Numerical techniques for large dimensional dynamical systems
Recent Trends in Nonlinear Science 2015
Basics of Krylov methods

Contents

1 Introduction. Krylov spaces

To solve a linear system $Ax = b$ there are techniques other than the different versions of Gaussian elimination. These are iterative techniques, where from an initial approximation $x^{(0)}$ to the solution x, a sequence $(x^{(n)})_{n=1}^{\infty}$ of approximations is generated, hopefully each better than the previous one. The computation of the approximations $x^{(n)}$ is typically aborted when the residual $r^{(n)} = b - Ax^{(n)}$ becomes sufficiently small. Iterative methods are a better choice that direct methods (those based on gaussian elimination) when the following two circumstances concur.

1. The matrix dimension n is so large that gaussian elimination becomes close to eternal, due to tis cost being $O(n^3/3)$ flops (general case) or $O(n^p)$ flops where $1 < p < 3$ (which is typically the case of sparse matrices or band matrices with growing bandwidth).

2. The cost of a matrix-vector product Ax is $O(n^p)$ flops with $1 \le p < 2$, as it is the case of Toeplitz o circulant matrices, band and sparse matrices, etc.

Nowadays, the growth in computer power and memory has turned many linear problems solvable by direct methods. However, there are still a lot of instances where iterative methods are a better (or, even, the only) choice. This is typically the case of discretization of partial differential equations (PDEs) set in three-dimensional domains.

Although there are quite a variety of iterative methods for linear systems, in these lecture notes we will be concerned only with those known as *Krylov methods*, and in particular with the generalized minimum residual method (GMRES). Mention will be made also of the most successful and well-known Krylov method, the conjugate gradient method.

Given a matrix A of dimension $m \times m$ and a vector $b \in \mathbb{R}^m$ the k-order *Krylov space* generated by A and b is

$$\mathcal{K}_k(A, b) = \lim\{b, Ab, \ldots, A^{k-1}b\}.$$

For simplicity, we will drop the explicit dependence of A and b and will write $\mathcal{K}_k$ instead of $\mathcal{K}_k(A, b)$. Since there cannot be more than m linearly independent vectors in $\mathbb{R}^m$, we have

$$\mathcal{K}_m = \mathcal{K}_{m+1} = \ldots = \mathbb{R}^m.$$

Besides, Cayley-Hamilton (every matrix is a root of its characteristic polynomial) tells us how to express $A^m b$ as a linear combination of $b, Ab, \ldots, A^{m-1}b$.

If for some $k < m$ it happens that
$$A^k b \in \mathcal{K}_k, \tag{1}$$
this implies that for certain scalars $\alpha_0, \ldots, \alpha_k \in \mathbb{R}$, we have
$$\alpha_0 b + \alpha_1 Ab + \ldots + \alpha_k A^k b = 0.$$

It is always possible to choose k y $\alpha_0, \ldots, \alpha_k$ such that $\alpha_0 \neq 0$ (why?), so that
$$b = -\frac{1}{\alpha_0}\big(\alpha_1 Ab + \cdots + \alpha_k A^k b\big).$$

Since A is a common factor of all the terms on the right-hand side above, we can write
$$b = A(\underbrace{\beta_0 b + \cdots + \beta_{k-1} A^{k-1}b}_{\in \mathcal{K}_k}).$$

In other words, if (1) holds, we can express $b = Ax$ with $x \in \mathcal{K}_k$.

This is the basic idea underlying all Krylov iterative methods to solve
$$Ax = b. \tag{2}$$

In practice, hardly ever (1) holds for
$$k \ll m.$$

However, in may practical instances the following is true (although we cannot explain why yet)
$$A^k b + \varepsilon r \in \mathcal{K}_k, \qquad \text{con,} \qquad \|r\| = 1, \quad 0 < \varepsilon \ll 1, \qquad k \ll m, \tag{3}$$
which allow us to express
$$b = Ax^{(k)} + \varepsilon r', \qquad x^{(k)} \in \mathcal{K}_k,$$
so that, hopefully,
$$x^{(k)} \approx x.$$

If $k \ll m$, y $m \gg 1$, computing $b, A^b, \ldots A^{k-1}b$ is much less computationally costly than solving $Ax = b$ by Gaussian elimination (or any of its variants), as well as finding the scalars to express $x^{(k)}$ as a linear combination of $b, A^b, \ldots A^{k-1}b$ (they are found by solving an appropriate $k \times k$ linear system), so that if (3) holds, Krylov methods are advantageous when compared to direct methods.

There are two basic ways to chose the approximation $x^{(k)} \in \mathcal{K}_k$ to the solucion x if
$$b - Az \neq 0, \qquad \text{for all } z \in \mathcal{K}_k,$$
which we now comment on.

1. The Galerkin aproach where it is demanded that
$$b - Ax^{(k)} \perp \mathcal{K}_k.$$

   The resulting method is known as the full orthogonalization method (FOM), and, for symmetric positive definite matrix, the conjugate gradient method, due to Hestenes and Stiefel in 1952 (see e.g., [5]).

2. The least squares approach, where the approximation $x^{(k)}$ satisfies

$$\left\| b - Ax^{(k)} \right\| = \min_{z \in \mathcal{K}_k} \left\| b - Az \right\|.$$

The corresponding methods are known as minimum residual (MINRES) method, due to Paige Michael Saunders in 1975 in the case of symmetric matrices and, for general matrices, the generalized minimum residual method (GMRES) due to Saad and Schultz in 1986 [6].

In practice, however, we have that although $b, Ab, \ldots A^{k-1}b$ are linearly independent, they are almost parallell (why? Do you happen to remember the power method to compute the dominant eigenvalue of a matrix? What was the underlying idea of the power method?). For this reason, in practice, on needs to compute a different basis of $\mathcal{K}_k(A, b)$. Typically, this is the one given by the Arnoldi process that we now explain.

2   Arnoldi Decomposition

Consider the following process: given a matrix A of dimension $m \times m$, and a vector $v_1 \in \mathbb{R}^m$ with $\|v_1\| = 1$, construct the sequence of vectors $(v_k)_{k=1}^n$ as follows.

Para $k = 1, \ldots n - 1$

1) $u_k = Av_k$,
2) $w_k = u_k - ((v_1^T u_k)v_1 + \cdots + (v_k^T u_k)v_k)$,
3) $v_{k+1} = \dfrac{w_k}{\|w_k\|}$.

The vectors v_j así are known as *Arnoldi vectors* of A generated from v_1. Notice that for this process to be possible, we must have $Av_k \neq 0$, for $j = 1, \ldots, n$. We now comment on several facts.

1. Notice that if we knew $u_1, \ldots, u_{n-1}$ beforehand, the previous process is Gram-Schmidt orthogonalization applied to vectors $.v_1, u_1, \ldots, u_{n-1}$.

2. As a consequence, we have that vectors w_k are orthogonal vectors, and so is the case of vectors v_k.

3. In fact, denoting by $\mathcal{V}_k = \lin(v_1, \ldots, v_k)$ the space spanned by the first k Arnoldi vectors, steps 1)-3) can be expressed as

$$v_{k+1} = \frac{(I - P_{\mathcal{V}_k})Av_k}{\|(I - P_{\mathcal{V}_k})Av_k\|}. \tag{4}$$

4. We will now show that

$$\lin(v_1, \ldots, v_k) = \lin(v_1, Av_1, \ldots, A^{k-1}v_1), \tag{5}$$

that is, $\{v_1, \ldots, v_k\}$ *are an orthonormal base of the Krylov space* $K_k(A, v_1)$. This is so since by denoting $\beta_k = \|(I - P_{\mathcal{V}_k})Av_k\|$ from (4) it follows that

$$Av_k = \beta_k v_{k+1} + P_{\mathcal{V}_k}Av_k \in \lin(v_1, \ldots, v_{k+1}),$$

and this holds for $k = 1, 2, \ldots, k'$, where k' is the first integer for which $(I - P_{\mathcal{V}_{k'}})Av_{k'} = 0$. Thus,

$$Av_1 \in \operatorname{lin}(v_1, v_2),$$
$$A^2 v_1 = AAv_1 \in \operatorname{lin}(Av_1, Av_2) \subset \operatorname{lin}(v_1, v_2, v_3),$$
$$\vdots$$
$$A^{k-1}v_1 = AA^{k-2}v_1 \in \operatorname{lin}(Av_1, \ldots, Av_{k-1}) \subset \operatorname{lin}(v_1, \ldots, v_k),$$

from where (5) easily follows.

To make most of the Arnoldi process, it is better to write it as follows.

Para $k = 1, \ldots n - 1$
  1) $u_k = Av_k,$
  2.1) $h_{1k} = v_1^T u_k, \ldots, h_{kk} = v_k^T u_k,$
  2.2) $w_k = u_k - (h_{1k}v_1 + \cdots + h_{kk}v_k),$
  3.1) $h_{k+1,k} = \|w_k\|,$
  3.2) $v_{k+1} = \dfrac{w_k}{h_{k+1,k}}.$

Notice that, since $u_k = Av_k$, we have

$$\begin{aligned} w_k &= Av_k - (h_{1k}v_1 + \cdots + h_{kk}v_k) \\ w_k &= h_{k+1,k}v_{k+1}, \end{aligned}$$

so that expressing Av_k in terms of the v_js we have

$$Av_k = h_{1k}v_1 + \cdots + h_{kk}v_k + h_{k+1,k}v_{k+1}. \tag{6}$$

Denoting by
$$V_k = [v_1, \ldots, v_k],$$

the matrix whos columns are the first k Arnoldi vectors, we have

$$Av_k = h_{1k}v_1 + \cdots + h_{kk}v_k + h_{k+1,k}v_{k+1} = V_k \begin{bmatrix} h_{1k} \\ \vdots \\ h_{kk} \end{bmatrix} + h_{k+1,k}v_{k+1}.$$

But taking into account that
$$AV_k = [Av_1, \ldots, Av_k],$$

we can write
$$AV_k = V_k H_k + h_{k+1,k}[0, \ldots, 0, v_{k+1}], \tag{7}$$

where

$$H_k = \begin{bmatrix} h_{11} & \cdots & h_{1,k-1} & h_{1k} \\ h_{21} & \cdots & h_{2,k-1} & h_{2k} \\ & \ddots & \vdots & \vdots \\ & & h_{k,k-1} & h_{k,k} \end{bmatrix}.$$

4

relation (7) being known as the k-order *Arnoldi decomposition* of A

Denoting by $e_k = [0, \ldots, 0, 1]^T$ thel k-th coordinate vector in $\mathbb{R}^k$, the $m \times k$ matrix whose first $k-1$ columns are null and the k-th one is v_{k+1} can be written as

$$[0, \ldots, 0, v_{k+1}] = v_{k+1}e_k^T.$$

Thus, the Arnoldi decomposition can be written as

$$AV_k = V_kH_k + h_{k+1,k}v_{k+1}e_k^T, \tag{8}$$

Since all the columns of V_k have been obtained by the Gram-Schmid process, they are orthogonal and have unit norm, so that it follows

$$V_k^TV_k = I,$$

(notice that in general this do not imply that the inverse of V_k is V_k^T, which is only true when V_k is a square matrix), con so that multiplying by V_k^T in eq:Arnoldi:decomp2, (recall also that the columns of V_k are orthogonal to v_{k+1}, so that $V_k^Tv_{k+1} = 0$) we have

$$V_k^TAV_k = H_k. \tag{9}$$

Furthermore, by introducing the $(k+1) \times k$ matrix

$$\tilde{H}_k = \begin{bmatrix} h_{11} & \ldots & h_{1,k-1} & h_{1k} \\ h_{21} & \ldots & h_{2,k-1} & h_{2k} \\ & \ddots & \vdots & \vdots \\ & & h_{k,k-1} & h_{k,k} \\ & & & h_{k+1,k} \end{bmatrix}.$$

the Arnoldi decomposition (8)) can be written as

$$AV_k = V_{k+1}\tilde{H}_k, \tag{10}$$

Remark 1 Computation of Eigenvalues. The Arnoldi decomposition is used in a technique known as *Implicitly restarted Arnoldi methods* [2] to compute a few selected eigenvalues of a large matrix. The idea is to select form the eigenvalues of matrix H_k a few, say $k' < k$ such that are wanted for a particular reason (largest real part, smallest real part, etc), and readapt and reduce the Arnoldi decomposition so that the shorter matrix $H_{k'}$ has only those k' wanted eigenvalues. Once this is done, the Arnolid decomposition is enlarged to order k and the process is repeated until convergence (or a maximum number of iterations is reched). There is a very succesful software implementin this method, known as ARPACK [3], of which MATLAB and OCTAVE commands `eigs` are a user-friendly interface.

3 The GMRES method

All Krylov method to solve the linear system $Ax = b$ take

$$v_1 = \frac{b}{\|b\|}.$$

as starting vector in the Arnoldi process. Notice then that the system $Ax = b$ can be written as

$$Ax = \|b\| \, v_1.$$

As mentioned above the approximation $x^{(k)} \in \mathcal{V}_k$ is found as the solution of a least squares problem,

$$\left\| b - Ax^{(k)} \right\| = \min_{z \in \mathcal{V}_k} \|b - Az\| .$$

To compute he approximation $x^{(k)}$ in practice, the above least squares problem is expressed in terms of the Arnoldi vectors as we now explain.

Let us first recall that the Arnoldi vectors $v_1, \ldots, v_k$ or columns of matrix V_k form an orthonormal basis of $\mathcal{V}_k$. Thus,

$$z \in \mathcal{V}_k \quad \Rightarrow \quad z = V_k y, \qquad \text{for some } y \in \mathbb{R}^k.$$

The vector y is the vector of coordinates of z in the Arnoldi basis. Consequently

$$
\begin{aligned}
z \in \mathcal{V}_k \quad \Rightarrow \quad b - Az &= \|b\| \, v_1 - AV_k y \\
&= \|b\| \, v_1 - V_{k+1} \tilde{H}_k y \qquad \text{(due to (10))} \\
&= V_{k+1}(\|b\| \, e_1 - \tilde{H}_k y),
\end{aligned}
$$

where

$$e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{k+1}.$$

Thus, in terms of the coordinates in the Arnoldi basis, the quantity $\|b - Az\|$ to be minimized in the least squares problem can be expressed as follows:

$$
\begin{aligned}
z \in \mathcal{V}_k \quad \Rightarrow \quad \|b - Az\|^2 &= \left\| V_{k+1}(\|b\| \, e_1 - \tilde{H}_k y) \right\|^2 \\
&= (\|b\| \, e_1 - \tilde{H}_k y)^T \underbrace{V_{k+1}^T V_{k+1}}_{=I} (\|b\| \, e_1 - \tilde{H}_k y) \\
&= (\|b\| \, e_1 - \tilde{H}_k y)^T (\|b\| \, e_1 - \tilde{H}_k y) \\
&= \left\| \|b\| \, e_1 - \tilde{H}_k y \right\|^2,
\end{aligned}
$$

Thus, the approximation $x^{(k)}$ is computed as follows:

1. Find the solution $y^{(k)}$ of the least squares problem

$$\tilde{H}_k y = e_1. \tag{11}$$

2. Then set $x^{(k)} = \|b\| \, V_k y^{(k)}$.

In practice, the value k for which the approximation $x^{(k)}$ is compued is that for which

$$\frac{\left\| b - Ax^{(k)} \right\|}{\|b\|} \leq \epsilon.$$

where ϵ is a prescribed tolerance.

The least squares problem (11) is solved by means of a QR decomposition of the matrix $\tilde{H}_k$, in a process that we describe next. But first, it will be useful to write matrix $\tilde{H}_k$ by bloks as follows

$$\tilde{H}_k = \begin{bmatrix} H_k \\ h_{k+1,k}e_k^T \end{bmatrix}, \qquad \text{where} \quad e_k = [0,\ldots,0,1]^T \in \mathbb{R}^k,$$

Let $\tilde{Q}_k$ an orthogonal $(k+1) \times (k+1)$, so that

$$\tilde{Q}_k^T \tilde{Q}_k = I, \tag{12}$$

and such that

$$\tilde{H}_k = \tilde{Q}_k \tilde{R}_k, \qquad \text{con} \quad \tilde{R}_k = \begin{bmatrix} R_k \\ 0^T \end{bmatrix}, \quad R_k \text{ is a } k \times k \text{ upper triangular matrix.}$$

A few lines below we will show how to compute such matrix $\tilde{Q}_k$. If we write this matrix by blocks as

$$\tilde{Q}_k = [Q_k, q_{k+1}], \qquad Q_k \text{ matriz } (k+1) \times k \quad \text{y} \quad q_{k+1} \in \mathbb{R}^{k+1},$$

the it is well known[1]that

$$\text{the solution of l. s. problem} \quad \tilde{H}_k y^{(k)} = e_1, \qquad \text{is the solution of} \quad R_k y^{(k)} = Q_k^T e_1, \tag{13}$$

and the residual $e_1 - \tilde{H}_k y^{(k)}$ satisfies that

$$\left\| b - Ax^{(k)} \right\| = \|b\| \left\| e_1 - \tilde{H}_k y^{(k)} \right\| = \|b\| \left| q_{k+1}^T e_1 \right|.$$

The matrix

$$\tilde{R}_k = \tilde{Q}_k \tilde{H}_k,$$

is computed as

$$\tilde{R}_k = G_k \ldots G_1 \tilde{H}_k,$$

where G_j, $j = 1,\ldots,k$ is th Givens rotations

$$G_j = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \begin{matrix} c_j & -s_j \\ s_j & c_j \end{matrix} & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix},$$

---

[1]The l. s. solution is the solution Gauss normal equations $\tilde{H}_k^T \tilde{H}_k y^{(k)} = \tilde{H}^k e_1$, but notice that due to (12) we have that $\tilde{H}_k^T \tilde{H}_k = \tilde{R}_k^T \tilde{Q}_k^T \tilde{Q}_k \tilde{R}_k = \tilde{R}_k^T \tilde{R}_k$, and, furthermore $\tilde{R}_k^T \tilde{R}_k = [R_k^T, 0] \begin{bmatrix} R_k \\ 0^T \end{bmatrix} = R_k R_k^T$. On the other hand $H_k^T e_1 = [R_k, 0] \begin{bmatrix} Q_k^T \\ q_{k+1}^T \end{bmatrix} e_1 = R_k^T Q_k e_1$, so that Gauss normal equations become $R_k R_k^T y^{(k)} = R_k^T Q_k e_1$ from where statement (13) follows.

7

such that if we denote

$$H_k^{(j)} = G_{j-1} \ldots G_1 \tilde{H}_k,$$

then

$$\begin{bmatrix} c_j & -s_j \\ s_j & c_j \end{bmatrix} \begin{bmatrix} h_{j,j}^{(j)} \\ h_{j+1,j}^{(j)} \end{bmatrix} = \begin{bmatrix} \sqrt{(h_{j,j}^{(j)})^2 + (h_{j+1,j}^{(j)})^2} \\ 0 \end{bmatrix}.$$

Observe that the givens rotations G_j are orthogonal matrices, so that $G_j^{-1} = G_j^T$, $j = 1, \ldots, k$, and, hence,

$$\tilde{Q}_k = G_1^T \ldots G_k.$$

In practice, the rotations are computed and applied as the Arnoldi process progress, as we now explain. Observe that $\tilde{H}_{k+1}$ can be written as

$$\tilde{H}_k = \begin{bmatrix} \tilde{H}_{k-1} & h_k \\ 0^T & h_{k+1,k} \end{bmatrix}.$$

so that applying the previous Givens rotations we have

$$\begin{bmatrix} \tilde{Q}_{k-1}^T & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{H}_{k-1} & h_k \\ 0^T & h_{k+1,k} \end{bmatrix} = \begin{bmatrix} \tilde{R}_{k-1} & r_k \\ 0^T & r'_{k,k} \\ 0^T & h_{k+1,k} \end{bmatrix}.$$

Taking then

$$c_k = \frac{r'_{k,k}}{\sqrt{(r'_{k,k})^2 + h_{k+1,k}^2}}, \qquad s_k = -\frac{h_{k+1,k}}{\sqrt{(r'_{k,k})^2 + h_{k+1,k}^2}}, \qquad G_k = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0^T & c_k & -s_k \\ 0^T & s_k & c_k \end{bmatrix},$$

we have

$$G_k \begin{bmatrix} \tilde{Q}_{k-1}^T & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{H}_{k-1} & h_k \\ 0^T & h_{k+1,k} \end{bmatrix} = \begin{bmatrix} \tilde{R}_{k-1} & r_k \\ 0^T & r_{k,k} \\ 0^T & 0 \end{bmatrix} = \tilde{R}_k.$$

where

$$r_{kk} = \sqrt{(r'_{k,k})^2 + h_{k+1,k}^2}.$$

Thus, a simple algorithm implementing the GMRES method could be as follows

Given A, b and a tolerance TOL > 0, do as follows.

1) Set $v_1 \leftarrow b/\|b\|$, $k = 1$, $r_1 \leftarrow 2\text{TOL}$, and $d^{(0)} = e_1$

2) For k=1,2,... :

    2.1) $h_{1k} = v_1^T u_k, \ldots, h_{kk} = v_k^T u_k,$

    2.2) $w_k = u_k - (h_{1k}v_1 + \cdots + h_{kk}v_k),$

    2.3) $h_{k+1,k} = \|w_k\|,$

    2.4) $v_{k+1} = \dfrac{w_k}{h_{k+1,k}}.$

    2.5)   Compute

$$
\begin{bmatrix} r_{1,k} \\ \vdots \\ r_{k-1,k} \\ r'_{k,k} \end{bmatrix} = G_{k-1} \ldots G_1 \begin{bmatrix} h_{1,k} \\ \vdots \\ h_{k,k} \end{bmatrix},
$$

    2.6)   Set

$$
r_{k,k} = \sqrt{(r'_{k,k})^2 + h_{k+1,k}^2}, \qquad c_k = \frac{r'_{k,k}}{r_{k,k}}, \qquad s_k = -\frac{h_{k+1,k}}{r_{k,k}} \qquad \text{2.7)} \quad \text{Set}
$$

$$
\begin{bmatrix} d_1^{(k)} \\ \vdots \\ d_{k-1}^{(k)} \\ \tilde{d}_k^{(k)} \\ 0 \end{bmatrix} = \begin{bmatrix} d_1^{(k-1)} \\ \vdots \\ d_k^{(k-1)} \\ 0 \end{bmatrix},
$$

    2.8)   Set

$$
\begin{bmatrix} d_k^{(k)} \\ d_{k+1}^{(k)} \end{bmatrix} = \begin{bmatrix} c_k & -s_k \\ s_k & c_k \end{bmatrix} \begin{bmatrix} \tilde{d}_k^{(k)} \\ 0 \end{bmatrix}.
$$

    2.9)   if $|d_{k+1}^{(k)}| \leq \text{TOL}$, then go to 3)

3) Solve the system

$$
\begin{bmatrix} r_{1,1} & \cdots & r_{1,k} \\ & \ddots & \\ & & r_{k,k} \end{bmatrix} \begin{bmatrix} y_1^{(k)} \\ \vdots \\ y_k^{(k)} \end{bmatrix} = \begin{bmatrix} d_1^{(k)} \\ \vdots \\ d_k^{(k)} \end{bmatrix}
$$

and set $x^{(k)} = V_k y^{(k)}$.

With respect to the convergence of the method, as commented in the introduction, if for some $k > 0$ we have $h_{k+1,k} = 0$ then $\mathcal{V}_{\parallel}$ is an invariant subspace of A (notice that $A\mathcal{V}_j \subset \mathcal{V}_{j+1}$, $j = 1, \ldots, k-1$, and $h_{k+1,k} = 0$ implies that $A\mathcal{V}_k \subset \mathcal{V}_k$) and, being A nonsingular, this allows us to express v_1 (or for that purpose b) as $b = Ax^{(k)}$. In the worst possible case, this happens for $k = m$. But for such large value of k the method is far from useful in practice.

A much cited error bound is (see e.g. [8])

$$
\frac{\|b - Ax^{(k)}\|}{\|b\|} \leq \kappa(V) \inf_{p \in P_k} \max_{\lambda \in \sigma(A)} |p(\lambda)| \tag{14}
$$

9

where $\kappa(V)$ is the condition number of the the matrix V whose columns are the eigenvectors (or generalized eigenvectors) of A $\sigma(A)$, as usual, stands for the spectrum of A and P_k is the set of polynomials p of degree at most k with $p(0) = 1$.

When A is symmetric so that $\kappa(V) = 1$, this bound suggests us that the method will converge fast if eigenvalues are clustered around a small number of values away from the origin. When A is not symmetric, unless we have information about $\kappa(V)$ (which rarely happens in practice) the above bound is very little informative, since, as discussed in [1]. In fact, given a polynomial of degree m,

$$p(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_{m-1} x^{m-1} + x^m,$$

for the system

$$
\begin{bmatrix}
0 & 0 & \ldots & 0 & -\alpha_0 \\
1 & 0 & \ldots & 0 & -\alpha_1 \\
0 & 1 & & 0 & -\alpha_2 \\
\vdots & & \ddots & \vdots & \vdots \\
0 & 0 & & 1 & -\alpha_{m-1}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_m
\end{bmatrix}
=
\begin{bmatrix}
1 \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
$$

we have

$$1 = \left\| b - Ax^{(1)} \right\| = \left\| b - Ax^{(2)} \right\| = \ldots = \left\| b - Ax^{(m-1)} \right\|,$$

and

$$\left\| b - Ax^{(m)} \right\| = 0.$$

This is irrespective of the eigenvalues of the coefficient matrix, which, as it is well-known, are the roots of the polynomial p. Thus, for any distribution of eigenvalues of A we can always find an example of a matrix with those eigenvalues and yet the GMRES method provides useless approximations except when $k = m$. Most of examples in practice, though, are not as disheartening as this one.

It is difficult to find works in the literature that try to analyze the convergence of the GMRES method without making assumptions on the condition number of V. A remarkable exception to the general rule is the work by Igor Moret [4], where the author studies linear systems of the form

$$(I + K)u = f,$$

for K a compact operator in a Hilbert space. Bounds are obtained in terms of the degree of compactness of K, as measured by the products of its singular values.

In any case, there are many practical instances where the system $Ax = b$ has to be preconditioned. This is to replace the system by

$$\mathcal{P}^{-1}Ax = \mathcal{P}^{-1}f$$

where the operation $x \mapsto \mathcal{P}^{-1}x$ is little costly from the computational point of view and the matrix $\mathcal{P}^{-1}A$ is "better-behaved" than A, in the sense that

$$\mathcal{P}^{-1}A = \lambda I + E$$

E having small norms and conditions number, or its eigenvectors are well conditioned, etc.

Preconditioning techniques is a very active field of research nowadays. The reader is referred to [7] for an study where theory and practice are well interlaced. See also Chapter 10 in [5].

Remark 1 If before solving the linear system $Ax = b$ an approximation $x^{(0)}$ to x is known, then, the GMRES method computes an approximation $x^{(k)} = x^{(0)} + e^{(k)}$, where $e^{(k)}$ is an approximation to the solution e of $Ae = r$, r being the residual $b - Ax^{(0)}$ and such that $e^{(k)} \in \mathcal{K}_k(A, r)$.

Remark 2 In order to avoid large values of k some authors recommend to restart the method after k_r iterations, that is to compute the approximation $x^{(k_r)}$ and use it as an initial guess for for a new application of the GMRES method. The resulting method is known as *restarted* GMRES.

4 Exercises

Exercise 1 Vectors who are not null and orthogonal one another are linerly independent (Do you remember how to prove it?). In $\mathbb{R}^m$ how many non null vectors which are orthogonal one another can possibly exist then? Show that if A has dimension $m \times m$ and $h_{k+1,k} \neq 0$, for $k = 1, \ldots, m-1$, then $h_{k+1,k} = 0$.

Exercise 2 If in the Arnoldi decomposition of A we have $|h_{k+1,k}| \leq \epsilon \ll 1$, and we approximate the solution x of $Ax = b$ by $x^{(k)} = V_k y^{(k)}$, where $y^{(k)}$ is the solución of $H_k y^{(k)} = e_1$, find an upper bound of the error $\|x - x^{(k)}\|$ in terms of ϵ.

Exercise 3 Taking into account that H_k is un upper Hessemberg matrix (that is, null below its first subdiagonal), show that it is possible to obtain the QR factorization of H_k in $O(k^2)$ flops. How many flops exactly? What is the cost of solving $H_k y^{(k)} = e_1$?

Exercise 4 Write a MATLAB or OCTAVE function which, given a square matrix A, and an number k, returns the matrices V y H of the k-order Arnoldi decomposition of A. Try your function with a 4×4, and $k = 4$, and check `A*V-V*H` is 0 or a matrix with entries close to the round-off unit. Is this true if we try with $k = 2$ or $k = 3$? Why?

Exercise 5 Write a MALTAB or OCTAVE function that given A, b, a tolerance TOL and an integer $k_{\max}$, solves the linear system $Ax = b$ by the GMRES method, and returning the approximation $x^{(k)} \in \mathcal{K}_k(A, b)$ such that either $\|b - Ax^{(k)}\| / \|b\| < \text{TOL}$ or $k = \max$. Try your function with a 4×4 system and $k_{\max} = 4$.

References

[1] A. Greenbaum, V. Ptak, amd Z. Strakoš, Any nonincreasing convergence curve is possible for GMRES, *SIAM J. Matrix Anal. Appl.*, 17 (1996), 465–469.

[2] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration* SIAM J. Matrix Anal. Appl.,17 (1996), 789–821.

[3] R. B. Lehoucq, D. C. Sorensen and C. Yang, *ARPACK Users's guide. Solution of Large-Ecale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.

[4] I. Moret, A note on the superlinear convergence of GMRES, *SIAM J. Numer. Anal.* 34 (1997), 513–516.

[5] Y. Saad, *Iterative Methods for sparse Linear Systems (Second Edition)* SIAM, Philadelphia, 2003.

[6] Y. Saad and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 7 (1986), 856–869.

[7] H. Elman, D. Silvester and A. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2006.

[8] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.