in constructing GMRES from a CG or Arnoldi method, is to solve the minimization problems (2) and (3) respectively. This is done by computing a QR factorization of H or G, and update it in every iteration. This is in a way simpler than a

Step	**Annotated Algorithm:** $AM^{-1}R = RH$, where $e^tH = 0^t$
1a	$\{Re_0 = r_0\}$
4	$R \rightarrow \left(R_L \mid r_M \mid R_R \right), H \rightarrow \begin{pmatrix} H_{TL} & h_{TM} & H_{TR} \\ h^t_{ML} & \eta_{MM} & h^t_{MR} \\ 0 & h_{BM} & H_{BR} \end{pmatrix}$
2	$\left\{ \begin{array}{l} AM^{-1}R_L = \left(R_L \mid r_M \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix}, \quad \left(e^t_L \mid 1 \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix} = 0^t \\ \left(R_L \mid r_M \right)^t M^{-1} \left(R_L \mid r_M \right) = \begin{pmatrix} \Omega_{TL} & 0 \\ 0 & \omega_{MM} \end{pmatrix}. \end{array} \right\}$
3	**while** $R_R <> ()$ **do**
2,3	$\left\{ \left\{ \left(\begin{array}{l} AM^{-1}R_L = \left(R_L \mid r_M \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix}, \quad \left(e^t_L \mid 1 \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix} = 0^t \\ \left(R_L \mid r_M \right)^t M^{-1} \left(R_L \mid r_M \right) = \begin{pmatrix} \Omega_{TL} & 0 \\ 0 & \omega_{MM} \end{pmatrix}. \end{array} \right) \wedge \left(R_R <> () \right) \right\} \right\}$
5a	$\begin{array}{l} \left(R_L \mid r_M \mid R_R \right) \rightarrow \\ \left(R_0 \mid r_1 \mid r_2 \mid R_3 \right) \end{array}, \begin{pmatrix} H_{TL} & h_{TM} & H_{TR} \\ h^t_{ML} & \eta_{MM} & h^t_{MR} \\ 0 & h_{BM} & H_{BR} \end{pmatrix} \rightarrow \begin{pmatrix} H_{00} & h_{01} & h_{02} & H_{03} \\ h^t_{10} & \eta_{11} & \eta_{12} & h^t_{13} \\ 0 & \eta_{21} & \eta_{22} & h^t_{23} \\ 0 & 0 & h_{32} & H_{33} \end{pmatrix}$
6	$\left\{ \begin{array}{l} AM^{-1}R_0 = \left(R_0 \mid r_1 \right) \begin{pmatrix} H_{00} \\ h^t_{10} \end{pmatrix}, \quad \left(e^t_L \mid 1 \right) \begin{pmatrix} H_{00} \\ h^t_{10} \end{pmatrix} = 0^t \\ \left(R_0 \mid r_1 \right)^t M^{-1} \left(R_0 \mid r_1 \right) = \begin{pmatrix} \Omega_{00} & 0 \\ 0 & \omega_{11} \end{pmatrix}. \end{array} \right\}$
8	$t \leftarrow AM^{-1}r_1$ $h_{01} = \Omega_0^{-1} R_0^t t, \quad t \leftarrow t - R_0 h_{01}$ $h_{11} = \omega_1^{-1} r_1^t t, \quad t \leftarrow t - r_1 h_{11}$ $h_{21} = -h_{11} - e^t h_{01}, \quad t \leftarrow t / h_{21}, \quad r_2 = t$
7	$\left\{ \begin{array}{l} AM^{-1} \left(R_0 \mid r_1 \right) = \left(R_0 \mid r_1 \mid r_2 \right) \begin{pmatrix} H_{00} & h_{01} \\ h^t_{10} & \eta_{11} \\ 0 & h_{21} \end{pmatrix}, \quad \left(e^t_L \mid 1 \mid 1 \right) \begin{pmatrix} H_{00} & h_{01} \\ h^t_{10} & \eta_{11} \\ 0 & h_{21} \end{pmatrix} = 0^t \\ \left(R_0 \mid r_1 \mid r_2 \right)^t M^{-1} \left(R_0 \mid r_1 \mid r_2 \right) = \begin{pmatrix} \Omega_0 & 0 & 0 \\ 0 & \omega_1 & 0 \\ 0 & 0 & \omega_2 \end{pmatrix}. \end{array} \right\}$
5b	...
2	$\left\{ \begin{array}{l} AM^{-1}R_L = \left(R_L \mid r_M \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix}, \quad \left(e^t_L \mid 1 \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix} = 0^t \\ \left(R_L \mid r_M \right)^t M^{-1} \left(R_L \mid r_M \right) = \begin{pmatrix} \Omega_{TL} & 0 \\ 0 & \omega_{MM} \end{pmatrix}. \end{array} \right\}$
	endwhile
2,3	$\left\{ \left(\begin{array}{l} AM^{-1}R_L = \left(R_L \mid r_M \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix}, \quad \left(e^t_L \mid 1 \right) \begin{pmatrix} H_{TL} \\ h^t_{ML} \end{pmatrix} = 0^t \\ \left(R_L \mid r_M \right)^t M^{-1} \left(R_L \mid r_M \right) = \begin{pmatrix} \Omega_{TL} & 0 \\ 0 & \omega_{MM} \end{pmatrix}. \end{array} \right) \wedge \neg \left(R_R <> () \right) \right\}$
1b	$\{P_{\text{post}}\}$

Figure 1: Worksheet for the single recurrence derivation of CG

regular QR, since we factor a Hessenberg matrix, which implies that already computed columns do not need to be updated.

The PME is:

$$\begin{pmatrix} H_{TL} & h_{TM} & H_{TR} \\ h_{ML}^t & \eta_{MM} & h_{MR}^t \\ 0 & h_{BM} & H_{BR} \end{pmatrix} = \begin{pmatrix} Q_{TL} & q_{TM} & Q_{TR} \\ q_{ML}^t & \gamma_{MM} & q_{MR}^t \\ 0 & q_{BM} & Q_{BR} \end{pmatrix} \begin{pmatrix} R_{TL} & r_{TM} & R_{TR} \\ 0 & \rho_{MM} & r_{MR}^t \\ 0 & 0 & R_{BR} \end{pmatrix}$$

with the condition that

$$Q^t Q = I.$$

Doing the usual $3\times 3 \rightarrow 4\times 4$ shtick, we get before equations

$$\begin{pmatrix} H_{00} \\ h_{10} \end{pmatrix} = \begin{pmatrix} Q_{00} \\ q_{10}^t \end{pmatrix} \begin{pmatrix} R_{00} \end{pmatrix}$$

and after equations

$$\begin{pmatrix} H_{00} & h_{01} \\ h_{10}^t & \eta_{11} \\ 0 & \eta_{21} \end{pmatrix} = \begin{pmatrix} Q_{00} & q_{01} \\ q_{10}^t & \gamma_{11} \\ 0 & \gamma_{21} \end{pmatrix} \begin{pmatrix} R_{00} & r_{01} \\ 0 & \rho_{11} \end{pmatrix}$$

The update needs to satisfy

$$H_{*1} = Q_{*0} r_{01} + Q_{*1}\rho_{11}$$

Multiply this by Q_{*0} to get

$$Q_{*0}^t H_{*1} = r_{01},$$

followed by

$$u := H_{*1} - Q_{*0} r_{01}, \quad \rho_{11} = \sqrt{u^t u}, \quad Q_{*1} = u/\rho_{11}.$$

3.3 Minimized residuals

We now have a remarkable number of equations from which to derive GMRES. First of all, we can derive CG from a single recurrence $AR = RH$ where H has zero column sums and R is orthogonal, or split recurrences $APD = R(I-J)$ and $P(I-U) = R$, again with orthogonality of R. Then, we can use Lanczos $AN = NG$ with $N^t N = I$, which differs from CG by a scaling $R = N\Omega$, giving $H = \Omega^{-1} G \Omega$.

$$\begin{cases} AR = RH, \quad e^t H = 0^t, \quad R^t R = \Omega^2 \\ H = (I-J)D^{-1}(I-U) \\ APD = R(I-J), \quad P(I-U) = R \\ H = \Omega^{-1} N \Omega \end{cases}$$

The GMRES coefficient can then be computed from H or G:

$$\begin{cases} \tilde{U} = \text{mmin}(\Omega H, \Omega E) \\ \Omega\tilde{U} = \text{mmin}(G, \Omega E) \end{cases} .$$

Furthermore, we have the freedom of using Householder or Givens transformations for solving the least squares problem. This gives us a plethora of methods, that are all equivalent in exact arithmetic, but may have pronounced differences in actual computer arithmetic. Also, this system of equations contains quantities that do not need to be computed: if H is computed explicitly, D and U need not be (the other way around may not be true). If G is computed, H, D, U are not needed at all. Et cetera. In an automated derivation system this system should give rise to a true graph of possible computations; only the end points matter, not what is computed in between.

References

[1] Victor Eijkhout. Background theorems for Krylov solver theory. Technical Report TR-11-04, Texas Advanced Compuitng Center, 2011.

[2] Yousef Saad and Martin H. Schultz. GMRes: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.