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

1. **Basements of Krylov type and Lanczos type methods**

2. The formal orthogonal polynomial approach of the Lanczos type methods

3. The CGS algorithm

4. The BiCGStab algorithm

5. The FOM algorithm

6. The GMRES algorithm

7. The QMR algorithm
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 \(K_m(A, x_0)\) is defined by

\[
K_m(A, x_0) = \text{span} \{ x_0, Ax_0, ..., A^{m-1}x_0 \}.
\]

If \(y \in K_m(A, x_0)\), \(\exists\) 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 \(K_m(A, x)\) is exactly \(m\) if and only if \(m \leq r\) and \(K_m(A, x) = K_r(A, x)\) if \(m \geq r\).

From the Cayley-Hamilton theorem, \(r \leq n\).
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 \( \Rightarrow \) iterative methods.

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

\[ K_k = L_k = \mathcal{K}_{m_k} (A, x_0) \] \( \Rightarrow \) FOM algorithm.

\[ K_k = \mathcal{K}_{m_k} (A, x_0) \] and \( L_k = AK_{m_k} (A, x_0) \) \( \Rightarrow \) 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.

Jean-Marie Chesneaux - UPMC
Contents

1. Basements of Krylov type and Lanczos type methods

2. The formal orthogonal polynomial approach of the Lanczos type methods

3. The CGS algorithm

4. The BiCGStab algorithm

5. The FOM algorithm

6. The GMRES algorithm

7. The QMR algorithm
Lanczos method

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

1. $x_k - x_0 \in \mathcal{K}_k(A, r_0) = \text{span} \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \}$,

2. $r_k = b - Ax_k \perp \mathcal{K}_k(A^T, y)$.

where $r_0 = b - ax_0$. The first condition means that $x_k - x_0 = -a_1r_0 - \ldots - a_kA^{k-1}r_0$ and $r_k = r_0 + a_1Ar_0 + \ldots + a_kA^{k}r_0$.

The Petrov-Galerkin conditions are $( (A^T)^i y, r_k ) = ( y, A^i r_k ) = 0$ for $i = 1, \ldots, k - 1$. 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.
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
$c(\varepsilon^i) = (y, A^i r_0), \ i = 0, 1, \cdots$, then the orthogonality conditions can be written in the form
$c(\varepsilon^i P_k) = 0 \ \text{for} \ i = 0, \cdots, k - 1$.

$P_k$ 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.

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

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

$$A_{k+1}c(\varepsilon U_{k-1}P_{k}) - D_{k+1}c(U_{k-1}P_{k-1}) = 0$$
$$A_{k+1}c(\varepsilon U_{k}P_{k}) + B_{k-1}c(U_{k}P_{k}) - D_{k+1}c(U_{k}P_{k-1}) = 0$$

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 $Q_k(\varepsilon) = (-1)^k$

\[
\begin{vmatrix}
(y, A r_0) & \cdots & (y, A^{k-1} r_0) \\
\vdots & \ddots & \vdots \\
(y, A^{k-1} r_0) & \cdots & (y, A^{2k-2} r_0)
\end{vmatrix}
\]

be shown that

\[
P_{k+1}(\varepsilon) = P_k(\varepsilon) - \beta_k \varepsilon Q_k(\varepsilon) \quad \text{with} \quad \beta_k = \frac{c(P_k^2)}{c(\varepsilon P_k Q_k)}
\]

\[
Q_{k+1}(\varepsilon) = P_{k+1}(\varepsilon) + \alpha_k Q_k(\varepsilon) \quad \alpha_k = -\frac{c(\varepsilon P_k P_{k+1})}{c(\varepsilon P_k Q_k)}.
\]
Projection iterative methods of Krylov type

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

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

\[
\begin{align*}
\beta_k &= (\bar{r}_k, r_k) / (\bar{p}_k, Ar_k) \\
r_{k+1} &= r_k - \beta_k Ap_k \\
x_{k+1} &= x_k + \beta_k p_k \\
\bar{r}_{k+1} &= \bar{r}_k - \beta_k A^T \bar{p}_k \\
\alpha_k &= (\bar{r}_{k+1}, r_{k+1}) / (\bar{r}_k, r_k) \\
p_{k+1} &= r_{k+1} + \alpha_k p_k \\
\bar{p}_{k+1} &= \bar{r}_{k+1} + \alpha_k \bar{p}_k
\end{align*}
\]
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

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

The polynomial which minimizes the maximum on $[\lambda_1, \lambda_n]$ 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,*

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

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

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

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 another polynomial $v_k$ and the coefficients of $w_k$ and $v_k$ are given by solving a (small) linear system.
Contents

1. Basements of Krylov type and Lanczos type methods
2. The formal orthogonal polynomial approach of the Lanczos type methods
3. The CGS algorithm
4. The BiCGStab algorithm
5. The FOM algorithm
6. The GMRES algorithm
7. The QMR algorithm
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

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

1. Basements of Krylov type and Lanczos type methods
2. The formal orthogonal polynomial approach of the Lanczos type methods
3. The CGS algorithm
4. The BiCGStab algorithm
5. The FOM algorithm
6. The GMRES algorithm
7. The QMR algorithm
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

1. Basements of Krylov type and Lanczos type methods
2. The formal orthogonal polynomial approach of the Lanczos type methods
3. The CGS algorithm
4. The BiCGStab algorithm
5. The FOM algorithm
6. The GMRES algorithm
7. The QMR algorithm
Introduction

The choice for the subspaces are

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

\[ K_k = \mathcal{K}_k(A, x_0) \text{ and } L_k = A\mathcal{K}_k(A, x_0) \] 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 $j = 1, \ldots, k$ Do
   3. For $i = 1, \ldots, j$ 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
Algorithm: The Arnoldi-modified process

1. Take a vector $v_1$ of norm 1
2. For $j = 1, \ldots, k$ Do
   3. $w_j = Av_j$
   4. For $i = 1, \ldots, j$ 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:

\[(a, a) = 1 + \varepsilon_{a,a}, \quad (b, b) = 1 + \varepsilon_{b,b}, \quad (a, b) = \varepsilon_{a,b}.
\]

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

\[(v_1, b) = -(v, a)\varepsilon_{a,b} - (v, b)\varepsilon_{b,b}\]

\[(v_2, b) = (v, b) - (v, a)\varepsilon_{a,b} - [(v, b) - (v, a)\varepsilon_{a,b}] (1 + \varepsilon_{b,b})\]

\[= -(v, a)\varepsilon_{a,b}\varepsilon_{b,b} - (v, b)\varepsilon_{b,b}\]
Some properties

**Proposition 3** If $V_k$ is the $n \times k$ matrix with columns vector $v_1, \ldots, v_k$, $\overline{H}_k$ 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

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

and $V_k^T AV_k = H_k$.

**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 of degree $j$. Then, $A K_j = K_j$.
The FOM algorithm

Define \( r_0 = b - Ax_0, \beta = \|r_0\| \) 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 \( \|r_k\| = h_{k+1,k} \| e_k^T y_k \| \).
The restarted FOM(k)

Algorithm:

1. Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$

2. Compute $V_k$ and $H_k$ starting 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

1. Basements of Krylov type and Lanczos type methods
2. The formal orthogonal polynomial approach of the Lanczos type methods
3. The CGS algorithm
4. The BiCGStab algorithm
5. The FOM algorithm
6. The GMRES algorithm
7. The QMR algorithm
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

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

Proposition 5 The GMRES method has a finite convergence.

The vector $y_k$ minimizes

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

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 $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$

2. Compute $V_k$ and $H_k$ staring with $v_1$

3. Compute $y_k$ which minimizes $\|\beta e_1 - H_k y\|$ 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 ($\langle Ax, x \rangle > 0$, $\forall x \neq 0$), GMRES($k$) converges for any $k \geq 1$.

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

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

then $\exists C$ such that

$$
\|r_k\| \leq C \varepsilon_k \|r_0\|
$$

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

1. Basements of Krylov type and Lanczos type methods
2. The formal orthogonal polynomial approach of the Lanczos type methods
3. The CGS algorithm
4. The BiCGStab algorithm
5. The FOM algorithm
6. The GMRES algorithm
7. The QMR algorithm
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 $V_k = \text{span} \{ r_0, ..., A^{k-1}r_0 \}$ and $W_k = \text{span} \{ y, ..., (A^T)^{k-1}y \}$, $W_k^T AV_k$ 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,

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

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

Jean-Marie Chesneaux - UPMC
References


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

1. The IEEE finite precision arithmetic
2. Introduction
3. Backward analysis
4. The probabilistic approach : the CESTAC method
5. Application to Krylov type and Lanczos type methods
6. References
For a real number $x \neq 0$, 

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

with

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

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

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

<table>
<thead>
<tr>
<th>s</th>
<th>e + $2^7 - 1$</th>
<th>$a_1$</th>
<th>..........</th>
<th>$a_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>9</td>
<td>10</td>
<td>..........</td>
</tr>
</tbody>
</table>

IEEE simple precision implementation
The rounding modes

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

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

such that

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

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

1. The IEEE finite precision arithmetic

2. Introduction

3. Backward analysis

4. The probabilistic approach : the CESTAC method

5. Application to Krylov type and Lanczos type methods

6. References
Who is right?

Figure 1: $\log\left(\frac{\|r_k\|_\infty}{\|r_0\|_\infty}\right)$ 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

1. The IEEE finite precision arithmetic

2. Introduction

3. Backward analysis

4. The probabilistic approach : the CESTAC method

5. Application to Krylov type and Lanczos type methods

6. References
Backward analysis

James Hardy Wilkinson (1919 - 1986)

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

\[ y + \Delta_y = f(x + \Delta_x) \]

\( \Delta_y \) is the forward error, \( \Delta_x \) is the backward error.

If \( x^* \) is the compute result, we define

\[ \eta(x^*) = \min_{\Delta_y} \{ \| \Delta_y \| : y + \Delta_y = f(x^*) \} . \]

A backward analysis gives formulae like

\[ \| \Delta_x \| \leq K(f, y) \eta(x^*) . \]
For one implementation of the CG it has been proved that

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

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 from the Arnoldi process:

**Theorem 3** Let $E_k = AV_k - V_k A^H$, then exists a small constant $C$ such that

$$\|E_k\| \leq C u \|A\|_2.$$

It has been proved that, in the GMRES method, the loss of orthogonality is not essential.
1. The IEEE finite precision arithmetic

2. Introduction

3. Backward analysis

4. The probabilistic approach : the CESTAC method

5. Application to Krylov type and Lanczos type methods

6. References
The random rounding mode

\[ X^- \quad \text{ou} \quad X^+ \quad \text{with probability} \quad \frac{1}{2}. \]
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 $a \neq b$, $C_{a,b} = \log_{10} \left| \frac{a + b}{2.(a - b)} \right|$,  
2. $\forall a \in \mathbb{R}$, $C_{a,a} = +\infty$.

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

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: $R = \frac{1}{N} \cdot \sum_{i=1}^{N} R_i$.

- An estimation of the number of exact significant digits is given by:

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

$N = 2 \ \text{or} \ 3, \ \beta = 0.95 \ \text{and} \ \tau^\beta = 12,706 \ \text{or} \ 4,303.$
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 $C_{\frac{R}{r}}$ is obtained by applying the Student's test on $Z$.

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

Jean-Marie Chesneaux - UPMC
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) $\frac{C}{R} \leq 0.$

Therefore, two samples are equal if their subtraction is a computed zero (noted $\oplus .0$). 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

\[ \text{if } (\|r_k\| \leq \varepsilon) \text{ then} \]

\(\varepsilon\) too small \(\implies\) infinite loop
\(\varepsilon\) too big \(\implies\) inaccurate approximation

A good choice: \(\|r_k\|\) insignificant.

This is optimal from the computer point of view.

Idem for

\[ \text{if } (\|x_k - x_{k-1}\| \leq \varepsilon) \text{ then } \implies \text{if } (x_k = x_{k-1}) \text{ then}. \]

Possibility of new strategies for numerical algorithms
Contents

1. The IEEE finite precision arithmetic
2. Introduction
3. Backward analysis
4. The probabilistic approach: the CESTAC method
5. Application to Krylov type and Lanczos type methods
6. References
For the BiCGStab

When must we jump?

When do we restart?

When do we stop?

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

The dimension of the system is 1000:

\[
A = \begin{pmatrix}
a & 1 \\
-1 & a & 1 \\
\vdots & \vdots & \vdots \\
-1 & a & 1 \\
-1 & a &
\end{pmatrix}
\]

\[
x = \begin{pmatrix}
1 \\
1 \\
\vdots \\
1
\end{pmatrix}
\]

\[
x_0 = \begin{pmatrix}
0 \\
0 \\
\vdots \\
0
\end{pmatrix}
\]

\[
b = \begin{pmatrix}
a + 1 \\
a \\
\vdots \\
a - 1
\end{pmatrix}
\]

with \(a = 0.5\), \(y = b - Ax_0\) and for floating-point arithmetic, \(\varepsilon = 10^{-15}\).
Figure 2: $\log\left( \frac{\|r_k\|_\infty}{\|r_0\|_\infty} \right)$ versus the number of iterations in BICGSTAB
Figure 3: $\log\left(\frac{\|r_k\|_\infty}{\|r_0\|_\infty}\right)$ versus the number of iterations in BICGSTAB
Figure 4: $\log\left( \frac{\|r_k\|_\infty}{\|r_0\|_\infty} \right)$ versus the number of iterations

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

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


