Projection iterative methods of Krylov's type

for solving linear systems

Part I : Theoretical Study

Jean-Marie Chesneaux

Pierre et Marie Curie University, Paris

Contents

Basements of Krylov type methods

An interesting Web site :

http://www-history.mcs.st-and.ac.uk/ history/Mathematicians

If x_0 is a vector and A a matrix of dimension n, the Krylov subspace $\mathcal{K}_m(A, x_0)$ is defined by

$$\mathcal{K}_m(A, x_0) = span\left\{x_0, Ax_0, ..., A^{m-1}x_0\right\}.$$

If $y \in \mathcal{K}_m(A, x_0), \exists$ a polynomial q of degree less than $m-1$ such that $y = q(A)x_0$.

Proposition 1 *If r is the degree of the minimal polynomial of x, the dimension of $\mathcal{K}_m(A, x)$ is exactly m if and only if $m \leq r$ and $\mathcal{K}_m(A, x) = \mathcal{K}_r(A, x)$ if $m \geq r$.*

From the Cayley-Hamilton theorem, $r \leq n$.

Projection methods

Let be $Ax = b$ a general linear system.

If K_k and L_k are two subspaces of dimension m_k and x_0 a given vector, an approximation x_k can be defined by the two conditions

1. $x_k - x_0 \in K_k$,
2. $r_k = b - Ax_k \perp L_k$.

Condition 2 is the Petrov-Galerkin condition.

Let be U_k and V_k two bases of K_k and L_k. If $W_k = AU_k$, x_k exists if there is a vector $a \in \mathbb{R}^{m_k}$ such that $x_k - x_0 = U_k a$ or $r_k = r_0 - W_k a$ and the Petrov-Galerkin condition leads to

$$V_k^T r_k = V_k^T r_0 - V_k^T W_k a = 0.$$

Krylov type projection method

K_k and L_k are Krylov subspaces $\Longrightarrow$ iterative methods.

$K_k = \mathcal{K}_{m_k}(A, r_0)$ and $L_k = \mathcal{K}_{m_k}(A^T, y) \Longrightarrow$ Lanczos, CGS, BiCGStab and QMR methods.

$K_k = L_k = \mathcal{K}_{m_k}(A, x_0) \Longrightarrow$ FOM algorithm.

$K_k = \mathcal{K}_{m_k}(A, x_0)$ and $L_k = A\mathcal{K}_{m_k}(A, x_0) \Longrightarrow$ GMRES algorithm.

Theses methods may be matrix free, lower cost of computations for high dimension. Because computations are essentially matrix-vector products, they can easily be transposed on parallel computers.

Contents

Lanczos method

Let be x_0 and y two given vectors. The Lanczos method define a sequence x_k as following

1. $x_k - x_0 \in \mathcal{K}_k(A, r_0) = span\left\{r_0, Ar_0, ..., A^{k-1}r_0\right\}$,
2. $r_k = b - Ax_k \perp \mathcal{K}_k(A^T, y)$.

where $r_0 = b - ax_0$. The first condition means that
$x_k - x_0 = -a_1 r0 - ... - a_k A^{k-1} r_0$ and $r_k = r_0 + a_1 Ar_0 + ... + a_k A^k r_0$.
The Petrov-Galerkin conditions are $((A^T)^i y, r_k) = (y, A^i r_k) = 0$ for
$i = 1, ..., k-1$. The a_i's are the solution of a linear system.

Theorem 1 *If all the linear systems are regular, $\exists k \leq n$ such that $x_k = x_s$.*

Proposition 2 *All the systems are regular if A is a symmetric positive definite matrix and the Lanczos method is similar to the Conjugate Gradient method.*

Formal orthogonal polynomial approach

If we set $P_k(\varepsilon) = 1 + a_1^{(k)}\varepsilon + \cdots + a_k^{(k)}\varepsilon^k$ then we have $r_k = P_k(A)r_0$.

Moreover, if we define the linear functional c on the space of polynomials by $c(\varepsilon^i) = (y, A^i r_0), \quad i = 0, 1, \cdots$, then the orthogonality conditions can be written in the form

$$c(\varepsilon^i P_k) = 0 \quad \text{for } i = 0, \cdots, k-1.$$

P_k is the orthogonal polynomial of degree at most k belonging to the family of formal orthogonal polynomials with respect to c such that $P_k(0) = 1$.

This approach is due to C. Brezinski.

The existence and uniqueness of P_k is determined by the non null value of the following Henkel determinant.

$$H_k^{(1)} = \begin{vmatrix} (y, Ar_0) & (y, A^2 r_0) & \cdots & (y, A^k r_0) \\ (y, A^2 r_0) & (y, A^3 r_0) & \cdots & (y, A^{k+1} r_0) \\ \vdots & \vdots & & \vdots \\ (y, A^k r_0) & (y, A^{k+1} r_0) & \cdots & (y, A^{2k-1} r_0) \end{vmatrix} \neq 0.$$

We assume that P_k exists and that its degree is k.

The P_k's family satisfy a three term recurrence relationship

$$P_{k+1}(\varepsilon) = (A_{k+1}\varepsilon + B_{k+1})P_k(\varepsilon) - C_{k+1}P_{k-1}(\varepsilon).$$

with $P_0(\varepsilon) = 1$ and $P_{-1}(\varepsilon) = 0$.

The A_{k+1}, B_{k+1} and C_{k+1} verify

$$\begin{aligned} A_{k+1}c(\varepsilon U_{k-1}P_k) - D_{k+1}c(U_{k-1}P_{k-1}) &= 0 \\ A_{k+1}c(\varepsilon U_k P_k) + B_{k-1}c(U_k P_k) - D_{k+1}c(U_k P_{k-1}) &= 0 \end{aligned}$$

where U_k is an arbitrary polynomial of degree k.

Then

$$r_{k+1}(\varepsilon) = (A_{k+1}A + B_{k+1})r_k - C_{k+1}r_{k-1}.$$

with $r_0 = b - Ax_0$ and $r_{-1} = 0$.

Different choices for U_k leads to different implementation of the Lanczos method.

For the following, $U_k = P_k$ which leads to the Lanczos/Orthomin implementation or *Bi-Conjugate Gradient method* (BCG).

Let be $P_k^{(1)}$ the regular monic polynomial of degree n_k belonging to the family of formal orthogonal polynomials with respect to the functional $c^{(1)}$ defined by $c^{(1)}(\zeta^i) = c(\zeta^{i+1})$.

The existence and uniqueness condition is the same that the one for P_k, that is, $H_k^{(1)} \neq 0$.

Define $Q_k(\varepsilon) = (-1)^k \begin{vmatrix} (y, Ar_0) & \cdots & (y, A^{k-1}r_0) \\ \vdots & & \vdots \\ (y, A^{k-1}r_0) & \cdots & (y, A^{2k-2}r_0) \end{vmatrix} P_k^{(1)}(\varepsilon)$, it can be shown that

$$P_{k+1}(\varepsilon) = P_k(\varepsilon) - \beta_k \varepsilon Q_k(\varepsilon) \qquad \text{with} \quad \beta_k = c(P_k^2)/c(\varepsilon P_k Q_k)$$
$$Q_{k+1}(\varepsilon) = P_{k+1}(\varepsilon) + \alpha_k Q_k(\varepsilon) \qquad \alpha_k = -c(\varepsilon P_k P_{k+1})/c(\varepsilon P_k Q_k(\varepsilon)).$$

$$c(\varepsilon P_k Q_k) = (y, AQ_k(A)r_k) = (P_k(A^T)y, AQ_k(A)r_0).$$

Define $p_k = Q_k(A)r_0$, $\overline{r}_k = P_k(A^T)y$ and $\overline{p}_k = Q_k(A^T)y$, we have the following algorithm

$$\beta_k = (\overline{r}_k, r_k)/(\overline{p}_k, Ar_k)$$
$$r_{k+1} = r_k - \beta_k A p_k$$
$$x_{k+1} = x_k + \beta_k p_k$$
$$\overline{r}_{k+1} = \overline{r}_k - \beta_k A^T \overline{p}_k$$
$$\alpha_k = (\overline{r}_{k+1}, r_{k+1})/(\overline{r}_k, r_k)$$
$$p_{k+1} = r_{k+1} + \alpha_k p_k$$
$$\overline{p}_{k+1} = \overline{r}_{k+1} + \alpha_k \overline{p}_k$$

On the convergence of the method

We only study the case where A is symmetric positive definite matrix.

Therefore x_k minimizes the A-norm of the error e_k in the affine subspace $x_0 + \mathcal{K}_k(A, r_0) = x_0 + q(A)r_0$ where q is a polynomial of degree $\leq k-1$ and $e_k = P_k(A)e_0$..

If $A = UDU^T$, $A^{1/2} = UD^{1/2}u^T$ is also a symmetric positive definite matrix and

$$\|e_k\| \leq \min_{d(P)=k, P(0)=1} \|P(D)\| \, . \, \|e_0\|_A \, .$$

The polynomial which minimizes the maximum on [λ_1, λ_n] is given by

$$P_n(\varepsilon) = \frac{T_n\left(\frac{2\varepsilon-\lambda_n-\lambda_1}{\lambda_n-\lambda_1}\right)}{T_n\left(\frac{-\lambda_n-\lambda_1}{\lambda_n-\lambda_1}\right)}$$

where T_n is the Chebyshev polynomial of first kind.

Define $\kappa = \frac{\lambda_n}{\lambda_1}$, we obtain

Theorem 2 *For the Conjugate Gradient method,*

$$\begin{aligned} \frac{\|e_k\|_A}{\|e_0\|_A} &\leq 2\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k\right]^{-1} \\ &\leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \approx 2\left(1-\frac{2}{\sqrt{\kappa}}\right)^k \end{aligned}$$

The convergence is linear.

Breakdowns and near breakdowns

For a family of formal orthogonal polynomials, some polynomial may not *exist*.

In the lanczos method, it leads to divisions by zero.

The problem is

1. to recognize the occurrence of a breakdown,
2. to determine the degree of the next existing *regular* polynomial,
3. to compute this polynomial.

This has be solved by A. Draux in the case of monic orthogonal polynomials like $P_k^{(1)}$. If $P_k^{(1)}$ is the kth regular polynomial of degree n_k, the degree of the next regular is given by $n_{k+1} = n_k + m_k$ where

$$c^{(1)}\left(\varepsilon^i P_k^{(1)}\right) \begin{cases} = 0, i = 0, ..., n_k + m_k - 2, \\ \neq 0, i = n_k + m_k - 1 \end{cases}$$

The new recurrence relationship becomes

$$P_{k+1}^{(1)}(\varepsilon) = w_k(\varepsilon) P_k^{(1)}(\varepsilon) - \gamma_{k+1} P_{k-1}^{(1)}(\varepsilon)$$

where w_k is a monic polynomial of degree m_k.

A similar relationship occurs for the P_k's with an other polynomial v_k and the coefficients of w_k and v_k are given by solving a (small) linear system.

Contents

Avoiding the transpose - the CGS algorithm

From P. Sonneveld

Remember the formula

$$c(\varepsilon P_k Q_k) = (y, AQ_k(A)r_k) = (P_k(A^T)y, AQ_k(A)r_0)$$

which involves the expression $P_k(A)r_k = P_k^2(A)r_0$.

In the CGS algorithm the new residual is given by $r_k = P_k^2(A)r_0$.

The relation

$$P_{k+1}(\varepsilon) = P_k(\varepsilon) - \beta_k \varepsilon Q_k(\varepsilon)$$
$$Q_{k+1}(\varepsilon) = P_{k+1}(\varepsilon) + \alpha_k Q_k(\varepsilon)$$

are simply squared.

Putting

$$r_k = P_k^2(A)r_0$$
$$q_k = Q_k^2(A)r_0$$
$$v_k = P_{k+1}(A)Q_k(A)r_0$$
$$u_k = P_k(A)Q_k(A)r_0,$$

the new algorithm is

$$\beta_k = (y, r_k)/(y, Au_k)$$
$$v_k = u_k - \beta_k A q_k$$
$$r_{k+1} = r_k - \beta_k A(v_k + u_k)$$
$$\alpha_{k+1} = (y, r_{k+1})/(y, r_k)$$
$$q_{k+1} = r_{k+1} + 2\alpha_k v_k + \alpha_k^2 q_k$$
$$u_{k+1} = r_{k+1} + \alpha_k v_k$$
$$x_{k+1} = x_k + \beta_k(v_k + u_k)$$

Contents

The BiCGStab algorithm

From H.A Van der Vorst.

The CGS allows to avoid the use of the transpose but it amplifies the chaotic behaviour of the residual.

The aim of the BiCGStab algorithm is *to smooth* the convergence behaviour of the algorithm.

The new residual is defined by

$$r_k = W_k(A)P_k(A)r_0$$

with $W_{k+1} = (1 - a_k\varepsilon)W_k(\varepsilon), W_0(\varepsilon) = 1$ and a_k such that $\|r_{k+1}\|$ is minimal.

Because the recurrence between the W_k's is simpler than those between the P_k's, this algorithm is simpler than the CGS algorithm.

Putting

$$r_k = W_k(A)P_k(A)r_0$$
$$p_k = W_k(A)Q_k(A)r_0$$
$$u_k = W_k(A)P_{k+1}(A)r_0,$$

the new algorithm is

$$\beta_k = (y, r_k)/(y, Ap_k)$$
$$u_k = r_k - \beta_k Ap_k$$
$$a_k = (u_k, Au_k)/(Au_k, Au_k)$$
$$r_{k+1} = u_k - a_k Au_k$$
$$\alpha_k = \beta_k(y, r_{k+1})/a_k(y, r_k)$$
$$p_{k+1} = r_{k+1} + \alpha_k(Id - a_k A)p_k$$
$$x_{k+1} = x_k + \beta_k p_k + a_k u_k.$$

Contents

Introduction

The choice for the subspaces are

$K_k = L_k = \mathcal{K}_k(A, x_0)$ for FOM algorithm,

$K_k = \mathcal{K}_k(A, x_0)$ and $L_k = A\mathcal{K}_k(A, x_0)$ for the GMRES method.

There is no more bi-orthogonalization and no more recurrence relationships.

The aim is to minimize at each step a quantity relatively to a given norm.

The FOM method minimizes $(A(x_s - y), x_s - y)$ in $x_0 + K_k$ if A is symmetric positive definite.

The GMRES method minimizes $\|b - Ay\|_2$ in $x_0 + K_k$ if A is regular.

In both cases, the matrix $V_k^T W_k$ is regular and the Petrov-Galerkin condition $V_k^T r_k = V_k^T r_0 - V_k^T W_k a = 0$ can be satisfied whatever the bases for K_k and L_k.

Arnoldi's process

It builds an orthonormal base of a Krylov subspace.

Algorithm:

1. Take a vector v_1 of norm 1
2. For $j = 1, ..., k$ Do
3. For $i = 1, ..., j$ Do $h_{ij} = (Av_j, v_i)$
4. $w_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i$
5. $h_{j+1,j} = \|w_j\|_2$
6. If $h_{j+1,j} = 0$ then stop
7. $v_{j+1} = w_j / h_{j+1,j}$
8. Enddo

The Arnoldi-modified process

Algorithm:

1 Take a vector v_1 of norm 1

2 For $j = 1, ..., k$ Do

3 $w_j = Av_j$

4 For $i = 1, ..., j$ Do

5 $h_{ij} = (w_j, v_i)$

6 $w_j = w_j - h_{ij} v_i$

7 Enddo

8 $h_{j+1,j} = \|w_j\|_2$

9 If $h_{j+1,j} = 0$ then stop

10 $v_{j+1} = w_j / h_{j+1,j}$

11 Enddo

The new version is much more reliable than the old one. To improve the results, one can make a second orthogonalization. Another is superfluous.

If a and b are *quasi*-orthonornormale :

$$(a,a) = 1 + \varepsilon_{a,a},\ (b,b) = 1 + \varepsilon_{b,b},\ (a,b) = \varepsilon_{a,b}.$$

We apply the two process to a new vector v.

$v_1 = v - (v,a)a - (v,b)b$

$v_2 = v - (v,a)a - (v - (v,a)a, b)b$

and

$$\begin{aligned}
(v_1,b) &= -(v,a)\varepsilon_{a,b} - (v,b)\varepsilon_{b,b} \\
(v_2,b) &= (v,b) - (v,a)\varepsilon_{a,b} - \left[(v,b) - (v,a)\varepsilon_{a,b}\right](1 + \varepsilon_{b,b}) \\
&= -(v,a)\varepsilon_{a,b}\varepsilon_{b,b} - (v,b)\varepsilon_{b,b}
\end{aligned}$$

Some properties

Proposition 3 *If V_k is the $n \times k$ matrix with columns vector $v_1, ..., v_k$, $\overline{H}_k$ the $(k+1) \times k$ Hessenberg matrix of the h_{ij}'s and H_k the matrix $\overline{H}_k$ except the last row, we have*

$$AV_k = V_k H_k + w_k e_k^T = V_{k+1} \overline{H}_k \text{ and } V_k^T A V_k = H_k.$$

Proposition 4 *Arnoldi's process stops at step j ($h_{j+1,j} = 0$) if and only if the minimal polynomial of v_1 is if degree j. Then, $AK_j = K_j$.*

The FOM algorithm

Define $r_0 = b - Ax_0, \beta = \|r_O\|$ and $v_1 = r_0/\beta$. If the Arnoldi's process is applied, a matrix V_k is obtained.

The approximation x_k verifies $x_k - x_0 = V_k y_k$ for a vector $y_k \in \mathbb{R}^k$ and $r_k - x_0 = -AV_k y_k$. The Petrov-Galerkin condition is $V_k^T r_k = 0$.

From $V_k^T A V_k = H_k$, it shows that y_k is solution of
$H_k y_k = V_k^T r_0 = \beta V_k^T v_1 = \beta e_1$.

Then, $y_k = H_k^{-1}(\beta e_1)$ and $x_k = x_0 + V_k y_k$.

The stopping criterion must depend on $\|r_k\| = h_{k+1,k} \left| e_k^T y_k \right|$.

The restarted FOM(k)

Algorithm:

1. Compute $r_0 = b - Ax_0, \beta = \|r_O\|_2$ and $v_1 = r_0/\beta$
2. Compute V_k and H_k staring with v_1
3. Compute $y_k = H_k^{-1}(\beta e_1)$ and $x_k = x_0 + V_k y_k$
4. If satisfied then stop
5. Set $x_0 = x_k$ and go to 1

Contents

The GMRES method

Due to Y. Saad and M.H. Schultz . With $L_k = AK_k$, r_k must verify

$$(AV_k)^T r_k = 0 = (AV_k)^T r_0 - (AV_k)^T AV_k y_k.$$

Then

$$y_k = \left[(AV_k)^T AV_k\right]^{-1} (AV_k)^T r_0.$$

Proposition 5 *The GMRES method has a finite convergence.*

The vector y_k minimizes

$$\|r_0 - AV_k y\| = \left\|\beta v_1 - V_{k+1}\overline{H}_k y\right\| = \left\|\beta e_1 - \overline{H}_k y\right\|$$

because of the orthonormality of V_{k+1}.

This is a least-squares problem which can always be solved, for instance with a QR decomposition of $\overline{H}_k$ which leads to a triangular linear system.

Restarted GMRES(k)

Algorithm:

1. Compute $r_0 = b - Ax_0, \beta = \|r_O\|_2$ and $v_1 = r_0/\beta$
2. Compute V_k and H_k staring with v_1
3. Compute y_k which minimizes $\left\|\beta e_1 - \overline{H}_k y\right\|$ and $x_k = x_0 + V_k y_k$
4. If satisfied then stop
5. Set $x_0 = x_k$ and go to 1

On the convergence of GMRES

Proposition 6 *If A is a positive definite matrix ($(Ax, x) > 0,\ \forall x \neq 0$), GMRES(k) converges for any $k \geq 1$.*

Proposition 7 *If A is a diagonalizable matrix, if*

$$\epsilon_k = \min_{P(0)=1} \max_{i=1,..,n} |P(\lambda_i|$$

then $\exists\ C$ such that

$$\|r_k\| \leq C\epsilon_k \|r_0\|$$

If the eigenvalues of A belongs to an ellipse which excludes the origine, $\exists \alpha$ and C_1 such that $C \approx C_1 \alpha^k$

Miscellaneous

Proposition 8 *If the Arnoldi's process stops at step j, the least-squared problem gives the exact solution.*

For large and sparse linear system, a preconditioner is essential.

Variations

1. The use of the Householder orthogonalization instead of the Arnoldi's process.
2. Incomplete or truncated GMRES versions (DQGMRES)
3. Block methods
4. Others ...

Contents

The QMR algorithm

From R.W. Freund and N.M. Nachtigal.

The aim is to apply the GMRES philosophy to the Lanczos method.

From the classical approach, if $V_k = span\left\{r_0, ..., A^{k-1}r_0\right\}$ and $W_k = span\left\{y, ..., (A^T)^{k-1}y\right\}$, $W_k^T A V_k$ is a tridiagonal matrix T_k.

The matrix

$$\overline{T}_k = \begin{pmatrix} T_k \\ \delta_{k+1} e_k^T \end{pmatrix}$$

verify $AV_k = V_{k+1}\overline{T}_k$. Therefore,

$$\|b_A x_k\| = \left\|V_{k+1}\left(\beta e_1 - \overline{T}_k y_k\right)\right\|_2 .$$

y_k is taken to minimize $\left\|\beta e_1 - \overline{T}_k y_k\right\|_2$.

References

C. Lanczos, Solution of systems of linear equations by minimez iterations, J. Res. Natl. Bur. Stand., 49 (1952) 33-53.

Y. Saad, Iterative Methods for Solving Linear Systems, PWS Publ. Co., Boston, 1996.

Breakdowns in the implementation of the Lanczos for solving linear systems, Technical report n320, http://ano.univ-lille1.fr

A review of formal orthogonality in Lanczos-based methods, Technical report n426, http://ano.univ-lille1.fr

Projection iterative methods of Krylov's type

for solving linear systems

Part II : From the computer point of view

Jean-Marie Chesneaux

Pierre et Marie Curie University, Paris

Contents

The floating point arithmetic

For a real number $x \neq 0$,

$$x = \varepsilon.b^e.m$$

with

$$b \in \mathbb{N},\ \varepsilon \in \{-1, +1\},\ e \in \mathrm{Z},\ m \in [1, b).$$

To store x on computer is to store $\{\varepsilon, e, m\}$. Using the base 2:

$$e = \sum_{i=0}^{p} b_i.2^i \text{ and } m = \sum_{i=0}^{\infty} a_i.2^{-i} \text{ with } (a_i, b_i) \in \{0, 1\}$$

1	2 ... 9	10 32
s	e + $2^7 - 1$	a_1 ..... a_{23}

IEEE simple precision implementation

The rounding modes

Let be X_{min} (resp. X_{max}) the smaller (resp. the biggest) floating point number:

$$\forall x \in (X_{min}, X_{max}), \exists \left\{X^-, X^+\right\} \in \mathbb{IF}$$

such that

$$X^- < x < X^+ \text{and } \left(X^-, X^+\right) \cap \mathbb{IF} \neq \text{O}$$

To define the rule which, from x, gives X^- or X^+ is choosing the **rounding mode**.

The 4 rounding modes

The IEEE norm includes 4 rounding modes :

- *rounding to the zero* : X is the floating point number which is the closest to x between x and 0,
- *rounding to the nearest* : X is the floating point number which is the closest to x,
- *rounding to plus infinity* : $X = X^+$.
- *rounding to minus infinity* : $X = X^-$.

Contents

Who is right ?

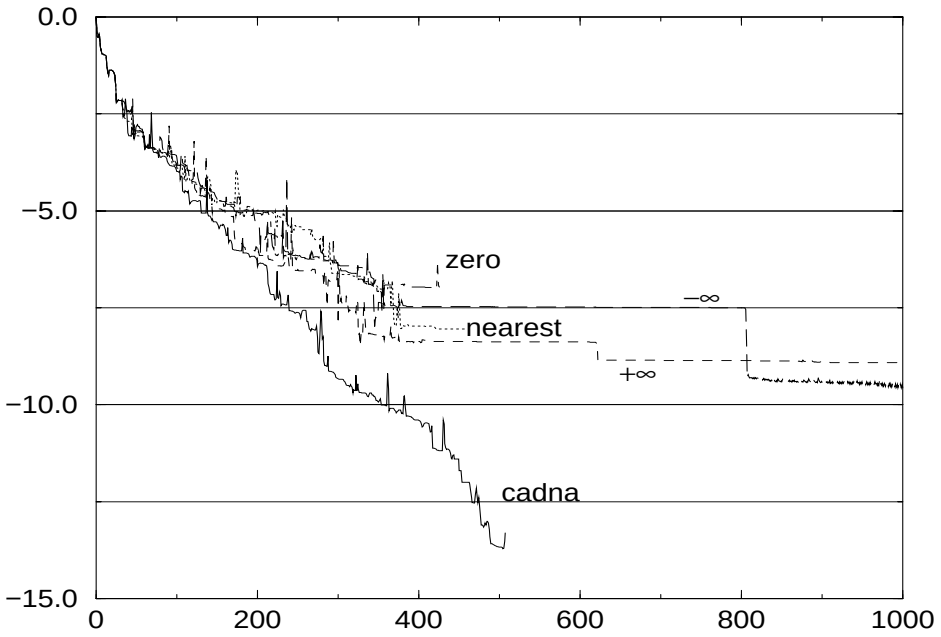


Figure 1: $log(\frac{\|r_k\|_\infty}{\|r_0\|_\infty})$ *versus the number of iterations*

BiCGStab on a matrix of order 2395 with 13151 non-zero elements (electronic circuit conception). Conditioning number is 1.76e+04.

Consequences of the finite arithmetic

1. unknown accuracy
2. Chaotic behaviour of convergence
3. Inefficient stopping criteria
4. The theoretical property are not yet satisfied : orthogonalization ...
5. Fuzzy dynamic control of the execution for restarted strategy.

To control the round-off error propagation

1. **Backward analysis**
2. Direct analysis
3. Interval arithmetic
4. **Stochastic arithmetic**

Direct analysis and interval arithmetic are not efficient for large scale computations in linear algebra.

Contents

Backward analysis

James Hardy Wilkinson (1919 - 1986)

The computed result is considered as an exact result of the exact algorithm on different data.

$y + \Delta_y = f\left(x + \Delta_x\right)$

Δ_y is the forward error, Δ_x is the backward error.

If x^* is the compute result, we define

$$\eta(x^*) = \min_{\Delta_y} \left\{ \|\Delta_y\| : y + \Delta_y = f\left(x^*\right) \right\}.$$

A backward analysis gives formulae like

$$\|\Delta_x\| \leq K(f, y)\eta(x^*).$$

Some results

For one implementation of the CG it has been proved that

$$\lim_{k\to\infty} \sup \frac{\|r_k\|_2}{\|x_k\|_2} \leq \mathbf{u}\kappa \|A\| C.$$

It has been proved that the behaviour of the CG in finite precision is the behaviour of the CG in exact arithmetic applied on a matrix of higher dimension.

Another result fro the Arnoldi process :

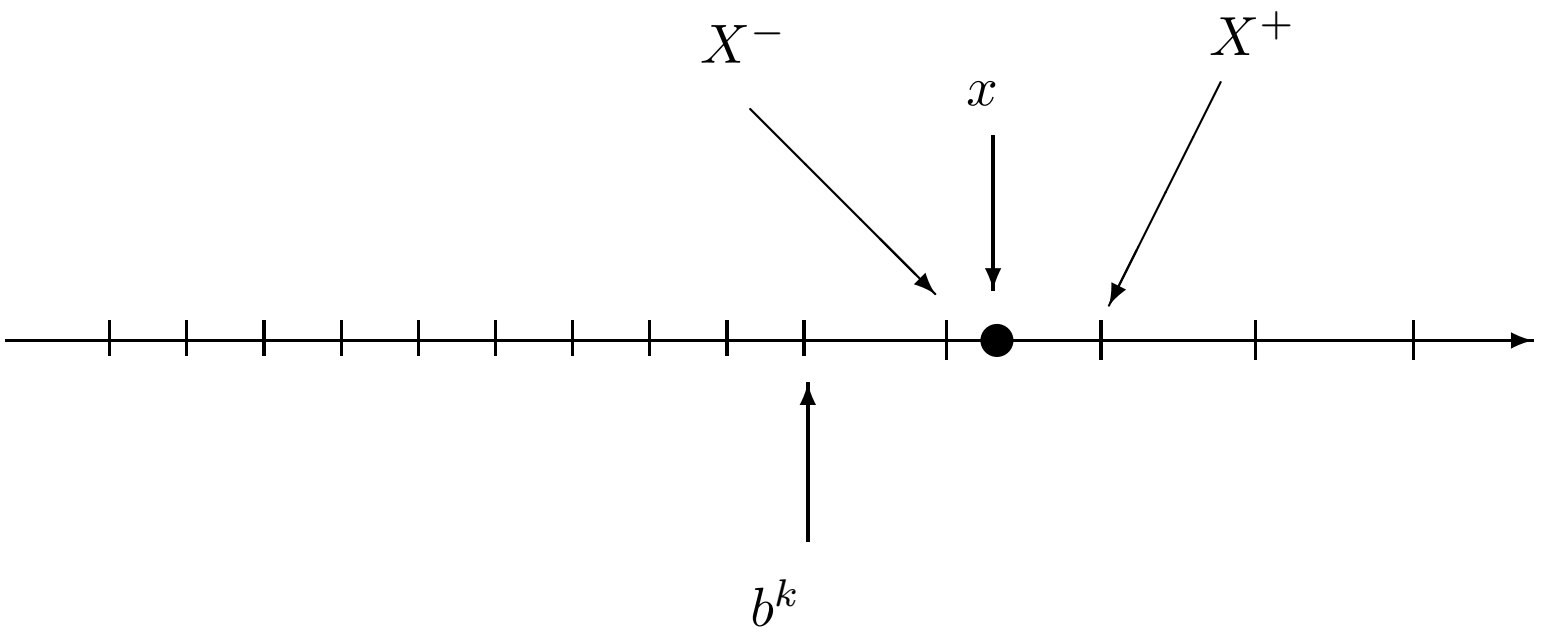
Theorem 3 *Let $E_k = A\overline{V}_k - \overline{V}_k\overline{H}$, then exists a small constant C such that*

$$\|E_k\| \leq C\mathbf{u} \|A\|_2 .$$

It has been proved that, in the GMRES method, the loss of orthogonality is not essential.

Contents

The random rounding mode



X^- ou X^+ with probability1/2.

On the numerical reliability of the rounding mode

$$X = \varepsilon.M.2^E \text{ with } X = x - \varepsilon.2^{E-p}.\alpha$$

- rounding to the nearest, $\alpha \in [-0.5, 0.5[$;
- rounding to the zero, $\alpha \in [0, 1[$;
- rounding to plus or minus infinity , $\alpha \in]-1, +1[$;
- random rounding, $\alpha \in]-1, +1[$

All these rounding modes are exact and, in practice numerically equivalent.

Concept of exact significant digits

Definition 1 *Let be a and b two real numbers, the number of significant digits in common between a and b may be defined as*

1. *for* $a \neq b$, $C_{a,b} = \log_{10} \left| \dfrac{a+b}{2.(a-b)} \right|$,
2. $\forall a \in \mathbb{R}$, $C_{a,a} = +\infty$.

Remark: if $|a-b| \ll |a+b|$, one can take $C_{a,b} \approx \log_{10} \left| \dfrac{a}{a-b} \right|$.

The number of exact significant digits of a computed result X is $C_{X,x}$ where x is the mathematical result.

The CESTAC method

Jean Vignes et Michel La Porte (1972)

- Performing N times the code using the random rounding mode to get N different results R_i.
- Taking as computed result : $\overline{R} = \frac{1}{N} . \sum_{i=1}^{N} R_i.$
- An estimation of the number of exact significant digits is given by :

$$C_{\overline{R}} = \log_{10} \left(\frac{\sqrt{N}. \left| \overline{R} \right|}{s.\tau_\beta} \right) \quad \text{with} \quad s^2 = \frac{1}{N-1} . \sum_{i=1}^{N} \left(R_i - \overline{R} \right)^2 ,$$

$N = 2$ **or** 3**,** $\beta = 0.95$ **and** $\tau_\beta = 12,706$ **or** $4,303$**.**

Few words on the theory

R is modelized by:

$$Z = r + \sum_{i=1}^{n} g_i(d).2^{-p} z_i \ ,$$

z_i's are iid uniform random variables on $[-1, +1]$.

1 - The expectation of Z is r

2 - The distribution of Z is quasi-gaussian

The formula of $C_{\overline{R}}$ is obtained by applying the Student's test on Z.

Few samples are only needed because the approximation do not need to be accurate.

Discret Stochastic Arithmetic

The new order relations are based on the concept of the **computed zero**.

A computed zero is a sample R_i's such as

a) $\forall i, R_i = 0,$

or

b) $C_{\overline{R}} <= 0.$

Therefore, two samples are equal if their substraction is a computed zero (noted $@.0$) . This relation is used for all the order relations.

From the computer point of vue, two results are equal if they cannot be distinguished because of the round-off errors.

The CADNA software implements automatically the DSA. Samples are called **stochastic numbers**.

Improvement of stopping criteria

$if\ \ (\|r_k\| \leq \varepsilon)\ \ then$

ε too small $\Longrightarrow$ infinite loop

ε too big $\Longrightarrow$ inaccurate approximation

A good choice : $\|r_k\|$ insignificant.

This is optimal from the computer point of view.

Idem for

$$if\ \ (\|x_k - x_{k-1}\| \leq \varepsilon)\ \ then \Longrightarrow if\ \ (x_k = x_{k-1})\ \ then.$$

Possibility of new strategies for numerical algorithms

Contents

For the BiCGStab

When must we jump ?

When do we restart ?

When do we stop ?

$$\beta_k = (y, r_k)/(y, Ap_k)$$
$$u_k = r_k - \beta_k A p_k$$
$$a_k = (u_k, Au_k)/(Au_k, Au_k)$$
$$r_{k+1} = u_k - a_k A u_k$$
$$\alpha_k = \beta_k (y, r_{k+1})/a_k (y, r_k)$$
$$p_{k+1} = r_{k+1} + \alpha_k (Id - a_k A) p_k$$
$$x_{k+1} = x_k + \beta_k p_k + a_k u_k.$$

Example 1

The dimension of the system is 1000 :

$$A = \begin{pmatrix} a & 1 & & & \\ -1 & a & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & a & 1 \\ & & & -1 & a \end{pmatrix} \quad x = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} \quad x_0 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad b = \begin{pmatrix} a+1 \\ a \\ \vdots \\ a \\ a-1 \end{pmatrix}$$

with $a = 0.5$, $y = b - Ax_0$ and for floating-point arithmetic, $\varepsilon = 10^{-15}$.

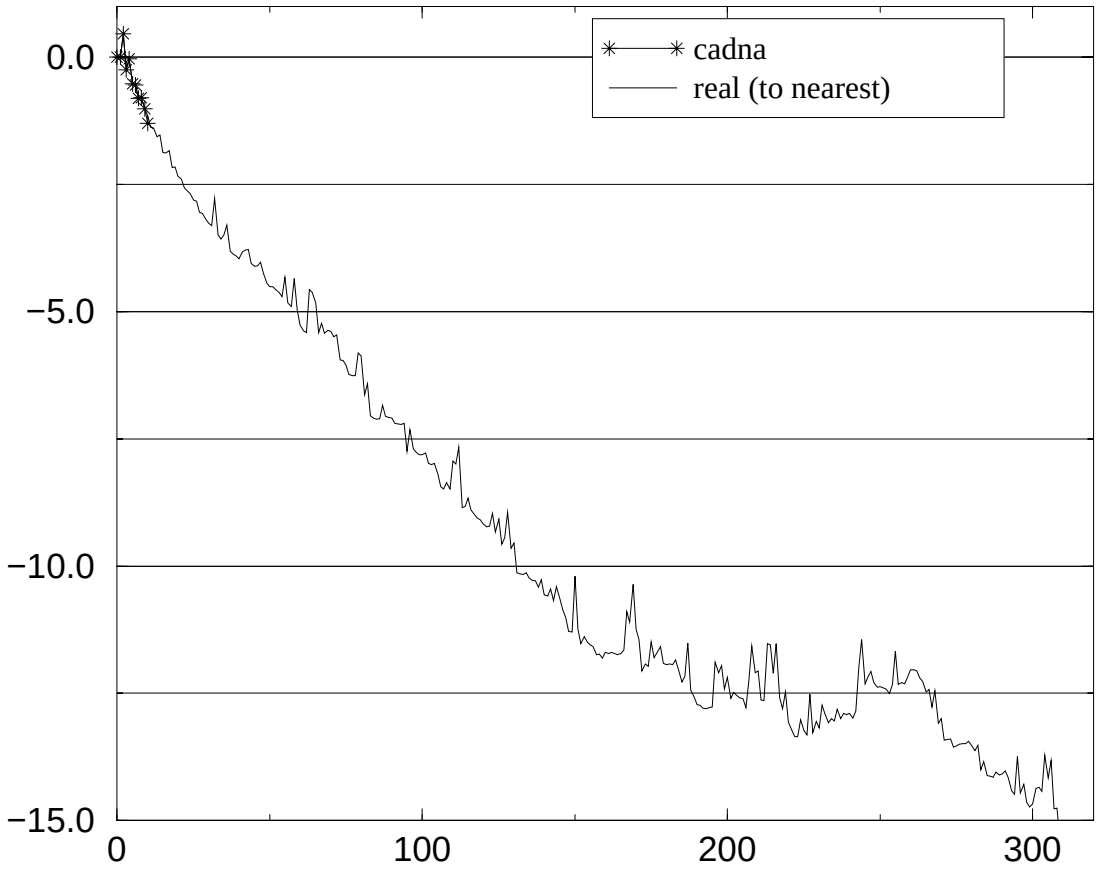


Figure 2: $log(\frac{\|r_k\|_\infty}{\|r_0\|_\infty})$ *versus the number of iterations in* BICGSTAB

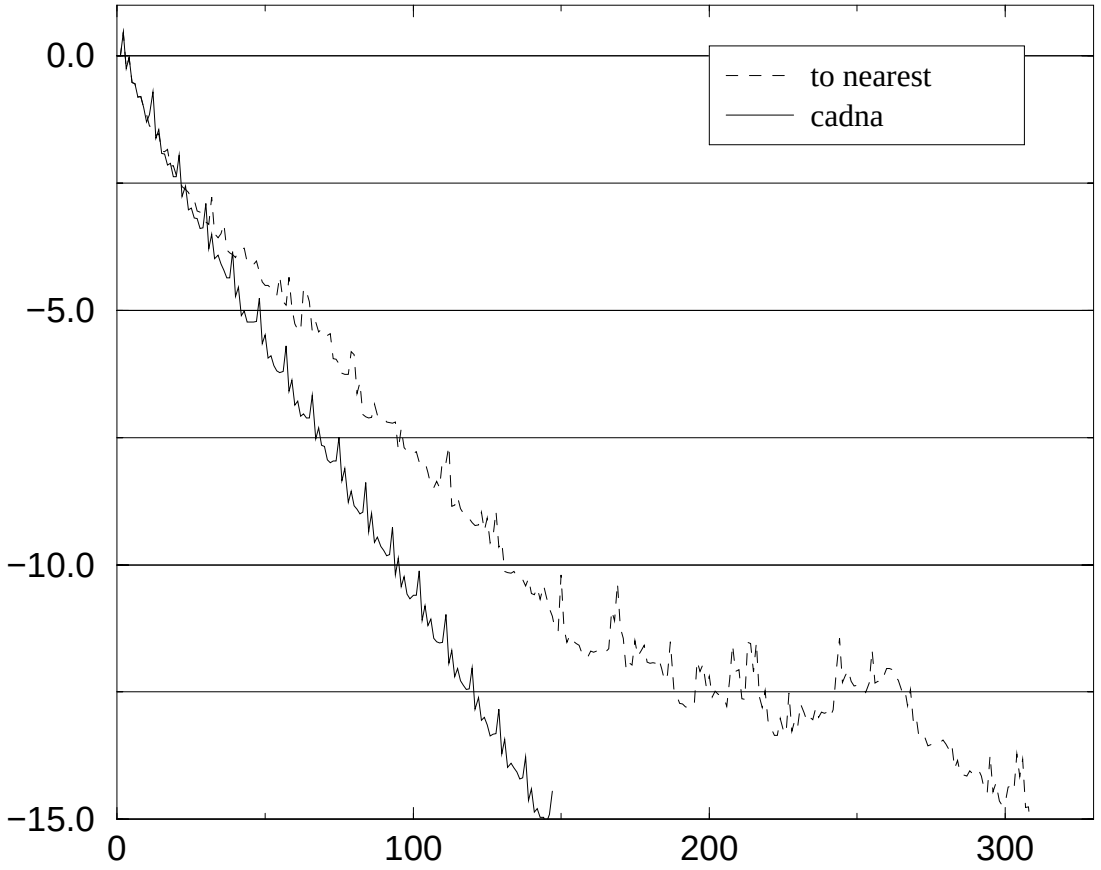


Figure 3: $log(\frac{\|r_k\|_\infty}{\|r_0\|_\infty})$ *versus the number of iterations in* BICGSTAB

Who is right ?

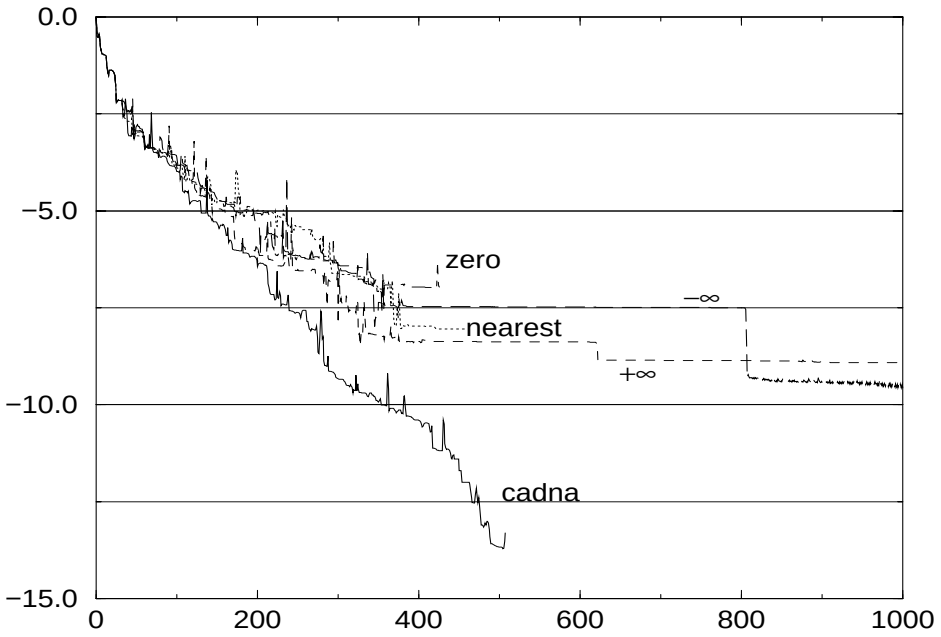


Figure 4: $log(\frac{\|r_k\|_\infty}{\|r_0\|_\infty})$ *versus the number of iterations*

BiCGStab on a matrix of order 2395 with 13151 non-zero elements (electronic circuit conception). Conditioning number is 1.76e+04.

References

F. Chaitin-Chatelin and V. Fraysse, Lecture on finite precision computations, SIAM, Philadelphia, 1996

N. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, Philadelphia, 1996

A. Greenbaum and Z. Strakos, Predicting the behaviour of finite precision Lanczos and Conjugate Gradient computations, Siam J. Matrix Anal. Appl., vol. 13, 1 (1992) pp 121-137.

A. Greenbaum, M. Rozloznik and Z. Strakos, Numerical stability of GMRES, BIT 37 (3) (1997) 706-719.

J.-M. Chesneaux : L'arithmétique stochastique et le logiciel CADNA, Habilitation à diriger des recherches, Université P. et M. Curie (1995).

M. Montagnac and J.-M. Chesneaux, Dynamical control of a BiCGStab, Applied Num. Math. vol. 32 (2000) 103-117

J. Vignes, A stochastic arithmetic for reliable scientific computation, Math. Comp. Simul., 35, 1993, pp. 233-261.