A NOTE ON GMRES ALGORITHM

PANAYOT S. VASSILEVSKI

Abstract. This notes describe a computational version of the GMRES (generalized minimum residual) algorithm.

1. Iterative methods, iterates, residuals, and minimum residual least-squares approach

We are interested in solving a given system of linear equations $A\mathbf{x} = \mathbf{b}$ with A being a general $n \times n$ non–symmetric matrix.

An iterative method generates a sequence of approximations (iterates), which successively approach the exact solution. If $\mathbf{x}$ is an approximation, then $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ is called residual. It is clear that if $\mathbf{r} = 0$, then the respective $\mathbf{x}$ solves the problem $A\mathbf{x} = \mathbf{b}$. One way to measure the accuracy of an approximation $\mathbf{x}$ to the exact solution $\mathbf{x}_* = A^{-1}\mathbf{b}$, is to measure the norm of the residual, $\|\mathbf{r}\|$. If for a certain accuracy ϵ, e.g., $\epsilon = 10^{-6}$, we have $\|\mathbf{r}\| \leq \epsilon\|\mathbf{r}_0\|$, where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ (and $\mathbf{x}_0$ is the initial iterate), then we declare the current iterate $\mathbf{x}$ to be a good approximation to $\mathbf{x}_*$. Note that we cannot measure the exact error $\mathbf{x} - \mathbf{x}_*$ since we do not know the exact solution $\mathbf{x}_*$ (this is what we want to find).

Any iterative method, starting with an arbitrary initial iterate $\mathbf{x}_0$ (typical choices are 0 or random vector), generates a sequence of successive iterates

$$\mathbf{x}_0,\ \mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_k,\ \ldots$$

and computes the corresponding residuals

$$\mathbf{r}_0,\ \mathbf{r}_1,\ \mathbf{r}_2,\ \ldots,\ \mathbf{r}_k,\ \ldots$$

so that $\|\mathbf{r}_k\| \mapsto 0$. In *minimum residual* algorithms, we have monotonicity, i.e.,

$$\|\mathbf{r}_0\| \geq \|\mathbf{r}_1\| \geq \|\mathbf{r}_2\| \geq \ldots\ \|\mathbf{r}_k\| \geq \cdots \mapsto 0.$$

This means that at every iteration step, we *improve* the approximation.

One way to achieve this is done as follows. Let $\mathbf{x}$ be the current iterate and $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ the respective residual. We choose a *search direction* $\mathbf{p}$ (to be described later), and then we construct the new iterate as $\mathbf{x}_{new} = \mathbf{x} + \alpha\mathbf{p}$, where the parameter α is chosen so that we get a smaller residual $\mathbf{r}_{new} = \mathbf{b} - A(\mathbf{x} + \alpha\mathbf{p}) = \mathbf{r} - \alpha A\mathbf{p}$, than $\mathbf{r}$. To achieve this, we solve the minimization problem for $\alpha \in \mathbb{R}$,

$$\|\mathbf{r}_{new}\| = \|\mathbf{r} - \alpha A\mathbf{p}\| \mapsto \min.$$

Date: March 1, 2018–beginning; Today is February 18, 2019.
1991 *Mathematics Subject Classification.* 65F10, 65N20, 65N30.
Key words and phrases. nonsymmetric matrices, GMRES algorithm.

In fact, at iteration step $k \geq 1$, we keep the search directions that we have chosen so far, $\mathbf{p}_0, \ \ldots, \ \mathbf{p}_{k-1}$ and add one more $\mathbf{p}_k$. For example, at step $k = 1$, we have $\mathbf{p}_0$ and we add $\mathbf{p}_1$. Then our new iterate will be

$$\mathbf{x}_{new} = \mathbf{x} + \alpha_1 \mathbf{p}_0 + \alpha_2 \mathbf{p}_1.$$

Then the new residual is

$$\mathbf{r}_{new} = \mathbf{b} - A(\mathbf{x} + \alpha_1 \mathbf{p}_0 + \alpha_2 \mathbf{p}_1) = \mathbf{r} - \alpha_1 A\mathbf{p}_0 - \alpha_2 A\mathbf{p}_1.$$

We can rewrite $\alpha_1 A\mathbf{p}_0 + \alpha_2 A\mathbf{p}_1 = [A\mathbf{p}_0, A\mathbf{p}_1] \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$. Now let

$$P = [\mathbf{p}_1, \ \mathbf{p}_2], B = AP = [A\mathbf{p}_1, \ A\mathbf{p}_2], \text{ and } \mathbf{y} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

Then

$$\mathbf{r}_{new} = \mathbf{r} - B\mathbf{y}.$$

We find α_1 and α_2, i.e., the vector $\mathbf{y}$, by solving the *least-squares* problem

$$\|\mathbf{r} - B\mathbf{y}\| \mapsto \min.$$

This minimization ensures the monotonicity $\|\mathbf{r}\| \geq \|\mathbf{r}_{new}\|$ (note that $\mathbf{r}_{new} = \mathbf{r}$ if $\alpha = 0$). We note that at step k, B is a $n \times (k+1)$ matrix. Initially, when $k = 0$, B has just one column.

Once, we solve the least squares problem, we have $\mathbf{x}_{new} = \mathbf{x} + P\mathbf{y}$, and $\mathbf{r}_{new} = \mathbf{r} - B\mathbf{y}$.

The choice of the next search direction $\mathbf{p}_k$ is based on the most recent residual $\mathbf{r}$ and the previous directions $\mathbf{p}_0, \ \ldots, \ \mathbf{p}_{k-1}$, using Gram-Schmidt. Assuming that we have already constructed $\mathbf{p}_0, \ \ldots, \ \mathbf{p}_{k-1}$ to be orthogonal set, we first compute

$$\widetilde{\mathbf{p}}_k = \mathbf{r} - c_0\mathbf{p}_0 - c_1\mathbf{p}_1 - \cdots - c_{k-1}\mathbf{p}_{k-1},$$

for appropriate constants c_j, to be orthogonal to all previous search directions, and then normalize it, to get $\mathbf{p}_k$, i.e., $\mathbf{p}_k = \frac{\widetilde{\mathbf{p}}_k}{\|\widetilde{\mathbf{p}}_k\|}$. The constants are $c_j = \mathbf{p}_j^T \mathbf{r}$.

To summarize, the above procedure involves:

- *matrix-vector* products with A, e.g., in the computations of $A\mathbf{p}_j, \ j = 1, \ \ldots, \ k$ which gives the columns of B. For large matrices A can be stored in CSR format. B is generally a dense but tiny matrix ($k << n$).
- solving by *least-squares* the problem $B\mathbf{y} = \mathbf{r}$. For this, we use the QR factorization of B (using Gram-Schmidt), i.e., $B = QR$, and we find $\mathbf{y}$ from $R\mathbf{y} = \mathcal{Q}^T\mathbf{r}$ using the fact that R is upper triangular $k \times k$ (i.e., small) matrix.
- Computing $\mathbf{x}_{new} = \mathbf{x} + P\mathbf{y}$ and $\mathbf{r}_{new} = \mathbf{r} - B\mathbf{y}$, both involving matrix-vector operations with tiny matrices P and B (much fewer columns than rows).
- Gram-Schmidt is also used to generate $\mathbf{p}_k$ from $\mathbf{r}$ and the previous search directions $\mathbf{p}_0, \ \ldots, \ \mathbf{p}_{k-1}$.

A complete algorithm is presented in the next section.

2. Problem formulation

We assume that we have a given vector inner product, $(\mathbf{u}, \mathbf{v})$, for example $(\mathbf{u}, \mathbf{v}) = \mathbf{v}^T\mathbf{u}$ and the corresponding norm $\|\mathbf{v}\| = \sqrt{(\mathbf{v},\ \mathbf{v})}$.

For the GMRES algorithm, we need the following input:

- $\max_{iter}$: maximal number of iterations, e.g., $\max_{iter} = 1000$.
- ϵ: tolerance, e.g., $\epsilon = 10^{-6}$.
- $m_{\max}$: maximal number of search directions, e.g., $m = 50$.
- $\mathbf{x}_0$: initial iterate, e.g., $\mathbf{x}_0 = 0$, or a random vector.

The GMRES method generates at every iteration, a set of orthogonal vectors $\mathbf{p}_1,\ \ldots,\ \mathbf{p}_m,\ m \leq m_{\max}$ and computes the next iterate $\mathbf{x}_{next}$ based on the current residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$, where $\mathbf{x}$ is the current iterate, and the search vectors $\mathbf{p}_1,\ \ldots,\ \mathbf{p}_m$ as follows,

$$\mathbf{x}_{next} = \mathbf{x} + \sum_{k=1}^{m} \alpha_k \mathbf{p}_k.$$

The coefficient vector $\boldsymbol{\alpha} = (\alpha_k)_{k=1}^m$ is computed by solving the following least-squares problem

$$\|\mathbf{r} - A\left(\sum_{k=1}^{m} \alpha_k \mathbf{p}_k\right)\| \mapsto \min.$$

Introducing, the matrices $P = [\mathbf{p}_1,\ \mathbf{p}_2,\ \ldots,\ \mathbf{p}_m]$ and $B = AP$, which is a $n \times m$ (rectangular) matrix, the above problem is equivalent to solving the least-squares problem

$$\|B\boldsymbol{\alpha} - \mathbf{r}\| \mapsto \min, \tag{2.1}$$

for which we can use the QR algorithm. It factorizes B as $B = QR$, where Q is $n \times m$ orthogonal matrix and R is an $m \times m$ upper triangular matrix. Then, the least-squares problem takes the form

$$R\boldsymbol{\alpha} = Q^T\mathbf{r}.$$

I.e., we compute $\boldsymbol{\alpha} = (\alpha_k)_{k=1}^m$ by solving the above small $m \times m$ upper triangular system.

Once the coefficients $\boldsymbol{\alpha} = (\alpha_k)_{k=1}^m$ are found, the next iterate is computed as

$$\mathbf{x}_{next} = \mathbf{x} + P\boldsymbol{\alpha} = \mathbf{x} + \sum_{k=1}^{m} \alpha_k \mathbf{p}_k.$$

The next residual is

$$\mathbf{r}_{next} = \mathbf{b} - A\mathbf{x}_{next} = \mathbf{r} - B\boldsymbol{\alpha}.$$

Before continuing with the iterations, we check if have reached the maximal number of allowed iterations steps, $\max_{iter}$, and if not, we check if we have reached the tolerance limit, i.e., we check if

$$\|\mathbf{r}_{next}\| \leq \epsilon \|\mathbf{r}_{initial}\|.$$

If not, we proceed with the iteration process. This is done as follows.

If $m < m_{\max}$, we add one more vector to the previous set of search vectors by Gram-Schmidt, i.e, computing

$$\widetilde{\mathbf{p}}_{m+1} = \mathbf{r}_{next} - \sum_{k=1}^{m} (\mathbf{r},\ \mathbf{p}_k)\mathbf{p}_k.$$

and then letting

$$\mathbf{p}_{m+1} = \frac{1}{\|\widetilde{\mathbf{p}}_{m+1}\|}\ \widetilde{\mathbf{p}}_{m+1}.$$

Then we set

- $m := m + 1$
- $\mathbf{r} := \mathbf{r}_{next}$
- $\mathbf{x} := \mathbf{x}_{next}$.

and repeat. If $m = m_{\max}$, restart, i.e., we set $m = 1$, and let $\mathbf{p}_1 = \frac{1}{\|\mathbf{r}\|}\ \mathbf{r}$, where $\mathbf{r}$ is the most recent residual, and then repeat the above procedure.

We summarize next the entire GMRES algorithm.

Algorithm 2.1 (GMRES algorithm)**.**

- *Input: A-$n \times n$ matrix, $\mathbf{b} \in \mathbb{R}^n$ - right hand side vector.*
- *Tolerance: ϵ.*
- *Maximal number of GMRES iterations:* $\max_{iter}$.
- *Maximal number of search directions:* $m_{\max}$.
- *Memory for vectors: solution* $\mathbf{x}$*, residual* $\mathbf{r}$*, search vectors* $\mathbf{p}_1,\ \ldots,\ \mathbf{p}_{m_{\max}}$.
- *Memory for Q and R, Q is $n \times m_{\max}$ array, and R is $m_{\max} \times m_{\max}$ array.*
- *Memory for the coefficient vectors $\boldsymbol{\alpha} = (\alpha_k) \in \mathbb{R}^{m_{\max}}$ and the right-hand side vector $\boldsymbol{\beta} \in \mathbb{R}^{m_{\max}}$ of the small problem with the upper triangular matrix R.*

The algorithm proceeds as follows:

- *Initiate: set initial iterate* $\mathbf{x}$ *(zero or random vector).*
- *Compute initial residual:*

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}.$$

- *Compute initial residual norm*

$$\delta_0 = \|\mathbf{r}\|.$$

- *Set $i = 0$.*
- *(Re)start: Set $i := i + 1$. Set $m = 1$ and choose (initial) search direction*

$$\mathbf{p}_1 = \frac{1}{\|\mathbf{r}\|}\ \mathbf{r}.$$

- *Set $P = \mathbf{p}_1$ and $B = A\mathbf{p}_1$ (one-column matrices).*
- *Loop:*
  (1) *Solve the least-squares problem*

$$\|\mathbf{r} - B\boldsymbol{\alpha}\| \mapsto \min,$$

  by first computing $B = QR$, the right-hand-side vector $\boldsymbol{\beta} = Q^T\mathbf{r}$, and then solving for $\boldsymbol{\alpha}$, $R\boldsymbol{\alpha} = \boldsymbol{\beta}$.

(2) *Compute next iterate*

$$\mathbf{x} := \mathbf{x} + P\boldsymbol{\alpha}.$$

(3) *Compute next residual*

$$\mathbf{r} := \mathbf{r} - B\boldsymbol{\alpha}.$$

(4) *Compute residual norm*

$$\delta = \|\mathbf{r}\|.$$

(5) *Check for convergence, i.e., if*

$$\delta < \epsilon\delta_0.$$

If yes, exit.

- *If* $i + 1 = \max_{iter}$ *exit.*
- *If* $m < m_{\max}$*, compute next search direction, by first computing*

$$\mathbf{p}_{m+1} = \mathbf{r} - \sum_{k=1}^{m} (\mathbf{r},\ \mathbf{p}_k)\mathbf{p}_k.$$

and its norm $\|\mathbf{p}_{m+1}\|$*. The actual search direction is normalized, i.e.,*

$$\mathbf{p}_{m+1} := \frac{\mathbf{p}_{m+1}}{\|\mathbf{p}_{m+1}\|}.$$

Augment the matrices P *and* B *with one more column:* $P := [P,\ \mathbf{p}_{m+1}]$ *and* $B := [B,\ A\mathbf{p}_{m+1}]$*. Then set* $m := m + 1$ *and repeat Loop.*

- *If* $m = m_{\max}$*, go to (Re)start.*

Department of Mathematics and Statistics, Portland State University, Portland, OR 97207

Email address: panayot@pdx.edu