Отчет по Д/з №3 по теме

«Метод простой итерации»

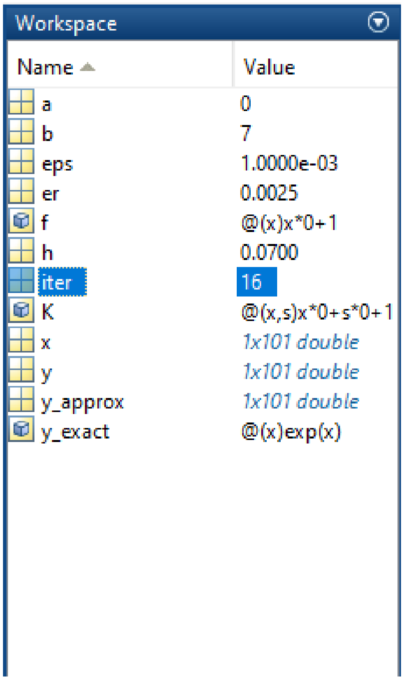
Выполнил

студент 3 курса 09-711 гр.

Нуриев Ильназ

Упр. 1. На каком шаге итерационного процесса была достигнута требуемая точность?

Ответ: Требуемая точность была достигнута на 16 шаге:



Упр. 2. Исследуйте, как зависит от шага сетки h точность вычислений и скорость сходимости итерационного процесса.

Решение: Заменим в коде, взятом из учебника, h — шаг сетки, er — относительная ошибка, iter — количество итераций на вектора, чтобы построить графики зависимости от h.

```
clear all
close all
clc
f = @(x) x*0 + 1;
K = @(x,s) x*0 + s*0 + 1;
a = 0;
b = 7;
n=100;
er=zeros(n,1);
i=zeros(n,1);
h=linspace(0.05,1);
eps = 1e-03;
y_exact = @(x) exp(x);
for j=1:n
x = a : h(j) : b;
[y_approx,iter] = IterVolt(x,h(j),eps,f,K);
i(j)=iter;
```

```
    y=y_exact(x);
    er(j) = norm(y-y_approx,inf)/norm(y,inf);
end
plot(h,er);
xlabel('h'); ylabel('er')
figure;
plot(h,i);
xlabel('h'); ylabel('iter')
```

График зависимости относительной ошибки от шага сетки:

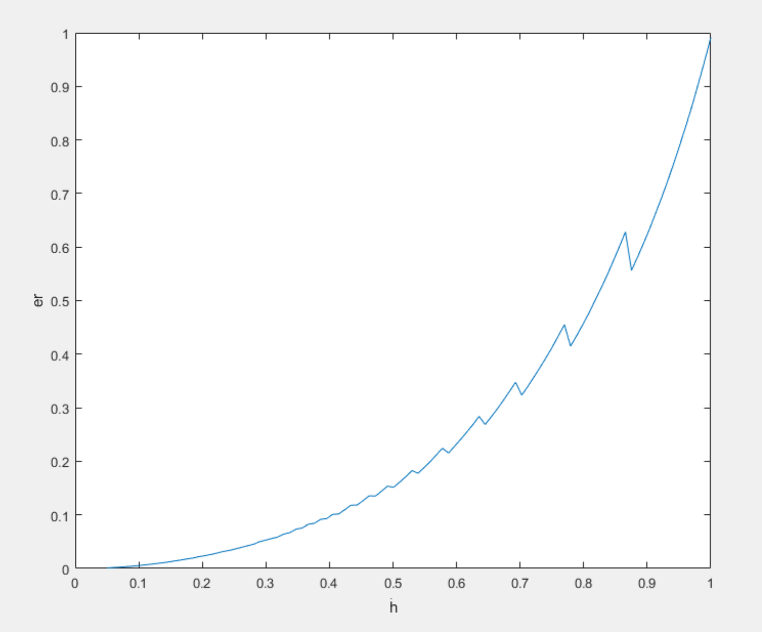
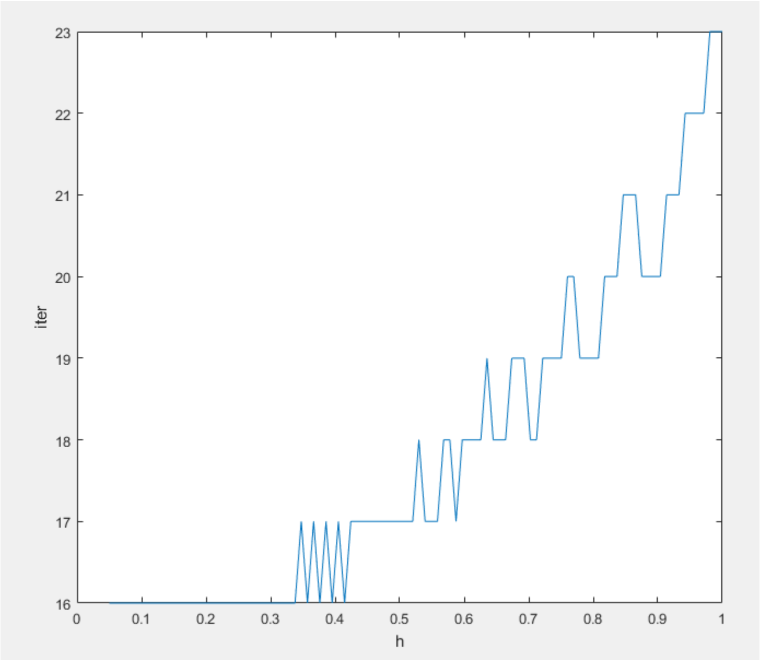


График зависимости количества итераций от шага сетки:



Значит при увеличении шага сетки увеличивается относительная ошибка (то есть точность уменьшается), а количество итераций, в основном, увеличивается (скорость схождения уменьшается).

Упр. 3. С помощью функции Iter_Volt.m найдите приближенное решение уравнения

$$y(x) = x - \int_0^x (x-s)y(s)ds, \quad x \in [0, 2\pi].$$

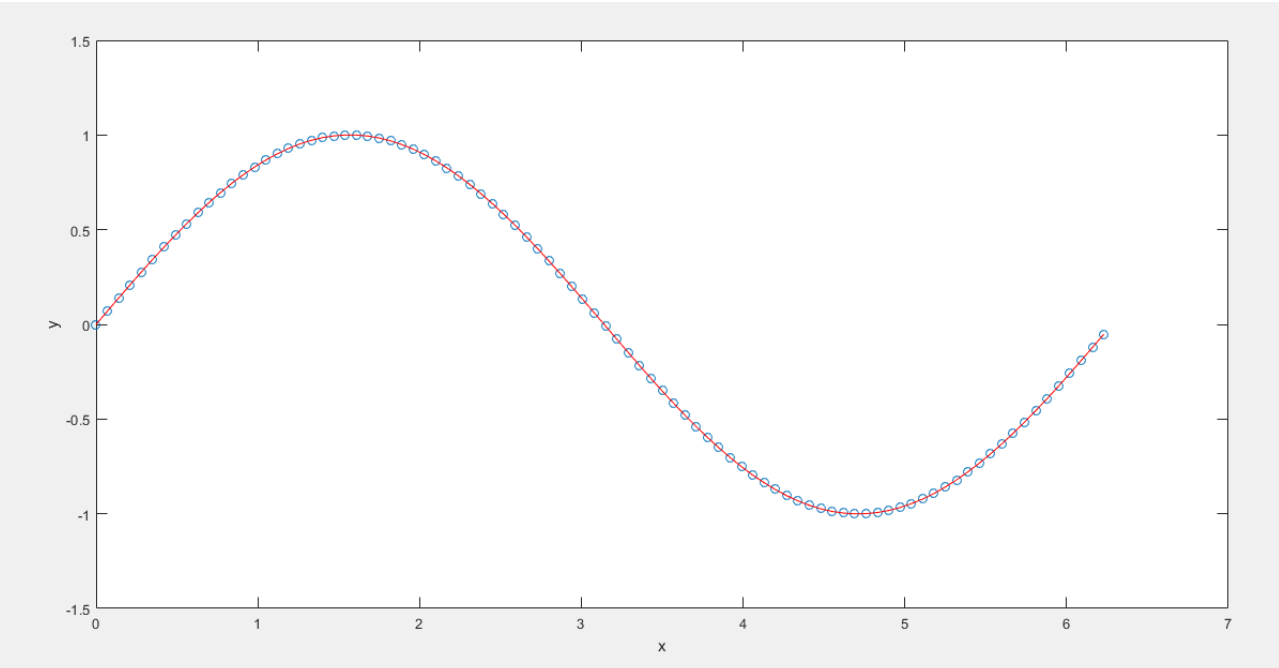
Его точное решение y(x)=sin(x).

Решение:

```
clear all
close all
clc
f = @(x) x;
K = @(x,s) s - x;
a = 0;
b = 2*pi;
h = 0.07;

eps = 1e-03;
y_exact = @(x) sin(x);
x = a : h : b;
[y_approx,iter] = IterVolt(x,h,eps,f,K);
disp(iter)
y=y_exact(x);
plot(x,y,'o',x,y_approx,'r');
er = norm(y-y_approx,inf)/norm(y,inf);
xlabel('x');
ylabel('y');
```

График результатов приближенного решения при h=0.7 и eps=0.001:



Упр. 5. Напишите функцию, предназначенную для решения методом простой итерации нелинейных уравнений Вольтерра второго рода. Найдите приближенное решение уравнения

$$y(x) = \int_0^x \frac{1 + y^2(s)}{1 + s^2} ds, \quad x \in [0, 10].$$

Точное решение y(x) = x.

Решение:

Nonlin_CalcInt.m

```
%Внесем везде y(x) внутрь ядра K (аналогично сделаем в
Nonlin_Iter_Volt.m и zdnie5.m)

function [yk] = CalcInt(y,h,x,n,K,f)
yk = y;
for i = 1 : n
yk(i) = 0;
for j = 1 : i
yk(i) = yk(i) + 2*K(x(i),x(j),y(j));
end
yk(i) = yk(i) - K(x(i),x(1),y(1)) - K(x(i),x(i),y(i));
yk(i) = f(x(i)) + yk(i)*h/2;
end
end
```

Nonlin_Iter_Volt.m

```
function [y,iter] = Nonlin_Iter_Volt(x,h,eps,f,K)
n = numel(x);
y=f(x);
yk = Nonlin_CalcInt(y,h,x,n,K,f);
iter = 0;
while norm(yk-y,inf)/norm(yk,inf) > eps
y = yk;
yk = Nonlin_CalcInt(y,h,x,n,K,f);
iter = iter + 1;
end
end
```

zdnie5.m

```
clear all
close all
clc
f = @(x) x*0;
K = @(x,s,y) (1+y*y)/(1+s*s);
a = 0;
b = 8;
```

```
h = 0.01;
eps = 1e-03;
y_exact = @(x) x;
x = a : h : b;
[y_approx,iter] = Nonlin_Iter_Volt(x,h,eps,f,K);
disp(iter)
y=y_exact(x);
plot(x,y,'o',x,y_approx,'r');
er = norm(y-y_approx,inf)/norm(y,inf);
xlabel('x');
ylabel('y');
```

График зависимости y от x (приближенные решения примерно совпадают с точным решением) при h=0.01 и eps=0.001: