Отчёт по предмету

«Линейные операторы и интегральные уравнения»

студента 3 курса гр.09-722

Октябрьской Алины

Тема:

«Метод простой итерации»

1. *Сколько итераций потребовалось, чтобы найти решение*
   Количество итераций записиывается в переменную iter. После завершения программы iter = 16;
2. *Как зависит от шага сетки скорость сходимости и точность вычислений*

```
clear all
clc
f = @(x) x*0 + 1;
K = @(x,s) x*0 + s*0 + 1;
a = 0;
b = 7;
h = 0.07;
eps = 1e-03;
y_exact = @(x) exp(x);

H = linspace(10^(-3), 0.7, 100);
N = zeros(1, 100);
Err = zeros(1, 100);
for i = 1:numel(H)
    h = H(i);
x = a : h : b;
[y_approx,iter] = IterVolt(x,h,eps,f,K);
y=y_exact(x);
N(i) = iter;
Err(i) = norm(y-y_approx,inf)/norm(y,inf);
end
plot(H, N);
ylim([15 20]);
figure
plot(H, Err);
```

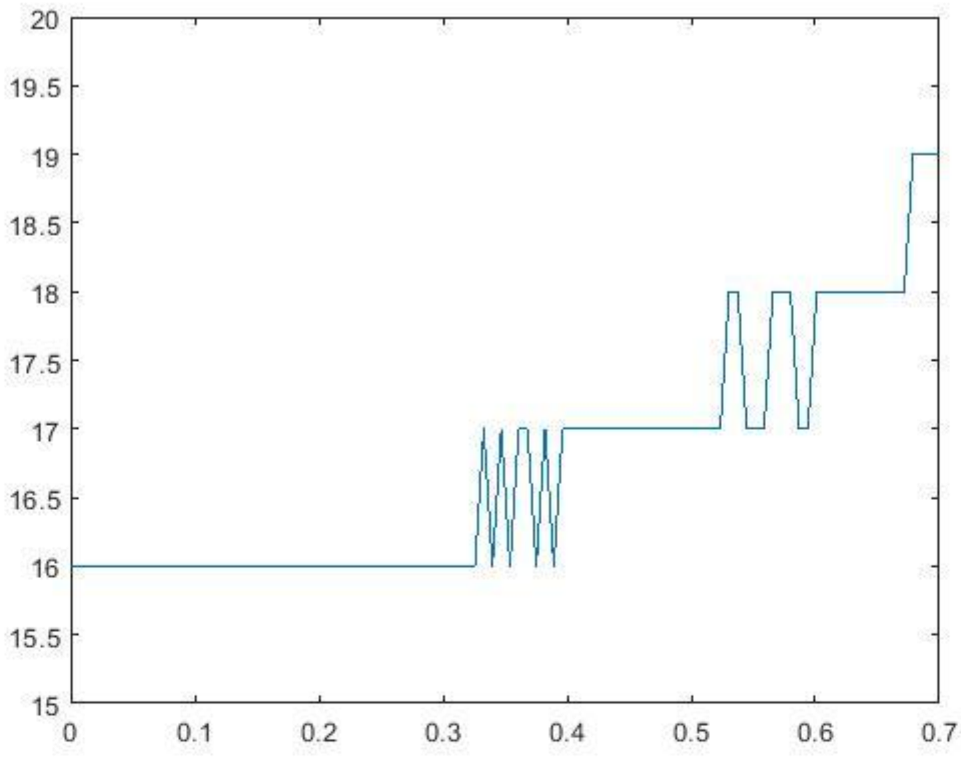


График зависимости количества итераций от длины шага h

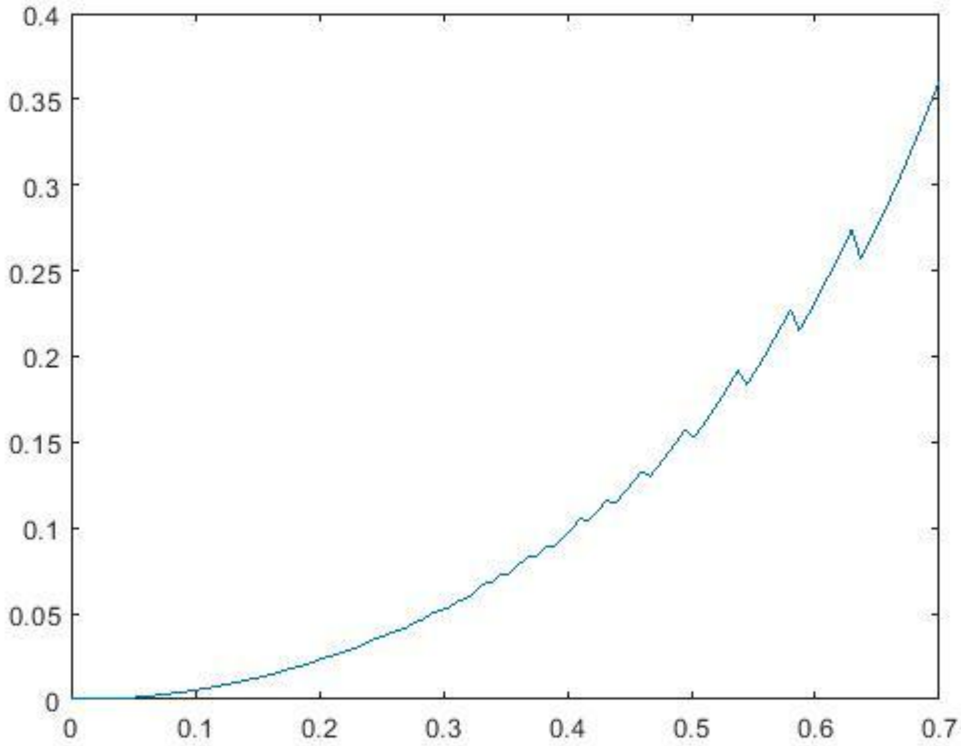


График зависимости точности от длины шага h

3. *Найти приближенное решение уравнения*

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

Если точное решение $y(x) = \sin(x)$

Изменила входные данные в main

```
clear all
clc
f = @(x) x;
K = @(x,s) -x + s;
a = 0;
b = 2*pi;
h = (b-a)/100;
eps = 1e-03;
y_exact = @(x) sin(x);

x = a : h : b;
[y_approx,iter] = IterVolt(x,h,eps,f,K);
y=y_exact(x);
err = norm(y-y_approx,inf)/norm(y,inf);

plot(x,y,'o',x,y_approx,'r');
xlabel('x');
ylabel('y');
set(gca, 'XTick', 0:pi:2*pi);
xlim([0 2*pi]);
set(gca, 'XTickLabel',{'0', 'pi', '2*pi'})
```

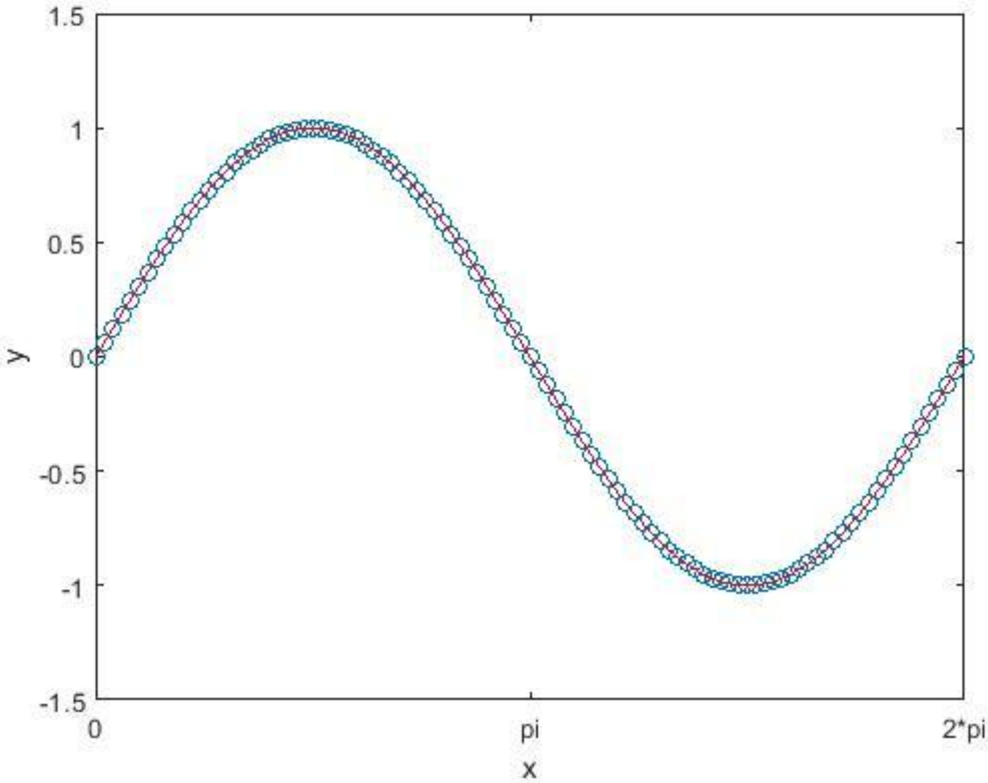


График решения

Количество итераций = 10

4. *Реализовать метод последовательных приближений на основе формулы Симпсона. Сравнить с методом трапеций.*

При одинаковом количестве узлов получилось одинаковое количество итераций, но ошибка больше при применении формулы Симпсона, т.к. число узлов четное.

Main

```
clear all
clc
f = @(x) x;
K = @(x,s) -x + s;
a = 0;
b = 2*pi;
h = (b-a)/100;
eps = 1e-03;
y_exact = @(x) sin(x);

x = a : h : b;
[y_approx,iter] = IterVolt(x,h,eps,f,K);
y=y_exact(x);
err = norm(y-y_approx,inf)/norm(y,inf);
[y_approx2,iter2] = IterVolt_test(x, h, eps,f,K);
err2 = norm(y-y_approx2,inf)/norm(y,inf);

plot(x,y,'o',x,y_approx2,'r');
xlabel('x');
ylabel('y');
```

```
set(gca, 'XTick', 0:pi:2*pi);
xlim([0 2*pi]);
set(gca, 'XTickLabel',{'0', 'pi', '2*pi'})
```

Фукнция `IterVolt_test`

```
function [yk,iter] = IterVolt_test(x,h,eps,f,K)
n = numel(x);
H = zeros(1,n);
H(1) = h/3;
H(end) = h/3;
H(2:2:end-1) = 4*h/3;
H(3:2:end-2) = 2*h/3;

y = f(x);
yk = CalcInt_test(y,H,x,n,K,f);
iter = 0;
while norm(yk-y,inf)/norm(yk,inf) > eps
y = yk;
yk = CalcInt_test(y,H,x,n,K,f);
iter = iter + 1;
end
end
```

Функция `CalcInt_test`

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

5. *Написать функцию, которая решает методом простой итерации нелинейное уравнение Вольтера второго рода. Найти приближенное решение уравнения*

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

Точное решение $y(x) = x$

Формулы, по которым будем считать

$$y_{k+1}(x) = f(x) + \int_a^x K(x, s, y_k(s))ds$$

В одной точке

$$y_{k+1}(x_i) = f(x_i) + \int_a^{x_i} K(x_i, s, y_k(s))ds$$

Представим итеграл в виде ряда

$$y_{k+1}(x_i) = f(x_i) + \sum_{j=1}^{i} A_j K(x_i, x_j, y(x_j)), i = 1,2, \dots$$

В данной задаче

$$f(x) = 0, K\big(x, s, y(s)\big) = \frac{1 + y^2(s)}{1 + s^2}$$

Значит

$$y_{k+1}(x_i) = \int_a^{x_i} A_j \frac{1 + {y_k}^2(s)}{1 + s^2} ds, i = 1,2, \dots$$

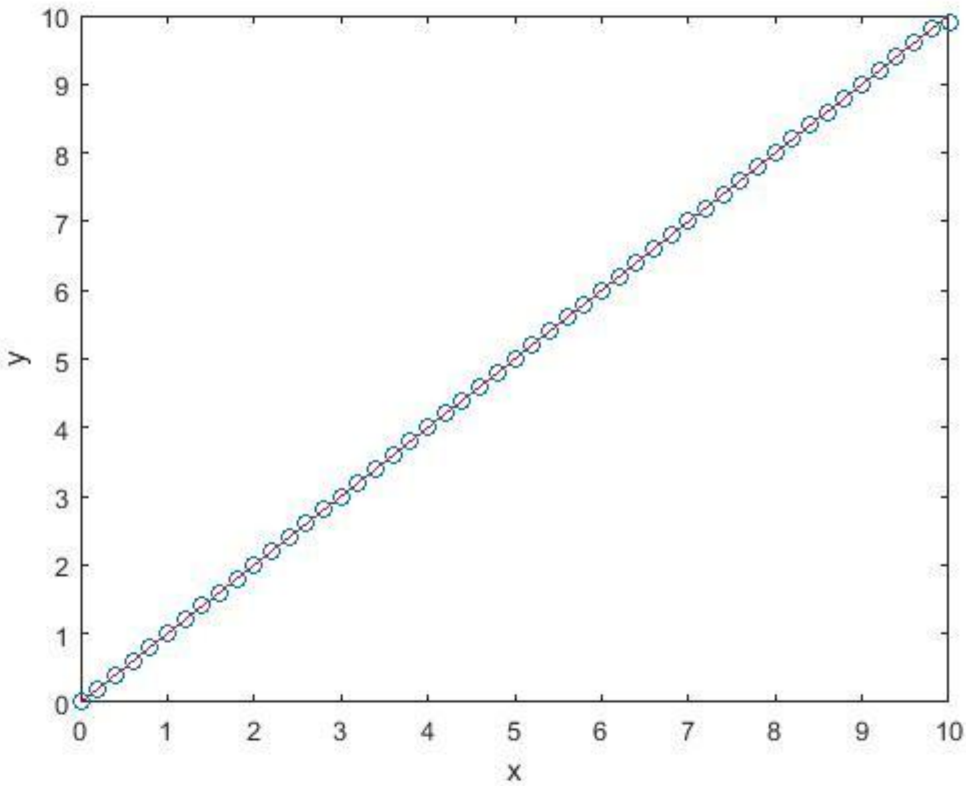
$$y_{k+1}(x_i) = \sum_{j=1}^{i} A_j \frac{1 + {y_k}^2(x_j)}{1 + {x_j}^2}, i = 2, 3, \dots$$

И для любого k $y_{k+1}(x_1) = 0$, т.к. $x_1 = a$

```
clear all
clc
a = 0;
b = 10;
n = 100;
h = (b-a)/n;
eps = 1e-06;
y_exact = @(x) x;

x = a : h : b;
[y_approx,iter] = IterVolt(x,eps);
y=y_exact(x);
err = norm(y-y_approx,inf)/norm(y,inf);

plot(x,y,'r',x,y_approx,'o');
xlabel('x');
ylabel('y');


function [yk,iter] = IterVolt(x,eps)
n = numel(x);
y = zeros(1, n); %начальное приближение
err = 10;
iter = 0;
while err > eps
yk = CalcInt(y,x,n);
iter = iter + 1;
err = norm(yk-y,inf)/norm(yk,inf);
y = yk;
end
end


 function [yk] = CalcInt(y,x,n)
yk = y;
A = x;
```

```
A(2:(end-1)) = x(2) - x(1);
A(1) = A(2)/2;
A(end) = A(1);
for i = 1 : n
yk(i) = 0;
for j = 2 : i
yk(i) = yk(i) + A(j)*((y(j)^2 + 1)/(1 + x(j)^2));
end
end
end
```

При n = 50 относительная ошибка между приближенным и точным решением = 0.0102. При n = 100 err = 0.005



Непрерывной линией обозначено точное решение, маркерами – приближенное решение.