École Internationale des Sciences du Traitement de l'Information

Analyse Numérique

Cours & Travaux Dirigés
3 avril 2013

ING1 Groupe B
Babillon Damien

2012 2013

Encadré par Chrysostome Baskiotis & Jean-Paul Forest, cb@eisti.eu & jpf@eisti.eu

Table des matières

1 TD premier **2**

2 Exo 1 **2**

3 système d'équations linéaires **3**

4 Algèbre linéaire **4**
4.1 Normes matricielles Exercice 4.1 4
4.2 Normes matricielles Exercice 4.2 4
4.3 Conditionnement d'une matrice Exercice 4.7 5
4.4 Conditionnement d'une matrice Exercice 4.8 5
4.5 Perturbations de A et de b Exercice 4.10 5

5 Résolution de systèmes linéaires **6**

6 résolution de systèmes d'équations linéaires (méthode itérative) **6**

7 Inversion de matrices **7**
7.1 Chapitre 6 exo supp 7
7.2 Chapitre 6 exo supp 9

Jean-Paul Forest tg305 de quoi on parle a quoi sa sert ?

- analyse des erreurs lors des calculs avec des machines vrai dans le passé
- calculs exactes mapple existe. Alors pourquoi calculs erreurs existent encore ?
- l'analyse numérique est indispensable parce que la pluspart des problèmes n'ont pas de solution analytique, ne sont pas suceptiblent de donner une formule magique.

- ex 1 : racine des polynomes $\frac{-b+-\sqrt{b^2-4ac}}{2a}$
  il n'est pas possible dans le cas général de trouver les racines d'un polynome de degrès 5 ou plus.
- ex 2 : équations différentielles (à peu près toutes)
- ex 3 : systèmes linéaires avec des millers de variables

donc résolution numérique, on fait des aproximations parce qu'on a pas le choix.

si $\mathbb{R}$ est divisé en intervals de $\frac{1}{2}$
si $fl(\sqrt{2}) = 1 alors fl(\sqrt{2} \times \sqrt{2}) = 1$
si $fl(\sqrt{2}) = \frac{3}{2} alors fl(\sqrt{2} \times \sqrt{2}) = \frac{9}{4} = \frac{5}{2}$

on résonne en $x + -\epsilon_1, y + -\epsilon_2$
déjà vu en physique. 20˚C $19,5 \leqslant x \leqslant 20.5$ nombres de chiffres significatifs.

L'analyse numérique, c'est les mathématiques des barres d'erreur. $\odot$

$a^2 + b^2 = (a+b)(a-b)$ incertitude sur a et b quelles est la boite la plus petite sur le résultat.

	signe	exposant	mantisse	chiffres significatifs
32 bits	1	8	23	7
64 bits	1	11	52	16

l'addition en représentation finie N'est PAS associative.

---

1 TD premier

2 Exo 1

1) Deux possibilités pour le signe. Base β, p fois pour la mantisse, avec un signe compris entre E_m et E_M, plus le zéro, mais moins deux cas particuliers. Nous avons donc :

$$\text{Signe : } 2$$
$$\text{Mantisse : } \beta^p$$
$$\text{Exposant : } E_M - E_m + 1 - 2$$
$$\text{Total : } Mantisse \times Exposant$$

Si nous appliquons :

$$\text{Signe} = 2$$
$$\text{Mantisse} = 10^3 = 1000$$
$$\text{Exposant} = 16 - (-15) - 1 = 30$$
$$\text{Total} = 2 \times 1000 \times 30 = 60000$$

2) (1) : Prenons x + y, soit : $0.125 \times 10^6 + 0.437 \times 10^{12}$.

Si on additionne les deux nombres : $0.437000125 \times 10^{12}$. La mantisse étant sur 3 places, nous devons tronquer à 0.437×10^{12}...

(2) : $x \times y = 0.125 \times 10^6 \times 0.437 \times 10^{12} = unTruc \times 10^{18}$
Nous avons un overflow...

(3) : z / y = ... $\times 10^{-22}$ – Underflow...

3) Nous faisons attention, car notre mantisse est toujours de 3 places. Il faut arrondir correctement !

(1) : $(x+y)+z = (0.4 \times 10^0 + 0.4 \times 10^0) + 0.1 \times 10^3) = ... = 0.101 \times 10^3$
(2) : $x + (y+z) = ... + (... + ...) = ... = 0.100 \times 10^3$

Conclusion : l'addition en représentation finie non associative !

Algèbre linéaire numérique

3 système d'équations linéaires

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$Ax = b,\ x \in \mathbb{R},\ n = 3$
idée transformer A.

$$\begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a'_{33} \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \end{bmatrix}$$

Idée $A = U.L$ avec U et L matrices triangulaire supérieur et inférieur.
Idée éliminer a_{12}eta_{13} pour obtenir le début d'une matrice triangulaire supérieur.
Méthode directe : Complexité $O(n^3)$. Idée utiliser une méthode indrecte pour améliorer al complexité et pouvoir controler l'erreur.

Décomposition LU $M^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{a_{21}^{(1)}}{a_{11}^{(1)}} & 1 & 0 \\ -\frac{a_{31}^{(1)}}{a_{11}^{(1)}} & 0 & 1 \end{bmatrix}$

$A^{(2)} = M^{(1)}A^{(1)}$
$A^{(3)} = M^{(2)}A^{(2)}$

$$[M^{(1)}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{a_{21}^{(1)}}{a_{11}^{(1)}} & 1 & 0 \\ \frac{a_{31}^{(1)}}{a_{11}^{(1)}} & 0 & 1 \end{bmatrix}$$

$[M^{(m)}]^{-1} = L^m$
$A^{(3)} = L^{(1)}L^{(2)}A^{(1)}$ (inversion de matrice $AX = I$) Complexité $O(n^3)$.
Si A symétrique $\Rightarrow A = L.D.L^T$
A définie positive $\forall \bar{x} \neq 0,\ x^T Ax > 0 : A = (L\sqrt{D})(L\sqrt{D})^T$ décomposition de Cholesky.

---

4 Algèbre linéaire

4.1 Normes matricielles Exercice 4.1

Calculer les normes 1, 2 et ∞ du vecteur $x = [1, 0, 1, -4]^T$.
$|u_1 - l| \leqslant k|u_0 - l|$
$|u_2 - l| \leqslant k|u_1 - l|$
...
$|u_n - l| \leqslant k|u_{n-1} - l|$

$|u_n - l| \leqslant k^n|u_0 - l|$

norme 1 $||v||_1 = |1| + |0| + |1| + |-4| = 6$

norme 2 $||v||_1 = \sqrt{|1|^2 + |0|^2 + |1|^2 + |-4|^2} = \sqrt{18} = 3\sqrt{2}$

norme ∞ $||v||_1 = \max(|1|, |0|, |1|, |-4|) = 4$

4.2 Normes matricielles Exercice 4.2

Calculer les normes 1, 2 et ∞ du vecteur $x = [1, 0, 1, -4]^T$.
Calculer les normes 1, 2 et ∞ de la matrice

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 3 & 2 \\ 1 & 1 & 2 \end{bmatrix}$$

norme 1 (max colonnes) $N_1 \rightarrow \max(5, 5, 5) = 5$

norme 2 defaut Scilab ; racine carrée de la valeur propre la plus grande (rayon spectral) en module
calcul du polyne caractérique $A - \lambda I = \begin{bmatrix} 2-\lambda & 1 & 1 \\ 2 & 3-\lambda & 2 \\ 1 & 1 & 2-\lambda \end{bmatrix}$ $\Rightarrow$ rayon spectral $= \lambda = 3$.
polynome caractéristique : $25 - 53\lambda + 29\lambda^2 - \lambda^3 = (\lambda - 1)(-\lambda^2 + 28\lambda - 25) \Rightarrow spec(A) = \{1,\ 14 - 3\sqrt{19},\ 14 + 3\sqrt{19}\} \Leftrightarrow p(A) = |14 + 3\sqrt{19}| \Leftrightarrow ||A||_2 = \sqrt{14 + 3\sqrt{19}} \approx 5,2$

norme ∞ defaut Mapple ; (max lignes) $N_\infty \rightarrow \max(4, 7, 4) = 7$

$||Ax|| \leqslant ||A||.||x||$

$x \to Ax \to AAx \to A^3x ... Ax$
$lub(A) = \sup_{x\in\mathbb{R}^n,\ x\neq 0} \frac{||Ax|| \text{module de x après transformation}}{||x|| \text{module de x avant transformation}} = \sup_{||x||=1} ||Ax||$
Car les matrices codent des opérations linéaires, $f(\lambda x) = |\lambda| f(x)$

4.3 Conditionnement d'une matrice Exercice 4.7

$\text{cond}(A) = \kappa(A) = ||A||.||A^{-1}||$ bon cas : une solution : droites qui se coupent mauvais cas : 0 ou ∞ solution : droites parallèles

Montrer que $\text{cond}(A) = \kappa(A) \geqslant 1$.
1 : $\text{lub}(I) = 1 \leqslant \text{lub}(A).\text{lub}(A^{-1}) = ||A||.||A^{-1}||$

4.4 Conditionnement d'une matrice Exercice 4.8

Montrer que $\text{cond}(A^2) \leqslant (\text{cond}(A))^2 \forall A$.
$\text{cond}(A^2) = ||A^2||\ ||(A^2)^{-1}|| \leqslant ||A||.||A||.||(A^2)^{-1}||.||(A^2)^{-1}||$

$$H_2 = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \ldots & \\ \ldots & & & & \end{bmatrix}$$

4.5 Perturbations de A et de b Exercice 4.10

Soit le système $Ax = b$ et supposons que nous avons une solution approchée $\widetilde{x} = x + \Delta x$. Dans ce cas on $Ax = b + \Delta b$.
1. Montrer que $||\Delta x|| \leqslant ||A^{-1}||.||\Delta b||$
$A(x + \delta x) = b + \Delta b$
$Ax + A\delta x = b + \Delta b$
$A\delta x = \Delta b$
Onsuppose que A^{-1} existe
$\Delta x = A^{-1}\Delta b \Rightarrow ||\Delta x|| \leqslant ||A^{-1}||.||\Delta b||$
2. Calculer une majoration de la norme de l'erreur relative $\frac{||\Delta x||}{||x||}$
$b = Ax$ donc
$||b|| = ||Ax|| \leqslant ||A||.||x||$
$||x|| \geqslant \frac{||b||}{||A||}$
via 1. $\frac{||\Delta x||}{||x||} \leqslant ||A||.||A^{-1}|| \frac{\Delta b||}{||b||}$

5 Résolution de systèmes linéaires

temps d'inversion de matrices scilab :

- *tic() ; inv(rand(1000, 1000)) ; toc* = 0.34sec.
- *tic() ; inv(rand(2000, 2000)) ; toc* = 1.35sec.
- *tic() ; inv(rand(4000, 4000)) ; toc* = 8.053sec.
- *tic() ; inv(rand(8000, 8000)) ; toc* = 58.89sec.

Lorsqu'on multiplie par 2 la taille de la matrice le temps est multiplié par 8. Donc complexité $O(n^3)$.

$Ax = b$
$L.U.x = b$, avec L matrice triangulaire inférieur et U matrice triangulaire supérieur. Permet d'utiliser la rétrorestitution.
Pour n^2 cases la complexité ne change pas si on utilise l'inversion standard de matrice. La résolution du l'équation prend autant de temps dans un cas ou dans l'autre.
$Ax = b \rightarrow x = A^{-1}b \equiv O(n^3)$ inversion directe
$Ax = b \rightarrow L.U.x = b \equiv O(n^3) \rightarrow x = U^{-1}.L^{-1}.b \equiv O(n)$ finalement la complexité est $O(n^3)$ par le méthode factorisation.

Donc quel intérêt ?
Le coût de l'inversion "directe" et de la factorisation sont du même ordre de grandeur.(Cependant la seconde méthode permet de faire varier le b sans avoir à recaluler $L.U$.) Par conre la résolution de l'équation pour un second membre fixé est linéaire.

en scilab *[L,U,P]= lu([1, 3 ;2, 5])* permet de récuperer les matrices L et U.
tic() ; [L,U,P]= lu(rand(4000, 4000)) ; toc = 3.847sec.

Objectif écrire notre fonction *lu*.

---

6 résolution de systèmes d'équations linéaires (méthode itérative)

Soit P le problème à résoudre : $P(x),\ x \in \mathbb{R}^n$
Solution du problème $x^* \in \mathbb{R}$

Méthode itérative choix $x_0 \in mathbbR$
Algo : $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$
$x_1 = \Phi(x_0),\ x_2 = \Phi(x_1),\ ...$
$\lim\limits_{k\rightarrow\infty} x_{k+1} = \lim\limits_{k\rightarrow\infty} \Phi(x_k) = x^*$
À la convergence : $\Phi(x_k) = x^* = \Phi(x_{k+1})$

$\forall k > N \Phi(x_k) = x^*$ ou x^* est un point fixe de l'algo Φ

Méthode itérative appliquées à la solution d'un sel $A = L + D + U$ matrices (voir décomposition LU)
$A = M - N$ avec M matrice régulière
$G = M^{-1}N$
...
$e_k = x_{k+1} - x^* = \Phi(x_k) - x^* = M^{-1}Nx_k + M^{-1}b - M^{-1}Nx^* - M^{-1}b = M^{-1}N(x_k - x^*) = M^{-1}Ne_k$ avec e_k l'erreur à la $k - ieme$ itération.
$e_{k+1} = G^k e_0$
$\lim_{k\to\infty} \Phi(x_k) = x^* \Rightarrow \lim G^k = 0 \Leftrightarrow ||G|| < 1$ (convergence)

1 Jacobi $M = D$, $N = -(L + U) = A - M$
$\Phi(x_k) = x_{k+1} = -D^{-1}(1 + U)x_k + D^{-1}b.$

2 Gauss-Seidel $M = L + D$, $N = U$
$\Phi(x_k) = -(L + D)^{-1}Nx_k + (L + D)^{-1}b$

3 Richardson $M = I$, $N = I - A$

Convergence Pour toutes les itérations $||G|| = ||M^{-1}N|| < 1$ Pour Jacobi si A est à diagonale décroissante $|a_{ii}| > \sum |a_{jj}|$
Pour Gauss-Seidel si A symétrique définie positive alors G-S converge
Pour Richardson ...

Relaxation Je me relax... va voir le poly

---

7 Inversion de matrices

Méthodes dierctes : Élimination gaussienne : complexité $O(n^3)$

Méthodes itératives : x_0, x_1, ..., x_k Jacobi, Gauss-Seidel (http ://interstices.info/jcms/i_55887/algorithme-pour-mettre-en-rang-une-equipe-de-football)

7.1 Chapitre 6 exo supp

aide p91 du poly

Exo 1 : $Ax = b$, montrer qu'il existe $c \in \mathbb{R} | ||x_{k+1} - x^*|| \leqslant c||x_k - x^*||$
$x_{k+1} - x^* = -D^{-1}(L+U)x_k + D^{-1}b - x^* = -D^{-1}(L+U)x_k + D^{-1}Ax^* - x^* = -D^{-1}(L+U)x_k + D^{-1}(L+D+U)x^* - x^* = -D^{-1}(L+U)x_k + D^{-1}(L+U)x^* + D^{-1}Dx^* - x^* = -D^{-1}(L+U)(x_k - x^*)$
$||x_{k+1} - x^*|| = || - D^{-1}(L+U)(x_k - x^*)|| \leqslant ||D^{-1}(L+U)||.||x_k - x^*||$
Donc on peut prendre $c = ||D^{-1}(L+U)||$.

mq $||x_k - x^*|| \leqslant \frac{c^k}{1-c}||x_1 - x_0||$
$||x_1 - x^*|| \leqslant c||x_0 - x^*|| = c||x_0 - x_1 + x_1 - x^*|| \leqslant c||x_0 - x_1|| + c||x_1 - x^*||$
$\Leftrightarrow ||x_1 - x^*|| - c||x_1 - x^*|| \leqslant c||x_0 - x_1||$
$\Leftrightarrow c \leqslant 1(1-c)||x_1 - x^*|| \leqslant c||x_0 - x_1||$
$\Leftrightarrow ||x_1 - x^*|| \leqslant \frac{c}{1-c}||x_0 - x_1||$

$||x_2 - x^*|| \leqslant c||x_1 - x^*|| \leqslant \frac{c^2}{1-c}||x_0 - x_1||$
$||x_3 - x^*|| \leqslant c||x_2 - x^*|| \leqslant \frac{c^3}{1-c}||x_0 - x_1||$
...
$||x_k - x^*|| \leqslant c||x_{k-1} - x^*|| \leqslant \frac{c^k}{1-c}||x_0 - x_1||$

On peut donc connaitre a priori le nombre d'itération à faire pour atteindre la précision désirée.

Exo 2 : $C_k = I - B_kA$ mq $(I - C_k)^{-1}$ existe.
$B_kA = I - C_k \Leftrightarrow B_k = (I - C_k)A^{-1} \Leftrightarrow (I - C_k)^{-1}B_k = A^{-1}$. A^{-1} et B_k existe donc $(I - C_k)^{-1}$ existe.
$B_k = (I - C_k)A^{-1}$
$B_k = A^{-1} - C_kA^{-1}$
$||A^{-1} - C_kA^{-1}|| = ||B_k||$
$||A^{-1}|| - ||C_kA^{-1}|| \leqslant ||B_k||$
$||A^{-1}|| - ||C_k||.||A^{-1}|| \leqslant ||B_k||$
$(1 - ||C_k||).||A^{-1}|| \leqslant ||B_k||$
avec $||C_k|| < 1, ||A^{-1}|| \leqslant \frac{||B_k||}{1-||C_k||}$

Exo 3 : $A = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$; a, $b \in \mathbb{R}$. Trouver a, b tel que l'algorithme itératif de Jacobi converge ?
Si $a = 0$ ou $b = 0$ d'après théorème ça converge.
De plus Si A et $2D - A$ sont symétriques définies positives, alors la méthode de Jacobi est convergente.
Sinon la correction : $J = -\begin{bmatrix} 1/a & 0 \\ 0 & 1/a \end{bmatrix} \begin{bmatrix} 0 & b \\ b & 0 \end{bmatrix} = \begin{bmatrix} 0 & -b/a \\ -b/a & 0 \end{bmatrix}$
valeurs propres $\begin{bmatrix} 0 & -b/a \\ -b/a & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \lambda = \pm\frac{b}{a}$ donc $|\frac{b}{a}| < 1$ et $|b| < |a|$.

Montrer que la méthode de Gauss-Seidel sur A converge plus vite que celle

Jacobi.
$M = \begin{bmatrix} a & 0 \\ b & a \end{bmatrix}$
$M^{-1} = \begin{bmatrix} 1/a & 0 \\ -b/a^2 & 1/a \end{bmatrix}$
$G = M^{-1}N = \begin{bmatrix} 0 & b/a \\ 0 & b^2/a^2 \end{bmatrix}$
valeusr propres : $\begin{bmatrix} 0 & b/a \\ 0 & b^2/a^2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
$\lambda \in \{0, \frac{b^2}{a^2}\}$
$P_J = \frac{b}{a}$ et $P_G = \frac{b^2}{a^2}$ donc la 2^{eme} converge plus vite.

7.2 Chapitre 6 exo supp

algorithme du gradient conjugué $x_0 \in \mathbb{R}^n,\ d_0 = r_0 = b - Ax_0$ pour $k = 0$

a $h_k = \frac{d_k^T r_k}{d_k^T A d_k}$

b $x_{k+1} = x_k + h_k d_k$

c $r_{k+1} = r_k - h_k A d_k [= b - Ax_k]$

d $d_{k+1} = r_{k+1} - \frac{d_k^T A r_{k+1}}{d_k^T A d_k} d_k$

Montrer que :

$$\forall x_0 \in \mathbb{R}^n \exists n_0 \in \mathbb{N} \mid \forall n \geqslant n_0\ d_n = 0$$

avec :

1. $r_i^T d_j = 0; 0 \leqslant j \leqslant i \leqslant n_0$
2. $r_i^T r_i > 0; 0 \leqslant i \leqslant n_0$
3. $r_i^T d_i = r_i^T r_i; 0 \leqslant i \leqslant n_0$
4. $d_i^T A d_j = 0; 0 \leqslant i < j \leqslant n_0$
5. $r_i^T r_j = 0; 0 \leqslant i < j \leqslant n_0$
6. $r_i = b - Ax_i; 0 \leqslant i \leqslant n_0$

démo par induction : demo $i = 1$ admit $i = k$ demo $i = k + 1$
Soit $i = 1$

1. $r_1^T d_0 = (r_0 - h_0 A d_0)^T d_0 = r_0^T d_0 - h_0 d_0^T A d_0 = r_0^T d_0 - \frac{d_0^T r_0}{d_0^T A d_0}(d_0^T A d_0) = r_0^T d_0 - d_0^T r_0 = 0$

2. $r_1^T r_1 > 0$ Si $r_1^T r_1 = ||r_1||^2 = 0 \Rightarrow r_1 = 0$ et $d_1 = 0 \Rightarrow$ absurde ou $n_0 = 1 \Rightarrow$ convergence.
3. $r_1^T d_1 = r_1^T (r_1 - \frac{d_0^T A r_1}{d_0^T A d_0} d_0) = r_1^T r_1 - \frac{d_0^T A r_1}{d_0^T A d_0} \underbrace{r_1^T d_0}_{=0} = r_1^T r_1$
4. $d_0^T A d_1 = d_0^T A(r_1 - \frac{d_0^T A r_1}{d_0^T A d_0} d_0 = d_0^T A r_1 - \frac{d_0^T A r_1}{d_0^T A d_0} d_0^T A d_0 = 0$
5. $r_0^T r_1 = r_0^T (r_0 - h_0 A d_0) = r_0^T r_0 - (\frac{d_0^T r_0}{d_0^T A d_0}) r_0^T A d_0 = 0$ avec $d_0 = r_0 \Leftrightarrow d_0^T = r_0^T$
6. $b - Ax_1 = b - A(x_0 + h_0 d_0) = b - Ax_0 - h_0 A d_0 = r_1$

Soit que les proposition sont vérifiées à $i = k$. démontrons à $i = k + 1$

1. $r_{k+1}^T d_j = (r_k - h_k A d_k)^T d_j = r_k^T d_j - h_k d_k^T A d_j = r_k^T d_j - \frac{d_k^T r_k}{d_k^T A d_k} (d_j^T A d_j) = 0$ ou bien $r_{k+1}^T d_j = (r_k - h_k A d_k)^T d_j = \underbrace{r_k^T d_j}_{=0(1)} - \underbrace{h_k d_k^T A d_j}_{=0(4)}$
2. $r_{k+1}^T r_{k+1} > 0$ Si $r_{k+1}^T r_{k+1} = ||r_{k+1}||^2 = 0 \Rightarrow r_{k+1} = 0$ et $d_{k+1} = 0 \Rightarrow$ absurde ou $n_0 = 1 \Rightarrow$ convergence.
3. $r_{k+1}^T d_{k+1} = r_{k+1}^T (r_{k+1} - \frac{d_k^T A r_{k+1}}{d_k^T A d_k} d_k) = r_{k+1}^T r_{k+1} - \frac{d_k^T A r_{k+1}}{d_k^T A d_k} \underbrace{r_{k+1}^T d_k}_{=0} = r_{k+1}^T r_{k+1}$
4. $d_j^T A d_{k+1} = d_j^T A(r_{k+1} - \frac{d_k^T A r_{k+1}}{d_k^T A d_k} d_k = d_j^T A r_{k+1} - \frac{d_k^T A r_{k+1}}{d_k^T A d_k} d_k^T A d_k = 0$
5. $r_j^T r_{k+1} = r_j^T (r_k - h_k A d_k) = r_j^T r_k - (\frac{d_k^T r_k}{d_k^T A d_k}) r_j^T A d_k = 0$
6. $b - Ax_k = b - A(x_k + h_k d_k) = b - Ax_k - h_k A d_k = r_{k+1}$