# Scilab: численный анализ

APESAM *

28 марта 2025 г.

## Содержание



*https://www.apesam.fr/

Вот несколько примеров того, как Scilab может быть использован для решения проблем классического числового анализа, возникающих в ранние университетские годы.

Все программы Scilab поставляются в виде исходного файла, который нужно просто загрузить в Scilab и запустить. Мы не предлагаем лекции по этим алгоритмам, но даем ссылку, которая расширяет их представление.

# Метод бисекции

Нажмите на эту ссылку для объяснения метода

Листинг 1: Метод бисекции

```
// Алгоритмдихотомии
// Найдитекореньфункции   f наинтервале  [a,b].
// Точностьопределяетсязначением   epsilon

function c=Dicho(f,a,b,epsilon)
    c=(a+b)/2;
    while (b-a)/2 >= epsilon
        c=(a+b)/2;
        if f(a)*f(c)<0 then
            b=c;
        else
            a=c;
        end
    end
endfunction

// Пример
function y=f(x)
    y=x^5+5*x-2;
endfunction

Dicho(f,0,1,1D-2)
```

Запуск программы с помощью Scilab

## Метод Гаусса — Жордана

Нажмите на эту ссылку для объяснения метода

Решение линейной системы $Ax = b$ с $A$ в виде квадратной матрицы: матрица должна быть инвертируемой, а последовательные точки не должны быть равны нулю.

Листинг 2: Метод Гаусса — Жордана 1

```
// МетодповоротаГауссадлярешениясистемы
// линейной Ax=b с A - инвертируемойквадратнойматрицей
// Последовательныеточкиповоротанедолжныбытьравнынулю

function x=Gauss(A,b)
    n=size(b,"*"); x=b;
    // АлгоритмГаусса
    for k=1:n-1
        for l=k+1:n
            p=A(l,k)/A(k,k);
            for m=k:n
                A(l,m)=A(l,m)-A(k,m)*p;
            end
            x(l)=x(l)-x(k)*p;
        end
    end
    // Методповорота
    x(n)=x(n)/A(n,n);
    for i=n-1:-1:1
        s=0;
        for j=i+1:n
            s=s+A(i,j)*x(j);
        end
        x(i)=(x(i)-s)/A(i,i);
    end
endfunction

// Пример
// решение: [-1;1;0]
A=[1 2 3;
   4 5 6;
   7 8 10];
b=[1;1;1];

Gauss(A,b)
```

Запуск программы с помощью Scilab

Тот же метод, но с поиском частичных точек поворота: квадратная матрица $A$ должна быть инвертируемой.

Листинг 3: Метод Гаусса — Жордана 2

```
// методповоротныхточекГауссаспоискомчастичныхповоротных
// точекдлярешениялинейнойсистемы     Ax=b с A ввиде
// инвертируемойквадратнойматрицыПоворот
// считаетсянулевым , еслиегоабсолютноезначениеменьше     eps
function x=Gauss2(A,b,eps)
    n=size(b,"*"); x=b;
    for k=1:n-1
        // еслидиагональныйчленблизокк     0
        // ищемненулевойэлементвстолбце
        if abs(A(k,k))<eps then
            kk=find(abs(A(k:n,k))>eps);
            if kk==[] then
                disp("неинвертируемая␣матрица");
                return;
            end
            // меняемместамистроки   k и kk в A и b
            kk=kk(1);
            lignek=A(k,:); A(k,:)=A(kk,:); A(kk,:)=lignek;
            lignek=b(k); b(k)=b(kk); b(kk)=lignek;
        end
        // АлгоритмГаусса
        for l=k+1:n
            p=A(l,k)/A(k,k);
            for m=k:n
                A(l,m)=A(l,m)-A(k,m)*p;
            end
            x(l)=x(l)-x(k)*p;
        end
    end
    // методвосхождения
    if abs(A(n,n))<eps then
        disp(Матрица« неинвертируема »);
        return;
    end
    x(n)=x(n)/A(n,n);
    for i=n-1:-1:1
        s=0;
        for j=i+1:n
            s=s+A(i,j)*x(j);
        end
        x(i)=(x(i)-s)/A(i,i);
    end
endfunction
// Пример
A=[0 2 3;
   4 5 6;
```

```
   7 8 10];
b=[1;1;1];
Gauss2(A,b,1D-10)
```

Запуск программы с помощью Scilab

# Метод Ньютона

Нажмите на эту ссылку для объяснения метода

Остановитесь после $n$ итераций.

Листинг 4: Метод Ньютона 1

```
// МетодНьютонадлянахождениянуляфункции     f
// производнойкоторойявляетсяфункция    fprim
// u0 приближенноеначальноезначениерешения
// остановкапосле  n итераций

function u=Newton1(f,fprim,u0,n)
    u=u0;
    for i=1:n
        fp=fprim(u);
        if abs(fp)<=%eps then
            error("производная равна нулю")
        end
        u=u-f(u)/fp;
    end
endfunction

// Пример
// функция y=f(x)
function y=f(x)
    y=x^3-x-1;
endfunction

// производнаяфункции  f : y=f'(x)
function y=fprim(x)
    y=3*x^2-1;
endfunction

// 5 итераций, начинаяс  1
Newton1(f,fprim,1,5)
```

Запуск программы с помощью Scilab

Остановка теста на значении функции.
Остановитесь после $n$ итераций.

Листинг 5: Метод Ньютона 2

```
// МетодНьютонадлянахождениянуляфункции      f
// производнойкоторойявляетсяфункция     fprim
// u0 приближенноеначальноезначениерешения
// останавливаемся, когда f(u) меньшеилиравно    eps

function u=Newton2(f,fprim,u0,eps)
    u=u0;
    while abs(f(u))>eps then
        fp=fprim(u);
        if abs(fp)<=%eps then
            error("производная␣равна␣нулю")
        end
        u=u-f(u)/fp
    end
endfunction

// Пример
// функция y=f(x)
function y=f(x)
    y=x^3-x-1;
endfunction

// производнаяфункции   f : y=f'(x)
function y=fprim(x)
    y=3*x^2-1;
endfunction

// Ошибка 10^(-12), начинаяс   1
Newton2(f,fprim,1,1D-12)
```

Запуск программы с помощью Scilab

## Численное интегрирование

Нажмите на эту ссылку для объяснения метода
нтеграция методом трапеций.

Листинг 6: методом трапеций

```
// Интегрированиефункции  y=f(x) потрапеции
// между a и b
// разделитенастепени   2
// останавливаемсяпосле  n итераций
// res = последовательныезначенияинтегралазавремяитераций

function res=trapeze(f,a,b,n)
    res=[(f(a)+f(b))/2];
    h=b-a;
    for k=0:n-2
        h=h/2;
        s=0;
        for i=1:2^k
            s=s+f(a+(2*i-1)*h);
        end
        res=[res,res(k+1)/2+h*s];
    end
endfunction

// Пример
function y=f(x)
    y=1/x;
endfunction

trapeze(f,1,2,12)
```

Запуск программы с помощью Scilab

Интегрирование по методу прямоугольников.

Листинг 7: методом прямоугольников

```
// Intégration par la méthode des rectangles de la fonction y=f
    (x)
//   entre a et b
// subdivisions en puissances de 2
// arrêt après n itérations
// res = les valeurs successives de l'intégrale au cours des
    itérations

function res=rectangle(f,a,b,n)
    res=[f(a)];
    h=b-a;
    for k=0:n-2
        h=h/2;
        s=0;
        for i=1:2^k
            s=s+f(a+(2*i-1)*h);
        end
        res=[res,res(k+1)/2+h*s];
    end
endfunction

// Пример
function y=f(x)
    y=1/x;
endfunction

rectangle(f,1,2,12)
```

Запуск программы с помощью Scilab

Метод Симпсона.

Листинг 8: методом Симпсона.

```
// интегрированиеСимпсонафункции   y=f(x) между a и b
// остановкапосле  n итераций

function res=Simpson(f,a,b,n)
    h=(b-a)/n;
    res=(f(a)+f(b))/2;
    k=1:n-1;
    res=res+sum(feval(a+k*h,f));
    k=0:n-1;
    res=res+2*sum(feval(a+(2*k+1)*h/2,f));
    res=res*h/3;
    res*h/3
endfunction

// Пример
function y=f(x)
    y=1/x;
endfunction

Simpson(f,1,2,8)
```

Запуск программы с помощью Scilab

Метод Ромберга.

Листинг 9: методом Ромберга.

```
// интегрированиепоРомбергуфункции   y=f(x) между a и b
// остановкапосле  n итераций

function res=Romberg(f,a,b,n)
    h=b-a;
    T(1)=(f(a)+f(b))/2;
    for i=0:n-2
        h=h/2; k=1:2^i;
        T(i+2)=T(i+1)/2+h*sum(feval(a+(2*k-1)*h,f));
    end
    for i=1:n-1
        for k=1:n-i
            T(k)=(4^i*T(k+1)-T(k))/(4^i-1);
        end
    end
    res=T(1);
endfunction

// Пример
function y=f(x)
    y=1/x;
endfunction

Romberg(f,1,2,10)
```

Запуск программы с помощью Scilab