THE GMRES METHOD

David F. Gleich

November 14, 2019

1 EFFICIENT GMRES

Recall the prototype-GMRES method.

```
Given A, b where we can only multiply by A.
for i=1 to maxiter
  Update the Arnoldi factorization Q_k, H_{k+1}.
  Solve for z_k by minimizing ||H_{k+1} z_k - ||b|| e_1||,
    i.e. z_k = argmin ||H_{k+1} z_k - ||b|| e_1||
  Let x_k = Q_k z_k.
  Check ||A x_k - b||
```

To implement this, we need to solve a lseast squares problem at each step. This takes $O(k^2)$ work because it's a Hessenberg matrix. Then we need to construct the solution and check the residual. These take $O(nk)$ and another matrix-vector product. We can do all of these steps more efficiently!

Here is the outline for the essential idea to optimize GMRES.

we only need to check the residual at each step, and do not need to compute $\mathbf{x}_k$.

So the method we'll look at optimizing is:

```
Given A, b where we can only multiply by A.
for i=1 to maxiter
  Update the Arnoldi factorization Q_k, H_{k+1}.
  Compute ||r_k|| where
      r_k = A x_k - b
      x_k = Q_k z_k
      z_k = argmin ||H_{k+1} z_k - ||b|| e_1||
    and stop once ||r_k|| is sufficiently small, i.e.
    Update ||r_k|| → ||r_{k+1}|| and stop if it's small enough.
Explicitly compute z_k = argmin ||H_{k+1} z_k - ||b|| e_1||
  and return x_k = Q_k z_k only at the end of the iteration
```

1.1 THE OPTIMIZATION IDEA

Let's study the quantity we want to compute, let $\|\mathbf{b}\| = \beta_0$, then

$$\|\mathbf{r}_k\| = \|\mathbf{b} - \boldsymbol{A}\mathbf{x}_k\| = \|\mathbf{b} - \boldsymbol{A}\boldsymbol{Q}_k\mathbf{y}_k\| = \|\boldsymbol{H}_k\mathbf{y}_k - \beta_0\mathbf{e}_1\|.$$

After a four steps, this is:

$$\Bigg\| \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}}_{\boldsymbol{H}_k} \mathbf{y}_4 - \beta_0 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Bigg\|.$$

We solve least squares problems via QR, so suppose that

$$\boldsymbol{H}_k = \boldsymbol{U}_k\boldsymbol{R}_k$$

is the QR factorization after k-steps. Then

$$\|\mathbf{r}_k\| = \|\boldsymbol{U}_k\boldsymbol{R}_k\mathbf{y}_k - \beta_0\mathbf{e}_1\| = \|\boldsymbol{R}_k\mathbf{y}_k - \beta_0\boldsymbol{U}_k^T\mathbf{e}_1\|.$$

Showing this after a few steps gives us the idea more clearly:

$$\|\mathbf{r}_k\| = \Bigg\| \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{R}_k} \mathbf{y}_4 - \beta_0 \underbrace{\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \end{bmatrix}}_{\boldsymbol{U}_k^T\mathbf{e}_1} \Bigg\|. = \beta_0\gamma_5.$$

(Remember we solve for $\mathbf{y}_k$ such that this term is zero in the first four components. So we just need to figure out what γ_5 is to get $\|\mathbf{r}_k\|$.

1.2 TAKING IT DEEPER

We need to note a two things here to continue our optimization:

1. We only need Givens rotations to get $\boldsymbol{H}_k \to \boldsymbol{R}_k$.
2. We only need *one* rotation to update $\boldsymbol{R}_k \to \boldsymbol{R}_{k+1}$. (Woah!)

Let's review step 1 and see how that will help us with step 2.

Finish and improve this figure, the point is we use $J_1, \ldots, J_4$ to do the Givens rotations. These give us $\boldsymbol{U}_4^T = \boldsymbol{J}_4\boldsymbol{J}_3 \ldots \boldsymbol{J}_1$.

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{J_1} \begin{bmatrix} * & * & * & * \\ \circ & * & * & * \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \quad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{J_2} \begin{bmatrix} \times & \times & \times & \times \\ 0 & * & * & * \\ 0 & \circ & * & * \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{J_3} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & * & * \\ 0 & 0 & \circ & * \\ 0 & 0 & 0 & \times \end{bmatrix} \quad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{J_4} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & \circ \end{bmatrix}$$

Now, suppose we have $\boldsymbol{U}_4$ and $\boldsymbol{R}_4$, how do we get $\boldsymbol{U}_5, \boldsymbol{R}_5$?

$$\boldsymbol{H}_5 = \begin{bmatrix} \boldsymbol{H}_4 & \mathbf{h} \\ 0 & h_{6,5} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} & \begin{bmatrix} \times \\ \times \\ \times \\ \times \\ \times \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} & h_{6,5} \end{bmatrix}$$

If we rotate by $\boldsymbol{U}_4^T$, we get:

$$\begin{bmatrix} \boldsymbol{U}_4^T & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{H}_5 = \begin{bmatrix} \boldsymbol{U}_4^T & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{H}_5 = \begin{bmatrix} \boldsymbol{R}_4 & \boldsymbol{U}_4^T \mathbf{z}_4 \\ 0 & h_{6,5} \end{bmatrix}$$

So at this point, we just have the one Givens rotation: $\boldsymbol{J}_5$ that we need to do to fixup the element $h_{6,5}$ and so $\boldsymbol{U}_5^T = \boldsymbol{J}_5 \boldsymbol{J}_4 \cdots \boldsymbol{J}_1$, which is just one update.

1.3 SEEKING GAMMA.

Note that the elements of gamma are just the first column of $\boldsymbol{U}_k^T$. Let $\mathbf{g}_k = \boldsymbol{U}_k^T \mathbf{e}_1 = \begin{bmatrix} \gamma_1 & \gamma_2 & \ldots & \gamma_k \end{bmatrix}^T$. Then by our previous relationship:

$$\mathbf{g}_{k+1} = \boldsymbol{U}_{k+1}^T \mathbf{e}_1 = \boldsymbol{J}_{k+1} \boldsymbol{U}_k^T \mathbf{e}_1 = \boldsymbol{J}_{k+1} \mathbf{g}_k.$$

But this is weird, because $\boldsymbol{J}$ is a $k+1 \times k+1$ matrix and $\mathbf{g}_k$ is a length k vector. So what we really mean is

$$\boldsymbol{J}_{k+1} \begin{bmatrix} \mathbf{g}_k \\ 0 \end{bmatrix}$$

where we grew the vector by one element in order to make it work. Note that we don't need to actually update $\mathbf{g}_k$ even though it should change.

1.4 THE WHOLE ALGORITHM

```
g = β₀e₁
for k=1 to ...
  Update Q_k, H_k
  Let η_{k+1} = H_{k+1,k}.
  Let z_k = H_{1:k,k}.
  Apply J_1...J_{k-1} to z_k, and update H
  Create J_k to elminate η_{k+1}.
  Determine g_k from J_k g_{k-1} growing by zeros as needed.
  If g_k(end) is small enough, then stop iterating.
At this point, H has the factor R, and (if we do keep g accurate), then
g is the right hand side, so we can just solve R_k y_k = g_k and then
output x = Q_k y.
```

2 GMRES VS. FOM

This section is incomplete.

See notes.

The major point is that FOM is like CG in that it solves a linear system based on the truncated Arnoldi factorization.

The large scale study by Peter Brown (http://dx.doi.org/10.1137/0912003) concluded: there is little difference, but liked the minimum residual property of GMRES.