

Contents

- Исходные данные
- Определяем шаги между исходными точками
- Подготавливаем данные для применения метода прогонки
- Определяем коэффициенты исходных кубических сплайнов
- подготавливаем данные для графиков
- строим графики
- Функция построения сплайна
- Целевая функция, к которой интерполируемся сплайнами
- Метод прогонки в виде функции
- Проверка достаточного условия применимости метода
- Прямая прогонка
- Обратная прогонка

```
clc, clear
```

Исходные данные

```
x = [-0.25,0,0.5,1]
[m,n] = size(x);
h = zeros(1,n-1);
```

```
x =  
  
-0.2500      0      0.5000      1.0000
```

Определяем шаги между исходными точками

```
for i = 1:(n-1)  
    h(i) = x(i+1) - x(i);  
end  
h
```

```
h =  
  
0.2500      0.5000      0.5000
```

Подготавливаем данные для применения метода прогонки

```
a = zeros(1,n-3);  
b = zeros(1,n-2);  
c = zeros(1,n-3);  
d = zeros(1,n-2);  
  
for i = 1:(n-2)  
  
    if(i == 1)  
        b(i) = 2*(h(i)+h(i+1));
```

```

        c(i) = h(i+1);
        d(i) = 3/h(i+1) * (TargetFunction(x(i+2))-TargetFunction(x(i+1)))...
            -3/h(i)*(TargetFunction(x(i+1))-TargetFunction(x(i)));
elseif(i == (n-2))
    a(i-1) = h(i);
    b(i) = 2*(h(i)+h(i+1));
    d(i) = 3/h(i+1) * (TargetFunction(x(i+2))-TargetFunction(x(i+1)))...
        -3/h(i)*(TargetFunction(x(i+1))-TargetFunction(x(i)));
else
    a(i-1) = h(i);
    b(i) = 2*(h(i)+h(i+1));
    c(i) = h(i+1);
    d(i) = 3/h(i+1) * (TargetFunction(x(i+2))-TargetFunction(x(i+1)))...
        -3/h(i)*(TargetFunction(x(i+1))-TargetFunction(x(i)));
end

end
end

```

Определяем коэффициенты исходных кубических сплайнов

```

C(1) = 0;
C(2:(n-1)) = Progon(a,b,c,d);

a = zeros(1,n-1);
for i = 1:(n-1)
    a(i) = TargetFunction(x(i));
end
a

b = zeros(1,n-1);
for i = 1:(n-2)
    b(i) = 1/(h(i))*(TargetFunction(x(i+1))-TargetFunction(x(i)))-...
        h(i)*(C(i+1)+2*C(i))/3;
end
b(n-1) = 1/(h(n-1))*(TargetFunction(x(n))-TargetFunction(x(n-1)))-2/3*...
    h(n-1)*C(n-1);
b

C

d = zeros(1,n-1);
for i = 1:(n-2)
    d(i) = 1/(3*h(i))*(C(i+1)-C(i));
end
d(n-1) = (-C(n-1))/(3*h(n-1));
d

```

подготавливаем данные для графиков

```

x0 = x;
y0 = TargetFunction(x);
count = 0.01;

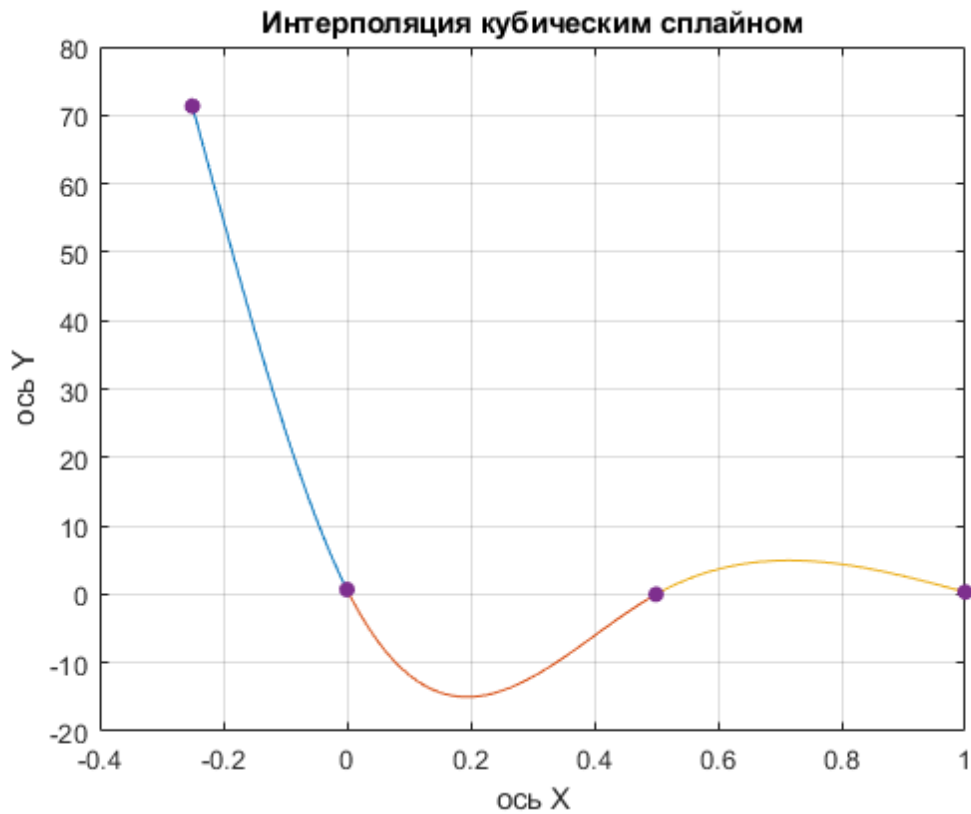
x1 = -0.25:count:0;
x2 = 0:count:0.5;
x3 = 0.5:count:1;

Splain1 = Splain(x1,x,a(1),b(1),C(1),d(1),1);
Splain2 = Splain(x2,x,a(2),b(2),C(2),d(2),2);
Splain3 = Splain(x3,x,a(3),b(3),C(3),d(3),3);

```

строим графики

```
plot(x1,Splain1,x2,Splain2,x3,Splain3);
title('Интерполяция кубическим сплайном')
xlabel('ось X')
ylabel('ось Y')
hold on;
scatter(x0, y0, 'filled');
grid on;
```



Функция построения сплайна

```
function y = Splain(x0,x,a,b,C,d,k)
    y = a+b*(x0-x(k))+C*(x0-x(k)).^2+d*(x0-x(k)).^3;
end
```

Целевая функция, к которой интерполируемся сплайнами

```
function y = TargetFunction(x)
    %y = 2.^x;
    R1 = exp((x.^4 + x.^2 - x + sqrt(5))/5);
    R2 = sinh((x.^3 + 21 .* x + 9)./(21.*x + 6));
    y = R1+R2-3.0;
end
```

a =

71.3393 0.6932 -0.0033

b =

-333.6165 -180.5197 50.6668

C =

0 612.3871 -150.0141

d =

816.5162 -508.2675 100.0094

Метод прогонки в виде функции

```
function x = Progon(a,b,c,d)
```

```
[m,n] = size(b);
```

Проверка достаточного условия применимости метода

```
for i = 1:n
    if i == 1
        if abs(b(i)) < abs(c(i))% Индивидуальное условие для первой строки
            error('Не выполнено достаточное условие применимости метода');
        end
    end

    if (1 < i && i < n)
        if abs(b(i)) < abs(a(i-1)) + abs(c(i))
            error('Не выполнено достаточное условие применимости метода');
        end
    end

    if i == n
        if abs(b(i)) < abs(a(i-1))% Индивидуальное условие для последней строки
            error('Не выполнено достаточное условие применимости метода');
        end
    end
end
```

Прямая прогонка

```
alfa = zeros(1,n-1); % Изначально заполняем нулями вектор
                        % коэффициентов альфа с заданным размером (n-1)
beta = zeros(1,n);    % Изначально заполняем нулями вектор
                        % коэффициентов бета с заданным размером (n)

for i = 1:n
    if i == 1          % Индивидуальный расчёт для коэффициентов первой строки
        gamma = b(i);
        alfa(i) = -c(i)/gamma;
        beta(i) = d(i)/gamma;
    end

    if (1 < i && i < n)
        gamma = b(i)+a(i-1)*alfa(i-1);
        alfa(i) = -c(i)/gamma;
        beta(i) = (d(i)-a(i-1)*beta(i-1))/gamma;
    end
end
```

```
    if i == n          % Индивидуальный расчёт для коэффициентов последней строки
        gamma = b(i)+a(i-1)*alfa(i-1);
        beta(i) = (d(i)-a(i-1)*beta(i-1))/gamma;
    end
end
```

Обратная прогонка

```
x = zeros(1,n);
for i = n:-1:1
    if i == n % Индивидуальный расчёт вектора решений для последней строки
        x(n) = beta(n);
    end

    if i < n
        x(i) = beta(i) + alfa(i)*x(i+1);
    end
end
```

```
end
```