

ANALYSE NUMÉRIQUE

Corrigés des Travaux Pratiques 2024 – 2025

Séance 3

1. Le raisonnement de la page 17 du Chapitre 1 montre qu'un algorithme qui a la stabilité directe doit satisfaire (pour une norme donnée, ici on considère la norme euclidienne $\|\cdot\|_2$)

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq C_1 C_2 \kappa_\epsilon(A) u \approx C_1 C_2 \kappa(A) u, \quad (1)$$

avec \mathbf{x} la solution exacte, $\hat{\mathbf{x}}$ la solution calculée, $u = 1.1 \cdot 10^{-16}$, $C_1, C_2 \geq 1$ petits, et $\epsilon = C_2 u$. Pour les systèmes linéaires le conditionnement est donné par

$$\kappa(A) = \|A^{-1}\|_2 \|A\|_2$$

et peut être estimé avec l'instruction `cond`. Le membre de droite dans (1) (sans facteur $C_1 C_2$ et avec conditionnement estimé via `cond`) pour les trois systèmes vaut :

système	$\kappa(A)$	$u\kappa(A)$
(1)	2.25	$2.5 \cdot 10^{-16}$
(2)	$3.3 \cdot 10^8$	$3.7 \cdot 10^{-8}$
(3)	1.002	$1.1 \cdot 10^{-16}$

2. factorisation LU :

```

fonction [L U] = an_lu(A)
% [L U] = an_lu(A) retourne les matrices triangulaire
% inférieure L et triangulaire supérieure U telles que
%
%           A = LU,
% pour autant que A soit carrée et qu'une telle
% factorisation existe;
m = size(A,1); n = size(A,2);
if(n~=m)
    error('A doit être carrée');
else
    for k = 1:n
        for j = k:n
            U(k,j) = A(k,j);
        end
        L(k,k) = 1;
        for i = k+1:n
            L(i,k) = A(i,k)/A(k,k);
        end
        for i = k+1:n
            for j = k+1:n
                A(i,j) = A(i,j) - L(i,k)*A(k,j);
            end
        end
    end
end

```

```

    end
  end
end % if

```

Le programme pour la résolution du système LU est comme suit :

```

function x = an_sol(A,b)
[L U] = an_lu(A); % A = LU
y = L\b;
x = U\y;

```

Pour les trois systèmes l'erreur relative sur la solution $\frac{\|\hat{x}-x\|}{\|x\|}$ vaut :

- (1) $1.4 \cdot 10^{-16}$
- (2) $1.3 \cdot 10^{-11}$
- (3) $1.3 \cdot 10^{-11}$

En comparant avec les résultats du premier exercice on constate que la précision sur la solution du troisième système n'est pas celle d'une méthode stable (perte de 5 décimales en précision).

3.

```

clear; close;
x = [-1; -1; -1; -1; -1];
A = [1 -1000 0 0 0; ...
     1 1 -1000 0 0; ...
     0 1 1 -1000 0; ...
     0 0 1 1 -1000; ...
     -1000 0 0 1 1];
b = [999; 998; 998; 998; 998];
% factorisation à l'aide de l'instruction lu
% d'Octave, voir help lu
[L U P] = lu(A);
b = P*b; % max n-1=4 permutations des éléments de b
y = L\b;
xc = U\y;
% erreur relative
norm(x-xc)/norm(x) % alors?

```