

A QUICK INTRODUCTION TO KRYLOV SOLVERS FOR THE SOLUTION OF LINEAR SYSTEMS

L. Giraud (Inria)

HiePACS project - Inria Bordeaux Sud-Ouest

Inria School
Nov. 5, 2019

Outline

Reliability of the calculation

Algorithm selection

Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

Algebraic preconditioning techniques

Bibliography

Outline

Reliability of the calculation

- Backward error

- Sensitivity v.s. conditioning

- Backward stability of algorithms

Algorithm selection

Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

Algebraic preconditioning techniques

Bibliography

How to assess the quality of a computed solution ?

Relative norm should be preferred. Let x and \tilde{x} be the solution and its approximation, $\delta = \frac{\|x - \tilde{x}\|}{\|x\|}$ gives the number of correct digits in the solution. Example: $e = 2.7182818\dots$

Approximation	δ
2.	$2 \cdot 10^{-1}$
2.7	$6 \cdot 10^{-3}$
2.71	$3 \cdot 10^{-3}$
2.718	$1 \cdot 10^{-4}$
2.7182	$3 \cdot 10^{-5}$
2.71828	$6 \cdot 10^{-7}$

IEEE 754 standard only provides relative accuracy information.

How to assess the quality of a computed solution ?

If $\frac{\|x - \tilde{x}\|}{\|x\|}$ is large, **TWO** possible reasons:

- ▶ the mathematical problem is sensible to perturbations
- ▶ the selected numerical algorithm behaves poorly in finite precision calculation

The backward error analysis (Wilkinson, 1963) gives a framework to look at the problem.

Sensitivity to small perturbations (exact arithmetic)

$$\begin{pmatrix} 1 & 1 \\ 1.00000\underline{5} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2.000005 \end{pmatrix}$$

⇓

$$\begin{pmatrix} 1 & 1 \\ 1.00000\underline{6} & 1 \end{pmatrix} \begin{pmatrix} 0.83333\dots \\ 1.16666\dots \end{pmatrix} = \begin{pmatrix} 2 \\ 2.000005 \end{pmatrix}$$

Poor numerical behaviour in finite precision

Computation of $I_n = \int_0^1 x^n e^{-x} dx$, $n > 0$

By part integration $I_0 = 1 - 1/e$, $I_n = nI_{n-1} - 1/e$, $n > 1$

Computation with 16 significative digits:

$$\tilde{I}_{100} = 5.7 \cdot 10^{+141}$$

Backward recurrence: $I_{300} = 1(??)$, $I_{n-1} = \frac{1}{n} \left(I_n + \frac{1}{e} \right)$

Computation with 16 significative digits:

$$\tilde{I}_{100} = 3.715578714528098... \cdot 10^{-3}$$

Exact value:

$$\tilde{I}_{100} = 3.715578714528085... \cdot 10^{-03}$$

The art to select the good algorithm !!
More calculation does not imply larger errors !!

Backward error and conditioning

The backward error analysis, introduced by Wilkinson (1963), is a powerful concept for analyzing the quality of an approximate solution:

1. it is independent of the details of round-off propagation: the errors introduced during the computation are interpreted in terms of perturbations of the initial data, and the computed solution is considered as exact for the perturbed problem;
2. because round-off errors are seen as data perturbations, they can be compared with errors due to numerical approximations (consistency of numerical schemes) or to physical measurements (uncertainties on data coming from experiments for instance).

Backward error

Rigal and Gaches result (1967): the normwise backward error

$$\eta_{A,b}^N(\tilde{x}) = \min\{\varepsilon : (A + \Delta A)\tilde{x} = b + \Delta b, \quad (1)$$
$$\|\Delta A\| \leq \varepsilon\|A\|, \|\Delta b\| \leq \varepsilon\|b\|\}$$

is given by

$$\eta_{A,b}^N(\tilde{x}) = \frac{\|r\|}{\|A\|\|\tilde{x}\| + \|b\|} \quad (2)$$

where $r = b - A\tilde{x}$.

Backward error

Simplified proof ($\|\cdot\| = \|\cdot\|_2$): Right-hand side of (2) is a lower bound of (1). Let $(\Delta A, \Delta b)$ such that

$$(A + \Delta A)\tilde{x} = b + \Delta b, \|\Delta A\| \leq \varepsilon\|A\| \text{ and } \|\Delta b\| \leq \varepsilon\|b\|.$$

We have

$$\begin{aligned} & (A + \Delta A)\tilde{x} = b + \Delta b \\ \Rightarrow & b - A\tilde{x} = \Delta b - \Delta A\tilde{x} \\ \Rightarrow & \|b - A\tilde{x}\| \leq \|\Delta A\|\|\tilde{x}\| + \|\Delta b\| \\ \Rightarrow & \|r\| \leq \varepsilon(\|A\|\|\tilde{x}\| + \|b\|) \\ \Rightarrow & \frac{\|r\|}{\|A\|\|\tilde{x}\| + \|b\|} \leq \min\{\varepsilon\} = \eta_{A,b}^N(\tilde{x}) \end{aligned}$$

The bound is attained for

$$\Delta A_{min} = \frac{\|A\|}{\|\tilde{x}\|(\|A\|\|\tilde{x}\| + \|b\|)} r \tilde{x}^T$$

and

$$\Delta b_{min} = -\frac{\|b\|}{\|A\|\|\tilde{x}\| + \|b\|} r.$$

We have $\Delta A_{min} \tilde{x} - \Delta b_{min} = r$ with

$$\|\Delta A_{min}\| = \frac{\|A\|\|r\|}{\|A\|\|\tilde{x}\| + \|b\|} \quad \text{and} \quad \|\Delta b_{min}\| = \frac{\|b\|\|r\|}{\|A\|\|\tilde{x}\| + \|b\|}.$$

Normwise backward error

We can also define

$$\begin{aligned}\eta_b^N(\tilde{x}) &= \min\{\varepsilon : A\tilde{x} = b + \Delta b, \|\Delta b\| \leq \varepsilon\|b\|\} \\ &= \frac{\|r\|}{\|b\|}\end{aligned}$$

- ▶ Classically $\eta_{A,b}^N$ or η_b^N are considered to implement stopping criterion in iterative methods.
- ▶ Notice that $\frac{\|r_k\|}{\|r_0\|}$ (often seen in some implementations) reduces to η_b^N if $x_0 = 0$.

Sensitivity v.s. conditioning

Let suppose that we have solved $(A + \Delta A)(x + \Delta x) = b + \Delta b$.

We would like to know how $\frac{\|\Delta x\|}{\|x\|}$ depends on $\frac{\|\Delta A\|}{\|A\|}$ and $\frac{\|\Delta b\|}{\|b\|}$.

We denote $\omega = \max \left\{ \frac{\|\Delta A\|}{\|A\|}, \frac{\|\Delta b\|}{\|b\|} \right\}$.

$$(A + \Delta A)(x + \Delta x) = b + \Delta b$$

$$\Rightarrow \Delta x = -A^{-1}\Delta A x - A^{-1}\Delta A \Delta x + A^{-1}\Delta b$$

$$\Rightarrow \|\Delta x\| \leq \|A^{-1}\| \frac{\|\Delta A\|}{\|A\|} \|A\| \|x\| + \|A^{-1}\| \frac{\|\Delta A\|}{\|A\|} \|A\| \|\Delta x\| \\ + \|A^{-1}\| \|b\| \frac{\|\Delta b\|}{\|b\|}$$

$$\Rightarrow \|\Delta x\| \leq \omega \kappa^N(A) \|x\| + \kappa^N(A) \omega \|\Delta x\| + \omega \|A^{-1}\| \|b\|$$

$$\Rightarrow (1 - \omega \kappa^N(A)) \|\Delta x\| \leq \omega \kappa^N(A) \|x\| + \omega \|A^{-1}\| \|b\|$$

Sensitivity v.s. conditioning (cont)

If $\omega\kappa^N(A) < 1$ then

$$\begin{aligned}\frac{\|\Delta x\|}{\|x\|} &\leq \frac{\omega}{1 - \omega\kappa^N(A)} \left(\kappa^N(A) + \frac{\|A^{-1}\| \|b\|}{\|x\|} \right) & (3) \\ &\leq 2\omega \left(\kappa^N(A) + \frac{\|A^{-1}\| \|b\|}{\|x\|} \right) \text{ if } \omega\kappa^N(A) < 0.5 \\ &\leq 4\omega\kappa^N(A)\end{aligned}$$

that reads

Relative Forward error \lesssim (Condition number) \times (Backward error)

First order estimate of the forward error

Matrix	$\kappa^N(A)$	$\eta_{A,b}^N$	$\kappa^N(A) \times \eta_{A,b}^N$	FE
M1	$1 \cdot 10^2$	$1 \cdot 10^{-16}$	$1 \cdot 10^{-14}$	$2 \cdot 10^{-15}$
M2	$4 \cdot 10^5$	$9 \cdot 10^{-17}$	$3 \cdot 10^{-11}$	$8 \cdot 10^{-12}$
M3	$1 \cdot 10^{16}$	$2 \cdot 10^{-16}$	2	$6 \cdot 10^{-1}$
M4	$3 \cdot 10^1$	$3 \cdot 10^{-7}$	$2 \cdot 10^{-5}$	$9 \cdot 10^{-6}$
M5	$4 \cdot 10^1$	$6 \cdot 10^{-4}$	$3 \cdot 10^{-2}$	$2 \cdot 10^{-2}$

Algorithm: Gauss elimination with partial pivoting (MATLAB).

Matrices :

M1 - augment(20) M2 - dramadah(2) M3 - chebvand(20)

M4 - will(40) M5 - will(50) M6 - dd(20)

First order estimate of the forward error (cont)

- ▶ Large backward error: unreliable algorithm
⇒ change the algorithm (ex. use Gauss elimination with pivoting).
- ▶ Ill-conditioned matrices: large condition number
⇒ scale the matrix ,
⇒ think about alternative for the numerical formulation of the problem.

Gaussian elimination with partial pivoting

```
for k = 1 to n - 1 do
  for i = k + 1 to n do
    aik = aik/akk
    for j = k + 1 to n do
      aij = aij - aikakj
    end for
  end for
end for
```

LU factorization algorithm without pivoting

Let \tilde{x} the solution computed by Gaussian elimination with partial pivoting.

Then $(A + \Delta A)\tilde{x} = b$ with $\frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 8n^3\tau\psi + \mathcal{O}(\psi^2)$ where ψ is the machine precision. τ is the *growth factor*,

$\tau = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}$. If τ is large it might imply numerical instability (i.e. large backward error)

Backward stability of GMRES with robust orthogonalization

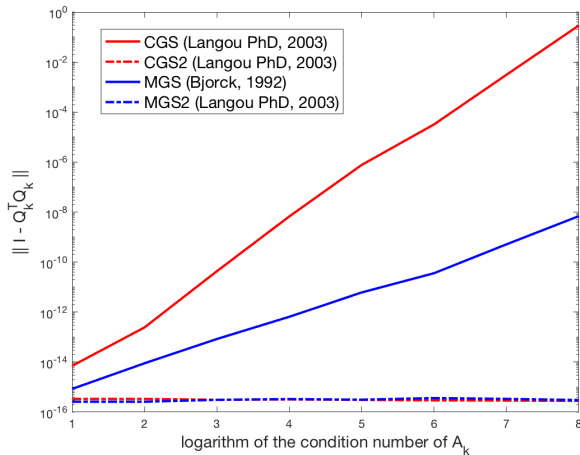
For not too ill-conditioned matrix, the Householder GMRES implementation ensures that

$$\eta_{A,b}^N(\tilde{x}_n) = \frac{\|b - Ax_n\|}{\|A\| \|\tilde{x}\| + \|b\|} \leq \mathcal{P}(n)\psi + \mathcal{O}(\psi^2)$$

[J. Drkošová, M. Rozložník, Z. Strakoš and A. Greenbaum, *Numerical stability of the GMRES method*, BIT, vol. 35, p. 309-330, 1995.]

[C.C. Paige, M. Rozložník and Z. Strakoš, *Modified Gram-Schmidt (MGS), Least Squares, and Backward Stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., vol. 28 (1), p. 264-284, 2006.]

Backward stability of GMRES with robust orthogonalization



Outline

Reliability of the calculation

Algorithm selection

Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

Algebraic preconditioning techniques

Bibliography

Sparse linear solver

Goal: solving $\mathcal{A}x = b$, where \mathcal{A} is **sparse**



Usual trades off

Direct

- ▶ Robust/accurate for general problems
- ▶ BLAS-3 based implementations
- ▶ Memory/CPU prohibitive for large 3D problems
- ▶ Limited weak scalability

Iterative

- ▶ Problem dependent efficiency / accuracy
- ▶ Sparse computational kernels
- ▶ Less memory requirements and possibly faster
- ▶ Possible high weak scalability

Algorithm selection

Two main approaches:

- ▶ Direct solvers:

compute a factorization and use the factors to solve the linear system; that is, express the matrix A as the product of matrices having simple structures (i.e. diagonal, triangular).
Example: LU for unsymmetric matrices, LL^T (Cholesky) for symmetric positive definite matrices, LDL^T for symmetric indefinite matrices.

- ▶ Iterative solvers:

build a sequence $(x_k) \rightarrow x^*$.

Stationary v.s. Krylov methods

Alternative to direct solvers when memory and/or CPU constraints.

Two main approaches

- ▶ Stationary/asymptotic method:

$$x_k = f(x_{k-1})$$

with $x_k \rightarrow x^* \forall x_0$.

- ▶ Krylov method:

$$x_k = x_0 + \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\}$$

with $r_k = b - Ax_k$ subject to some constraints/optimality conditions.

Basic scheme

Let x_0 be given and $M \in \mathbb{R}^{n \times n}$ a nonsingular matrix, compute

$$x_k = x_{k-1} + M(b - Ax_{k-1}). \quad (4)$$

Note that $b - Ax_{k-1} = A(x_* - x_{k-1}) \Rightarrow$ the best M is A^{-1} .

Theorem

The stationary scheme defined by (4) converges to $x^ = A^{-1}b$ for any x_0 iff $\rho(I - MA) < 1$, where $\rho(I - MA)$ denotes the spectral radius of the iteration matrix $(I - MA)$.*

Some well-known schemes

Depending on the choice of M we obtain some of the best known stationary methods. Let decompose $A = L + D + U$, where L is lower triangular part of A , U the upper triangular part and D is the diagonal of A .

- ▶ $M = I$: Richardson method,
- ▶ $M = D^{-1}$: Jacobi method,
- ▶ $M = (L + D)^{-1}$: Gauss-Seidel method.

Notice that M has always a special structure and inverse must never been explicitly computed. $z = M^{-1}y$ reads *solve the linear system $Mz = y$* .

Outline

Reliability of the calculation

Algorithm selection

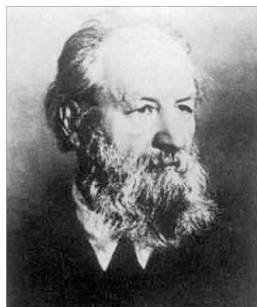
Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

Algebraic preconditioning techniques

Bibliography

Krylov method : some background



Aleksei Nikolaevich Krylov

1863-1945: Russia, Maritime Engineer

His research spans a wide range of topics, including shipbuilding, magnetism, artillery, mathematics, astronomy, and geodesy. In 1904 he built the first machine in Russia for integrating ODEs.

In 1931 he published a paper on what is now called the "Krylov subspace".

Definition

Let $A \in \mathbb{R}^{n \times n}$ and $r \in \mathbb{R}^n$ the space denoted by $\mathcal{K}(b, A, m)$ (with $m \leq n$) and defined by

$$\mathcal{K}(b, A, m) = \text{Span}\{b, Ab, \dots, A^{m-1}b\}$$

is referred to as the Krylov space of dimension m associated with A and r .

Why using this search space ?

For the sake of simplicity of exposure, we often assume $x_0 = 0$. This does not mean a loss of generality, because the situation $x_0 \neq 0$ can be transformed with a simple shift to the system

$$Ay = b - Ax_0 = \bar{b},$$

for which obviously $y_0 = 0$. The minimal polynomial $q(t)$ of A is the unique monic polynomial of minimal degree such that $q(A) = 0$. It is constructed from the eigenvalues of A as follows. If the distinct eigenvalues of A are $\lambda_1, \dots, \lambda_\ell$ and if λ_j has index m_j (the size of the largest Jordan block associated with λ_j), then the sum of all indices is

$$m = \sum_{j=1}^{\ell} m_j, \text{ and } q(t) = \prod_{j=1}^{\ell} (t - \lambda_j)^{m_j}. \quad (5)$$

When A is diagonalizable ℓ is the number of distinct eigenvalues of A ; when A is a Jordan block of size n , then $m = n$.

If we write

$$q(t) = \prod_{j=1}^{\ell} (t - \lambda_j)^{m_j} = \sum_{j=0}^m \alpha_j t^j,$$

then the constant term $\alpha_0 = \prod_{j=1}^{\ell} (-\lambda_j)^{m_j}$. Therefore $\alpha_0 \neq 0$ iff A is nonsingular. Furthermore, from

$$0 = q(A) = \alpha_0 I + \alpha_1 A + \dots + \alpha_m A^m, \quad (6)$$

it follows that

$$A^{-1} = -\frac{1}{\alpha_0} \sum_{j=0}^{m-1} \alpha_{j+1} A^j.$$

This description of A^{-1} portrays $x = A^{-1}b$ immediately as a member of the Krylov space of dimension m associated with A and b denoted by $\mathcal{K}(b, A, m) = \text{Span}\{b, Ab, \dots, A^{m-1}b\}$.

Taxonomy of the Krylov subspace approaches

The Krylov methods for identifying $x_m \in \mathcal{K}(A, b, m)$ can be distinguished in four classes:

- ▶ **The Ritz-Galerkin approach:**
construct x_m such that $b - Ax_m \perp \mathcal{K}(A, b, m)$.
- ▶ **The minimum norm residual approach:**
construct $x_m \in \mathcal{K}(A, b, m)$ such that $\|b - Ax_m\|_2$ is minimal
- ▶ **The Petrov-Galerkin approach:**
construct x_m such that $b - Ax_m$ is orthogonal to some other m -dimensional subspace.
- ▶ **The minimum norm error approach:**
construct $x_m \in A^T \mathcal{K}(A, b, m)$ such that $\|b - Ax_m\|_2$ is minimal.

Constructing a basis of $\mathcal{K}(A, b, m)$

- ▶ The obvious choice $b, Ab, \dots, A^{m-1}b$ is not very attractive from a numerical point of view since the vectors $A^i b$ becomes more and more colinear to the eigenvector associated with the dominant eigenvalue. In finite precision calculation, this leads to a lost of rank of this set of vectors.

Suppose that A is diagonalizable $A = VDV^{-1}$ and denote \underline{b} the component of b in V . In V , $A^m b \equiv VD^m \underline{b}$.

- ▶ A better choice is to use the Arnoldi procedure.

Outline

Reliability of the calculation

Algorithm selection

Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

- The FOM variants

- The GMRES variants

- Strategies for restarted GMRES

Algebraic preconditioning techniques

Bibliography

Arnoldi

Walter Edwin Arnoldi

1917-1995: USA.

His main research subjects covered vibration of propellers, engines and aircraft, high speed digital computers, aerodynamics and acoustics of aircraft propellers, lift support in space vehicles and structural materials.

"The principle of minimized iterations in the solution of the eigenvalue problem" in Quart. of Appl. Math., Vol.9 in 1951.

The Arnoldi procedure

This procedure builds an orthonormal basis of $\mathcal{K}(A, b, m)$.

ARNOLDI'S ALGORITHM

- 1: $v_1 = b/\|b\|$
- 2: **for** $j = 1, 2, \dots, m - 1$ **do**
- 3: Compute $h_{i,j} = v_i^T Av_j$ for $i = 1, \dots, j$
- 4: Compute $w_j = Av_j - \sum_{i=1}^j h_{i,j}v_i$
- 5: Compute $h_{j+1,j} = \|w_j\|$
- 6: Exit if ($h_{j+1,j} = 0$)
- 7: Compute $v_{j+1} = w_j/h_{j+1,j}$
- 8: **end for**

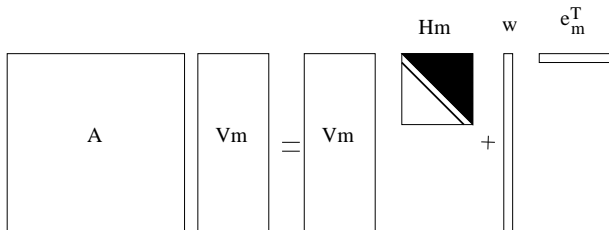
Proposition

Denote V_m the $n \times m$ matrix with column vector v_1, \dots, v_m ; \bar{H}_m the $(m+1) \times m$ Hessenberg matrix whose nonzero entries are h_{ij} and by H_m the square matrix obtained from \bar{H}_m by deleting its last row. Then the following relations hold:

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \quad (7)$$

$$= V_{m+1} \bar{H}_m \quad (8)$$

$$V_m^T AV_m = H_m \quad (9)$$



Surprising consequence of symmetry

- ▶ Equation (9) tells us that H_m is tridiagonal
- ▶ Consequently, the Arnoldi's orthogonalisation can only be performed with respect to the last two computed basis vectors
→ short term recurrence algorithms

Arnoldi for linear systems

This technique belongs to the **Ritz-Galerkin** techniques; it seeks an approximate solution x_m in the affine subspace $x_0 + \mathcal{K}_m$ by imposing the condition

$$b - Ax_m \perp \mathcal{K}_m$$

If $v_1 = \frac{b}{\beta}$ with $\beta = \|b\|$ in Arnoldi's method. $x_m \in \mathcal{K}_m$ means that $\exists y_m \in \mathbb{R}^m$ such that $x_m = V_m y_m$. Consequently, $b - Ax_m \perp \mathcal{K}_m$ reads

$$\begin{aligned} V_m^T (b - AV_m y_m) &= 0 \\ \Leftrightarrow V_m^T (b - AV_m y_m) &= 0 \\ \Leftrightarrow \beta e_1 - H_m y_m &= 0 \\ \Leftrightarrow y_m &= H_m^{-1}(\beta e_1) \end{aligned}$$

The FOM technique

The full orthogonalization method (FOM) is based on this approach.

FOM ALGORITHM

- 1: Set the initial guess $x_0 = 0$
- 2: $r_0 = b$; $\beta = \|r_0\|$ $v_1 = r_0/\|r_0\|$;
- 3: **for** $j = 1, 2, \dots$ **do**
- 4: $w_j = Av_j$
- 5: **for** $i = 1$ to j **do**
- 6: $h_{ij} = v_i^T w_j$
- 7: $w_j = w_j - h_{i,j}v_i$
- 8: **end for**
- 9: $h_{j+1,j} = \|w_j\|$
- 10: If $h_{j+1,j} = 0$ set $m = j$ Goto 13
- 11: $v_{j+1} = w_j/h_{j+1,j}$
- 12: **end for**
- 13: $y_m = H_m^{-1}(\beta e_1)$ and $x_m = x_0 + V_m y_m$.

The FOM technique

Proposition

The residual vector of the approximate solution x_m computed by FOM is such that

$$b - Ax_m = -h_{m+1,m}e_m^T y_m v_{m+1}$$

and

$$\|b - Ax_m\| = h_{m+1,m}|e_m^T y_m|. \quad (10)$$

$$\begin{aligned} b - Ax_m &= b - AV_m y_m \\ &= b - AV_m y_m \\ &= \beta v_1 - V_m H_m y_m - h_{m+1,m} v_{m+1} e_m^T y_m \\ &= V_m \underbrace{(\beta e_1 - H_m y_m)}_0 - h_{m+1,m} (e_m^T y_m) v_{m+1} \end{aligned}$$

Cost of FOM

Flops cost:

1. At step j , the Gram-Schmidt process costs: $\approx 4nj$ flops
dot product $\approx 2n$, saxpy $\approx 2n$
2. The matrix-vector product costs: $2nnz(A) - n$

Over m steps this leads to $\approx 2 \cdot m \cdot nnz(A) + 2 \cdot m^2 \cdot n$.

Storage: the most consuming part is the basis V_m that is $m \cdot n$.
If the convergence of the algorithm is slow (i.e. m large), it becomes unaffordable.

\Rightarrow Alternative:

1. restart,
2. truncate.

The restarted alternative: FOM(m)

FOM(M) ALGORITHM

- 1: Set the initial guess x_0
- 2: $r_0 = b - Ax_0$; $\beta = \|r_0\|$ $v_1 = r_0/\|r_0\|$;
- 3: **for** $j = 1, \dots, m$ **do**
- 4: $w_j = Av_j$
- 5: **for** $i = 1$ to j **do**
- 6: $h_{ij} = v_i^T w_j$
- 7: $w_j = w_j - h_{i,j}v_i$
- 8: **end for**
- 9: $h_{j+1,j} = \|w_j\|$
- 10: $v_{j+1} = w_j/h_{j+1,j}$
- 11: **end for**
- 12: $y_m = H_m^{-1}(\beta e_1)$ and $x_m = x_0 + V_m y_m$, If converged then Stop.
- 13: Set $x_0 = x_m$ goto 2

The truncated alternative: DIOM(m)

Governing idea: perform an incomplete orthogonalization in the Arnoldi process (window of width m) and exploit the structure of the resulting H_k to perform an incremental LU factorization of H_k and then derive an update of the iterate at each step of the algorithm.

Advantage: the cost of the orthogonalisation is reduced as well as the cost of the storage (only the last m vectors are stored).

Example with $k = 5$ and $m = 3$

$$H_5 = \begin{pmatrix} h_{11} & h_{12} & h_{13} & & & \\ h_{21} & h_{22} & h_{23} & h_{24} & & \\ & h_{32} & h_{33} & h_{34} & h_{35} & \\ & & h_{43} & h_{44} & h_{45} & \\ & & & h_{54} & h_{55} & \end{pmatrix}$$

Notice that we still have $AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T$ but V_k is no longer orthogonal.

The basics of the LU factorization

$$A = \begin{pmatrix} a_{1,1} & w^T \\ v & A_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ a_{1,1}^{-1}v & I \end{pmatrix} \begin{pmatrix} a_{1,1} & w^T \\ 0 & A^{(1)} \end{pmatrix}$$

with $A^{(1)} = A_{n-1} - \frac{vw^T}{\alpha}$

Typical Gaussian elimination step k :

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)} \cdot a_{kj}^{(k)}}{a_{kk}^{(k)}}$$

$$H_6 \equiv \left(\begin{array}{cccc|c} u_{11} & u_{12} & h_{13} & & \\ l_{21} & u_{22} & u_{23} & h_{24} & \\ & l_{32} & u_{33} & u_{34} & h_{35} \\ & & l_{43} & u_{44} & u_{45} & h_{46} \\ & & & l_{54} & u_{55} & u_{56} \\ \hline & & & & l_{65} & u_{66} \end{array} \right),$$

where u_{56} , l_{65} and u_{66} (i.e. the update of the LU factorization of H_6) can be computed using

$$\left(\begin{array}{cc|c} u_{44} & u_{45} & h_{46} \\ l_{54} & u_{55} & h_{56} \\ \hline & h_{65} & h_{66} \end{array} \right).$$

DIOM(M) ALGORITHM

- 1: Set the initial guess x_0
- 2: $r_0 = b - Ax_0$; $\beta = \|r_0\|$ $v_1 = r_0/\|r_0\|$;
- 3: **for** $j = 1, \dots$ **do**
- 4: $w_j = Av_j$
- 5: **for** $i = \max\{1, j - m + 1\}$ to j **do**
- 6: $h_{i,j} = v_i^T w_j$
- 7: $w_j = w_j - h_{i,j}v_i$
- 8: **end for**
- 9: $h_{j+1,j} = \|w_j\|$ $v_{j+1} = w_j/h_{j+1,j}$
- 10: Update the LU factorization of H_j using the LU factors of H_{j-1}
- 11: $\xi_j = \{\text{if } j = 1 \text{ then } \beta \text{ else } -l_{j,j-1}\xi_{j-1}\}$
- 12: $p_j = u_{jj}^{-1} \left(v_j - \sum_{i=j-k+1}^{j-1} u_{ij}p_i \right)$ (for $i \leq 0$ set $u_{ij}p_i = 0$)
- 13: $x_j = x_{j-1} + \xi_j p_j$
- 14: **end for**

For the sake of simplicity of exposure the DIOM(m) algorithm as been derived using a LU factorization without pivoting, a variant with partial pivoting exists.

The Generalized Minimal RESidual method (GMRES)

The Generalized Minimal RESidual method (GMRES) based on the minimum residual approach. Using the Arnoldi's relation

$$AV_m = V_{m+1}\bar{H}_m,$$

the algorithm builds the iterate $x_m \in \mathcal{K}(A, b, m)$ such that $\|b - Ax_m\|_2$ is minimal.

Because $x_m \in \mathcal{K}(A, b, m)$, $\exists y$ such that

$$x_m = V_m y.$$

Central equality

$$\begin{aligned}\min \|b - Ax_m\| &= \min \|b - AV_my\| \\ &= \min \|b - AV_my\| \\ &= \min \|\beta v_1 - V_{m+1} \bar{H}_m y\| \text{ where } \beta = \|b\| \\ &= \min \|\beta V_{m+1} e_1 - V_{m+1} \bar{H}_m y\| \\ &= \min \|V_{m+1}(\beta e_1 - \bar{H}_m y)\|.\end{aligned}$$

Because the columns of V_{m+1} are orthonormal

$$\|b - Ax_m\|_2 = \|\beta e_1 - \bar{H}_m y\|.$$

The GMRES method

The GMRES iterate is the vector of $\mathcal{K}(A, b, m)$ such that

$$\begin{aligned}x_m &= V_m y_m \text{ where} \\y_m &= \arg \min_{y \in \mathbb{R}^m} \|\beta \mathbf{e}_1 - \bar{H}_m y\|_2.\end{aligned}$$

The minimizer y_m is inexpensive to compute since it requires only the solution of an $(m + 1) \times m$ linear least-squares problem.

The GMRES algorithm

GMRES ALGORITHM

- 1: Set the initial guess x_0 ; $r_0 = b - Ax_0$; $\beta = \|r_0\|$
- 2: $v_1 = r_0/\|r_0\|$;
- 3: **for** $j = 1, \dots, m$ **do**
- 4: $w_j = Av_j$
- 5: **for** $i = 1$ **to** j **do**
- 6: $h_{i,j} = v_i^T w_j$; $w_j = w_j - h_{i,j}v_i$
- 7: **end for**
- 8: $h_{j+1,j} = \|w_j\|$
- 9: **if** $(h_{j+1,j}) = 0$ **goto** 12
- 10: $v_{j+1} = w_j/h_{j+1,j}$
- 11: **end for**
- 12: Define the $(j + 1) \times j$ upper Hessenberg matrix \bar{H}_j
- 13: Solve the least-squares problem $y_j = \arg \min \|\beta e_1 - \bar{H}_j y\|$
- 14: Set $x_j = x_0 + V_j y_j$

[Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 1986.]

Incremental QR factorization

The QR factorization of \bar{H}_{m+1} can be cheaply computed from the QR factorization of \bar{H}_m .

$$\bar{H}_{m+1} = \left(\begin{array}{c|c} \bar{H}_m & \begin{matrix} h_{1,m+1} \\ \vdots \\ h_{m+1,m+1} \end{matrix} \\ \hline 0 & h_{m+2,m+1} \end{array} \right).$$

It is enough to apply the m Givens rotations computed to factor \bar{H}_m to the last column of \bar{H}_{m+1} and build and apply a $(m+1)^{th}$ rotation to zero the $h_{m+2,m+1}$ entry.

Happy breakdown in GMRES

It exists one possible breakdown in the GMRES algorithm if $h_{k+1,k} = 0$ which prevents to increase the dimension of the search space.

Proposition

Let A be a nonsingular matrix. Then the GMRES algorithm breaks down at step k (i.e. $h_{k+1,k} = 0$) iff the iterate x_k is the exact solution of $Ax = b$.

Cost of GMRES

- ▶ If all but the last Given's rotation implemented by GMRES are applied we end-up with a QR factorization of H_m that can be used by FOM to compute $H_m^{-1}(\beta e_1)$.
- ▶ The memory and computational cost of GMRES are very similar to those of FOM.
- ▶ To alleviate the cost of the computation (in Gram-Schmidt process) and reduce the memory requirement, the two ideas implemented in FOM(m) and DIOM(m) can be applied to GMRES
 1. restarting GMRES(m),
 2. truncating DQGMRES(m) (Direct Quasi GMRES).

The restarted alternative: GMRES(m)

GMRES(M) ALGORITHM

- 1: Set the initial guess x_0
- 2: $r_0 = b - Ax_0$; $\beta = \|r_0\|$ $v_1 = r_0/\|r_0\|$;
- 3: **for** $j = 1, \dots, m$ **do**
- 4: $w_j = Av_j$
- 5: **for** $i = 1$ to j **do**
- 6: $h_{i,j} = v_i^T w_j$; $w_j = w_j - h_{i,j}v_i$
- 7: **end for**
- 8: $h_{j+1,j} = \|w_j\|$; $v_{j+1} = w_j/h_{j+1,j}$
- 9: **end for**
- 10: $y_m = \arg \min \|\beta e_1 - \bar{H}_m y\|$ and $x_m = x_0 + V_m y_m$, If converged then Stop.
- 11: Set $x_0 = x_m$ goto 2

The DQGMRES(m) algorithm

DQGMRES(M) ALGORITHM

- 1: Set the initial guess x_0 $r_0 = b - Ax_0$; $\beta = \|r_0\|$ $v_1 = r_0/\|r_0\|$;
- 2: **for** $j = 1, \dots, k$ **do**
- 3: $w_j = Av_j$
- 4: **for** $i = \max\{1, j - m + 1\}$ **to** j **do**
- 5: $h_{i,j} = v_i^T w_j$; $w_j = w_j - h_{i,j}v_i$
- 6: **end for**
- 7: $h_{j+1,j} = \|w_j\|$; $v_{j+1} = w_j/h_{j+1,j}$
- 8: Update the QR factorization of \bar{H}_j
- 9: **for** $i = j - k, \dots, j - 1$ **do**
- 10: Apply Q_i to the last column of \bar{H}_j
- 11: **end for**
- 12: Apply Q_j to \bar{H}_j that is Compute c_j and s_j , $\gamma_{j+1} = -s_j\gamma_j$, $\gamma_j = c_j\gamma_j$,
 $h_{jj} = c_j h_{j,j} + s_j h_{j+1,j}$
- 13: $p_j = h_{j,j}^{-1} \left(v_j - \sum_{i=k-m}^{j-1} h_{i,j} p_i \right)$; $x_j = x_{j-1} + \gamma_j p_j$
- 14: If converged then stop
- 15: **end for**

[Y. Saad and K. Wu, *DQGMRES: a direct quasi-minimal residual algorithm based on incomplete orthogonalisation*, *Numerical Linear Algebra with Applications*, 1996.]

Some relations between these Krylov methods

Proposition

1. GMRES v.s. FOM

Assume that m steps of GMRES and FOM have been performed (the step of FOM with a singular H_m are skipped). Let $\|r_{m^*}^{FOM}\|$ the smallest residual norm achieved in the first m steps and $\|r_m^{GMRES}\|$, the residual norm associated with the iterate computed by GMRES. We have:

$$\|r_m^{GMRES}\| \leq \|r_{m^*}^{FOM}\| \leq \sqrt{m} \|r_m^{GMRES}\|.$$

2. DQGMRES v.s. GMRES

Assume that V_{m+1} the Arnoldi vectors generated by DQGMRES is of full rank. We have the following relation between $\|r_m^{DQGMRES}\|$, the residual norm associated with the iterate computed by DQGMRES at step m and $\|r_m^{GMRES}\|$, the residual norm associated with the iterate computed by GMRES.

$$\|r_m^{DQGMRES}\| \leq \kappa(V_{m+1}) \|r_m^{GMRES}\|$$

Strategies at restart

Motivations

- ▶ In GMRES(m), only the approximate solution is kept between cycles/restarts.
- ▶ It is often observed that part of the spectrum slow down convergence.
- ▶ Attempt to **augment/deflate** the search space with approximated eigenvectors, that are extracted from the Krylov subspace at the end of each cycle.

Outline

Reliability of the calculation

Algorithm selection

Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

Algebraic preconditioning techniques

- Driving principles

- Preconditioner taxonomy and location

- Some classical algebraic preconditioners

Bibliography

Algebraic preconditioners

Outline

- ▶ Some backgrounds
- ▶ Governing principles
- ▶ Preconditioner taxonomy and examples
- ▶ Spectral corrections

Some properties of Krylov solvers

- ▶ CG

$$\|x_k - x^*\|_A = \min_{p \in P_{k-1}, p(0)=1} \|p(A)(x_0 - x^*)\|_A.$$

where P_{k-1} is the set of polynomial of degree $(k - 1)$.

- ▶ GMRES, MINRES

$$\|r_k\| = \min_{p \in P_{k-1}, p(0)=1} \|p(A)(r_0)\|_2.$$

- ▶ **Assumption:** if A diagonalizable with m distinct eigenvalues **then** CG, GMRES, MINRES converges in at most m steps. (Cayley-Hamilton theorem - minimal polynomial).
- ▶ **Assumption:** if r_0 has k components in the eigenbasis **then** CG, GMRES, MINRES converges in k steps.

Some properties of Krylov solvers

- ▶ Bound on the rate of convergence of CG

$$\|x_k - x^*\|_A \leq 2 \cdot \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A.$$

Driving principles to design preconditioners

Find a non-singular matrix M such that MA has “better” properties v.s. the convergence behaviour of the selected Krylov solver

- ▶ MA has less distinct eigenvalues,
- ▶ $MA \approx I$ in some sense.

The preconditioner constraints

The preconditioner should

- ▶ be cheap to compute and to store,
- ▶ be cheap to apply,
- ▶ ensure a fast convergence.

With a good preconditioner the solution time for the preconditioned system should be significantly less than for the unpreconditioned system.

MA has less distinct eigenvalues: an example

$$\text{Let } \mathcal{A} = \begin{pmatrix} A & B^T \\ C & 0 \end{pmatrix} \text{ and } \mathcal{P} = \begin{pmatrix} A & 0 \\ 0 & CA^{-1}B^T \end{pmatrix}.$$

Then $\mathcal{P}^{-1}\mathcal{A}$ has three distinct eigenvalues.

[Murphy, Golub, Wathen, SIAM SISC, 21 (6), 2000]

Preconditioner taxonomy

There are two main classes of preconditioners

- ▶ **Implicit preconditioners:**

approximate A with a matrix M such that solving the linear system $Mz = r$ is easy.

- ▶ **Explicit preconditioners:**

approximate A^{-1} with a matrix M and just perform $z = Mr$.

The governing ideas in the design of the preconditioners are very similar to those followed to define iterative stationary schemes. Consequently, all the stationary methods can be used to define preconditioners.

Stationary methods

Let x_0 be given and $M \in \mathbb{R}^{n \times n}$ a nonsingular matrix, compute

$$x_k = x_{k-1} + M(b - Ax_{k-1}).$$

Note that $b - Ax_{k-1} = A(x^* - x_{k-1}) \Rightarrow$ the best M is A^{-1} .
The stationary scheme converges to $x^* = A^{-1}b$ for any x_0 iff $\rho(I - MA) < 1$, where $\rho(\cdot)$ denotes the spectral radius.

Let $A = L + D + U$

- ▶ $M = I$: Richardson method,
- ▶ $M = D^{-1}$: Jacobi method,
- ▶ $M = (L + D)^{-1}$: Gauss-Seidel method.

Notice that M has always a special structure and the inverse must never be explicitly computed ($z = B^{-1}y$ reads *solve the linear system $Bz = y$*).

Preconditioner location

Several possibilities exist to solve $Ax = b$:

- ▶ Left preconditioner

$$MAx = Mb.$$

- ▶ Right preconditioner

$$AMy = b \text{ with } x = My.$$

- ▶ Split preconditioner if $M = M_1M_2$

$$M_2AM_1y = M_2b \text{ with } x = M_1y.$$

Notice that the spectrum of MA , AM and M_2AM_1 are identical (for any matrices B and C , the eigenvalues of BC are the same as those of CB)

Preconditioner location v.s. stopping criterion

The stopping criterion is based on backward error

$$\eta_{A,b}^N = \frac{\|b - Ax\|}{\|A\|\|x\| + \|b\|} < \varepsilon \text{ or } \eta_b^N = \frac{\|b - Ax\|}{\|b\|} < \varepsilon.$$

In PCG we can still compute η .

For GMRES, using a preconditioner means running GMRES on

Left precondition.	Right precondition.	Split precondition.
$MAx = Mb$	$AMy = b$	$M_2AM_1y = M_2b$

The free estimate of the residual norm of GMRES is associated with the preconditioned system.

Preconditioner location v.s. stopping criterion (cont)

	Left precondition.	Right precondition.	Split precondition.
$\eta_M(A, b)$	$\frac{\ MAx - Mb\ }{\ MA\ \ x\ + \ Mb\ }$	$\frac{\ AMy - b\ }{\ AM\ \ x\ + \ b\ }$	$\frac{\ M_2AM_1y - M_2b\ }{\ M_2AM_1\ \ y\ + \ M_2b\ }$
$\eta_M(b)$	$\frac{\ MAx - Mb\ }{\ Mb\ }$	$\frac{\ AMy - b\ }{\ b\ }$	$\frac{\ M_2AM_1y - M_2b\ }{\ M_2b\ }$

Using $\eta_M(b)$ for right preconditioned linear system will monitor the convergence as if no preconditioner was used.

Some classical algebraic preconditioners

- ▶ Incomplete factorization : IC , $ILU(p)$, $ILU(p, \tau)$
- ▶ SPAI (Sparse Approximate Inverse): compute the sparse approximate inverse by minimizing the Frobenius norm $\|MA - I\|_F$
- ▶ FSAI (Factorized Sparse Approximate inverse): compute the sparse approximate inverse of the Cholesky factor by minimizing the Frobenius norm $\|I - GL\|_F$
- ▶ AINV (Approximate Inverse): compute the sparse approximate inverse of the LDU or LDL^T factors using an incomplete biconjugation process

Outline

Reliability of the calculation








Algorithm selection

Why searching solutions in Krylov subspaces

Unsymmetric Krylov solvers based on the Arnoldi procedure

Algebraic preconditioning techniques

Bibliography

-  M. Benzi, *Preconditioning Techniques for Large Linear Systems: A Survey*, Journal of Computational Physics, 182, pp. 418-477, 2002.
-  A. Greenbaum, *Iterative methods for solving linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
-  J. Liesen and Z. Strakoš, *Krylov Subspace Methods: Principles and analysis*, Oxford science publications, 2013.
-  G. Meurant, *Computer solution of large linear systems*, vol. 28 of Studies in Mathematics and its Applications, North-Holland Publishing Co., Amsterdam, 1999.
-  Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Second edition, 2003.
-  V. Simoncini and D. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14, pp. 1-59, 2007.
-  H. A. van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge monographs on applied and computation mathematics, Cambridge University Press, Cambridge, 2003.