

Scilab : analyse numérique

APESAM ¹

2 avril 2025

1. <https://www.apesam.fr/>

Table des matières

Méthode de Dichotomie	3
Méthode du pivot de Gauss	4
Méthode de Newton	6
Calcul d'intégrales	8

Nous donnons ici quelques exemples d'utilisation du logiciel Scilab pour résoudre quelques problèmes classiques d'analyse numérique que l'on trouve au niveau de la licence.

Tous les programmes Scilab sont fournis sous la forme d'un fichier source qu'il suffit de charger dans Scilab et d'exécuter. Nous ne proposons pas d'exposés sur ces algorithmes mais nous fournissons un lien qui développe sa présentation.

Méthode de dichotomie

Cliquer sur ce [lien](#) pour une explication de la méthode

Listing 1 – Méthode de dichotomie

```
1 // Algorithme de dichotomie
2 // Recherche de la racine de la fonction f dans l'intervalle [a,b]
3 // La précision est donnée par la valeur de epsilon
4
5 function c=Dicho(f,a,b,epsilon)
6     c=(a+b)/2;
7     while (b-a)/2 >= epsilon
8         c=(a+b)/2;
9         if f(a)*f(c)<0 then
10             b=c;
11         else
12             a=c;
13         end
14     end
15 endfunction
16
17 // Exemple
18 function y=f(x)
19     y=x^5+5*x-2;
20 endfunction
21
22 Dicho(f,0,1,1D-2)
```

[Exécuter le programme avec Scilab](#)

Méthode du pivot de Gauss

Cliquer sur ce lien pour une explication de la méthode

Résolution d'un système linéaire $Ax = b$ avec A matrice carrée : la matrice doit être inversible et les pivots successifs ne doivent pas être nuls.

Listing 2 – Exemple de script Scilab

```
1 // Méthode du pivot de Gauss pour résoudre un système
2 // linéaire Ax=b avec A matrice carrée inversible
3 // Les pivots successifs ne doivent pas être nuls
4
5 function x=Gauss(A,b)
6     n=size(b,"*"); x=b;
7     // algorithme de Gauss
8     for k=1:n-1
9         for l=k+1:n
10            p=A(l,k)/A(k,k);
11            for m=k:n
12                A(l,m)=A(l,m)-A(k,m)*p;
13            end
14            x(l)=x(l)-x(k)*p;
15        end
16    end
17    // méthode de remontée
18    x(n)=x(n)/A(n,n);
19    for i=n-1:-1:1
20        s=0;
21        for j=i+1:n
22            s=s+A(i,j)*x(j);
23        end
24        x(i)=(x(i)-s)/A(i,i);
25    end
26 endfunction
27
28 // Exemple
29 // solution : [-1;1;0]
30 A=[1 2 3;
31    4 5 6;
32    7 8 10];
33 b=[1;1;1];
34
35 Gauss(A,b)
```

[Exécuter le programme avec Scilab](#)

Même méthode mais avec recherche de pivots partiels : la matrice carrée A doit être inversible.

Listing 3 – Exemple de script Scilab

```
1 // Méthode du pivot de Gauss avec recherche de pivots partiels
2 // pour résoudre un système linéaire  $Ax=b$ 
3 // avec  $A$  matrice carrée inversible. Un pivot est considéré
4 // comme nul lorsque sa valeur absolue est inférieure à  $\epsilon$ 
5 function x=Gauss2(A,b,eps)
6     n=size(b,"*"); x=b;
7     for k=1:n-1
8         // cas où le terme diagonal est près de 0
9         // recherche d'un élément non nul dans la colonne
10        if abs(A(k,k))<eps then
11            kk=find(abs(A(k:n,k))>eps);
12            if kk==[] then
13                disp("Matrice non inversible");
14                return;
15            end
16            // échange des lignes k et kk dans A et dans b
17            kk=kk(1);
18            lignek=A(k,:); A(k,:)=A(kk,:); A(kk,:)=lignek;
19            lignek=b(k); b(k)=b(kk); b(kk)=lignek;
20        end
21        // algorithme de Gauss
22        for l=k+1:n
23            p=A(l,k)/A(k,k);
24            for m=k:n
25                A(l,m)=A(l,m)-A(k,m)*p;
26            end
27            x(l)=x(l)-x(k)*p;
28        end
29    end
30    // méthode de remontée
31    if abs(A(n,n))<eps then
32        disp("Matrice non inversible");
33        return;
34    end
35    x(n)=x(n)/A(n,n);
36    for i=n-1:-1:1
37        s=0;
38        for j=i+1:n
39            s=s+A(i,j)*x(j);
40        end
41        x(i)=(x(i)-s)/A(i,i);
42    end
43 endfunction
44 // Exemple
45 A=[0 2 3;
46    4 5 6;
47    7 8 10];
48 b=[1;1;1];
49 Gauss2(A,b,1D-10)
```

Exécuter le programme avec Scilab

Méthode de Newton

Cliquer sur ce [lien](#) pour une explication de la méthode
Arrêt après n itérations.

Listing 4 – Exemple de script Scilab

```
1 // Méthode de Newton pour trouver un zéro de la fonction f
2 // dont la dérivée est la fonction fprim
3 // u0 valeur initiale approchée de la solution
4 // arrêt après n itérations
5
6 function u=Newton1(f,fprim,u0,n)
7     u=u0;
8     for i=1:n
9         fp=fprim(u);
10        if abs(fp)<=%eps then
11            error("La dérivée est nulle")
12        end
13        u=u-f(u)/fp;
14    end
15 endfunction
16
17 // Exemple
18 // fonction y=f(x)
19 function y=f(x)
20     y=x^3-x-1;
21 endfunction
22
23 // fonction dérivée de f : y=f'(x)
24 function y=fprim(x)
25     y=3*x^2-1;
26 endfunction
27
28 // 5 itérations en partant de 1
29 Newton1(f,fprim,1,5)
```

[Exécuter le programme avec Scilab](#)

Test d'arrêt sur la valeur de la fonction.

Listing 5 – Exemple de script Scilab

```
1 // Méthode de Newton pour trouver un zéro de la fonction f
2 // dont la dérivée est la fonction fprim
3 // u0 valeur initiale approchée de la solution
4 // arrêt lorsque f(u) est inférieur ou égal à eps
5
6 function u=Newton2(f,fprim,u0,eps)
7     u=u0;
8     while abs(f(u))>eps then
9         fp=fprim(u);
10        if abs(fp)<=%eps then
11            error("La dérivée est nulle")
12        end
13        u=u-f(u)/fp
14    end
15 endfunction
16
17 // Exemple
18 // fonction y=f(x)
19 function y=f(x)
20     y=x^3-x-1;
21 endfunction
22
23 // fonction dérivée de f : y=f'(x)
24 function y=fprim(x)
25     y=3*x^2-1;
26 endfunction
27
28 // Erreur de 10(-12) en partant de 1
29 Newton2(f,fprim,1,1D-12)
```

Exécuter le programme avec Scilab

Calcul d'intégrales

Cliquer sur ce [lien](#) pour une explication de la méthode
Intégration par la méthode des trapèzes.

Listing 6 – Exemple de script Scilab

```
1 // Intégration par la méthode des trapèzes de la fonction y=f(x)
2 // entre a et b
3 // subdivisions en puissances de 2
4 // arrêt après n itérations
5 // res = les valeurs successives de l'intégrale au cours des itérations
6
7 function res=trapeze(f,a,b,n)
8     res=[(f(a)+f(b))/2];
9     h=b-a;
10    for k=0:n-2
11        h=h/2;
12        s=0;
13        for i=1:2^k
14            s=s+f(a+(2*i-1)*h);
15        end
16        res=[res,res(k+1)/2+h*s];
17    end
18 endfunction
19
20 // Exemple
21 function y=f(x)
22     y=1/x;
23 endfunction
24
25 trapeze(f,1,2,12)
```

[Exécuter le programme avec Scilab](#)

Listing 7 – Exemple de script Scilab

```
1 // Intégration par la méthode des rectangles de la fonction y=f(x)
2 // entre a et b
3 // subdivisions en puissances de 2
4 // arrêt après n itérations
5 // res = les valeurs successives de l'intégrale au cours des itérations
6
7 function res=rectangle(f,a,b,n)
8     res=[f(a)];
9     h=b-a;
10    for k=0:n-2
11        h=h/2;
12        s=0;
13        for i=1:2^k
14            s=s+f(a+(2*i-1)*h);
15        end
16        res=[res,res(k+1)/2+h*s];
17    end
18 endfunction
19
20 // Exemple
21 function y=f(x)
22     y=1/x;
23 endfunction
24
25 rectangle(f,1,2,12)
```

Exécuter le programme avec Scilab

Listing 8 – Exemple de script Scilab

```
1 // Intégration par la méthode de Simpson de la fonction y=f(x)
2 // entre a et b. Arrêt après n itérations
3
4 function res=Simpson(f,a,b,n)
5     h=(b-a)/n;
6     res=(f(a)+f(b))/2;
7     k=1:n-1;
8     res=res+sum(feval(a+k*h,f));
9     k=0:n-1;
10    res=res+2*sum(feval(a+(2*k+1)*h/2,f));
11    res=res*h/3;
12    res*h/3
13 endfunction
14
15 // Exemple
16 function y=f(x)
17     y=1/x;
18 endfunction
19
20 Simpson(f,1,2,8)
```

[Exécuter le programme avec Scilab](#)

Listing 9 – Exemple de script Scilab

```
1 // Intégration par la méthode de Romberg de la fonction
2 // y=f(x) entre a et b. Arrêt après n itérations
3
4 function res=Romberg(f,a,b,n)
5     h=b-a;
6     T(1)=(f(a)+f(b))/2;
7     for i=0:n-2
8         h=h/2; k=1:2^i;
9         T(i+2)=T(i+1)/2+h*sum(feval(a+(2*k-1)*h,f));
10    end
11    for i=1:n-1
12        for k=1:n-i
13            T(k)=(4^i*T(k+1)-T(k))/(4^i-1);
14        end
15    end
16    res=T(1);
17 endfunction
18
19 // Exemple
20 function y=f(x)
21     y=1/x;
22 endfunction
23
24 Romberg(f,1,2,10)
```

Exécuter le programme avec Scilab