Integral equations

Taks 1

<u>**п.1.**</u> Розв'язати задачу 1.***K***, використовуючи чисельний варіант методу заміни ядра виродженим (кількість доданків розвинення **m >= 5**) з урахуванням варіанту ***V* = *mod(g+d,2)***.

Тут через ***K*** позначено номер за списком студента у групі, **g** – номер групи, ***d*** – день народження студента.

1.***8*** $$u(x)+\int_0^1 (1+t)tg(2xt)\cdot u(t)dt = V + x; \quad x\in[0,1].$$

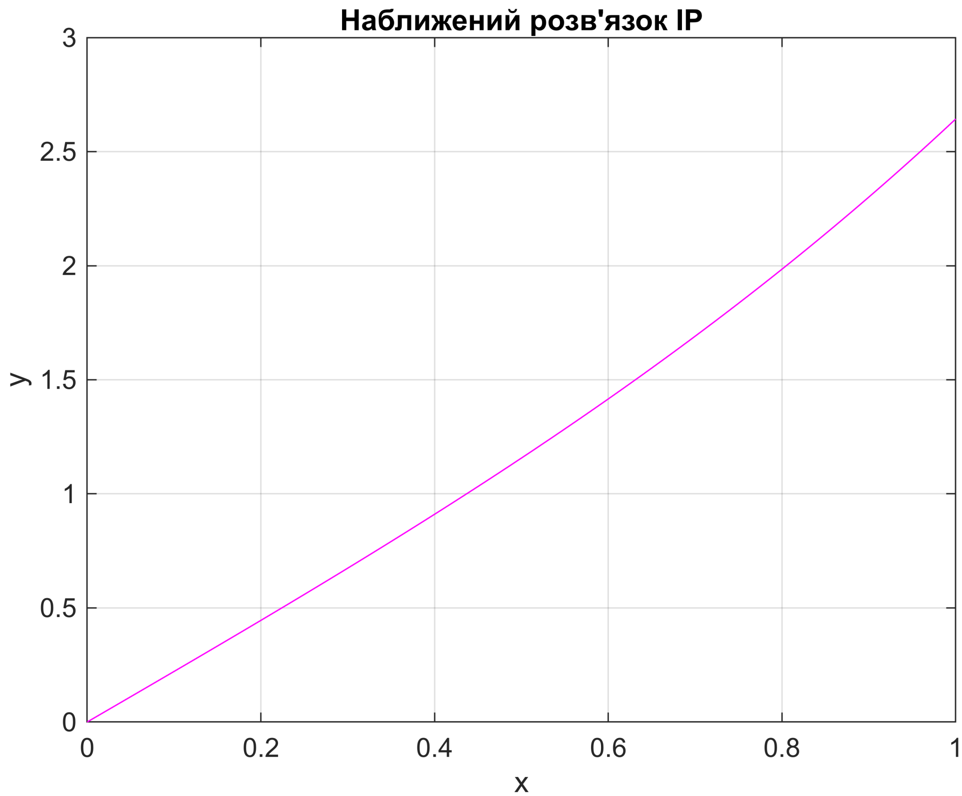
```
global m;
m = 6;

% kernel = @(x, t)((1 + t) * tan(2*x*t));

lambda = 1;
a = 0; b = 1;
n = 101;

[X,Y,n] = fr2_mzj(lambda, a, b, n, @fun1_k, @(x)(x));

plot(X,Y,'m'), title('Наближений розв''язок IP')
xlabel('x'), ylabel('y'), grid on;
```



Taks 2

<u>**п.2.**</u> Розв'язати задачу 2.***K*** (нелінійне інтегральне рівняння Фредгольма-Урисона), використовуючи чисельний метод і КФ з кількістю вузлів = 5 згідно варіанту ***V* = *mod(g+d,5)***. Значення ***V*** : **0** – формули трапецій; **1** - формули Сімпсона; **2** – формули Гаусса; **3** – формули Маркова; **4** – формули Чебишева. Для розв'язання нелінійної системи використати метод Ньютона.

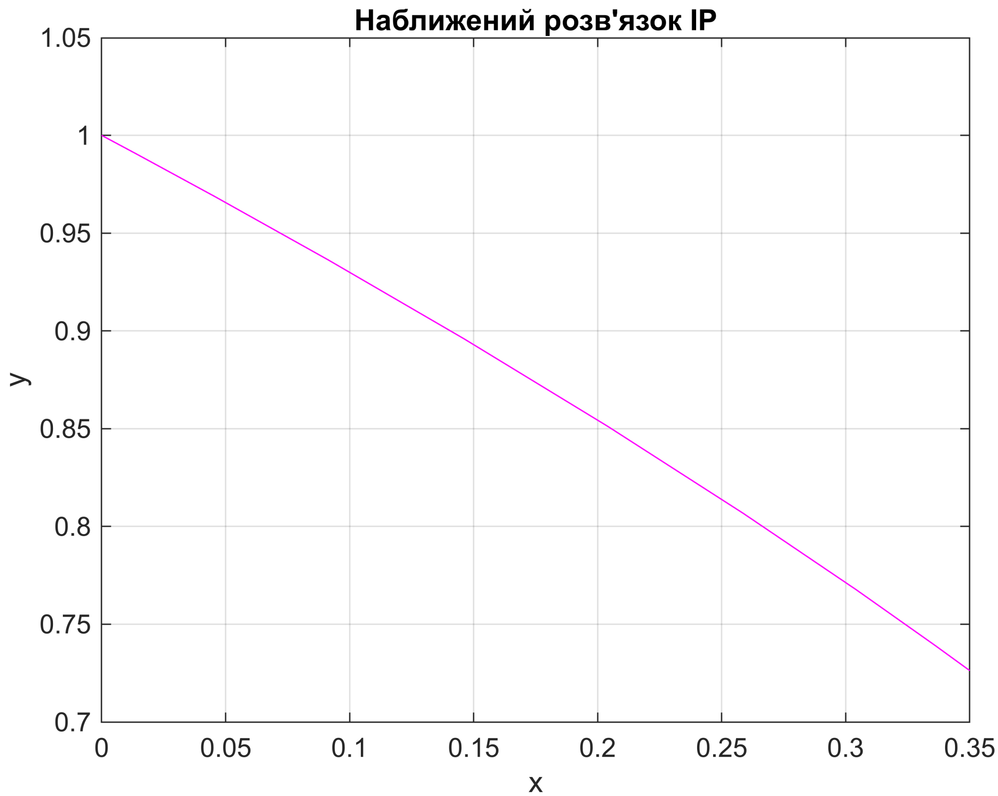
2.8

$$u(x)+\int_0^{0.35}\left(1-x-u^2(t)\right)dt=x; \quad x\in[0,0.35].$$

```
% test case
lambda = 1;
a = 0; b = 0.35;
n = 10; % we use Markov QD

eps = 1e-8; mi = 500;

Y = 0.05.*ones(n,1);
[X,Y] = frur_mk(a,b,n,@frurmk_k,@frurmk_f,Y,eps,mi);
plot(X,Y,'m'), title('Наближений розв''язок IP')
xlabel('x'), ylabel('y'), grid on
```




```
function [A,B] = fun1_k(x)
    A = [x, x^3/3, x^5/30, x^7/840, x^9/45360, x^11/3991680];
    B = [1, 1, x, x^2, x^3, x^4];
end


function [X, Y, n] = fr2_mzj(lambda, a, b, n, kfun, ffun)
    % Розв'язок лінійного неоднорідного інтегрального рівняння
    % типу Фредгольма другого роду методом заміни ядра виродженим
    % b
    % y(x)=f(x)+lambda* ∫ K(x,s)*y(s)ds,
    % a
    % де K(x,s)=SUM(k=1,m,Ak(x)*Bk(s))
    % вхідні дані :
    % lambda - числовий параметр рівняння;
    % a - нижня границя інтеграла;
    % b - верхня границя інтеграла;
    % n - число точок сітки
    % kfun - функція користувача, з описом ядра K(x,s);
    % ffun - функція користувача, з описом правої частини f(x);
    % вихідні дані:
    % X - масив вузлів сітки;
    % Y - масив наближеного розв'язку;
```

```
    % n - розмірність масивів X, Y.
    global m

    if n < 3
        n = 3;
    end

    if mod(n,2) == 0
        n = n + 1;
    end

    % формування вузлів
    X = linspace(a, b, n); h = X(2) - X(1); n1 = n - 1;

    % формування коефіцієнтів P(i), i=1,...,n квадратурної формули
    P = zeros(n,1); FF = P; Y = P;
    P(1) = h ./ 3; c2 = 2 .* P(1); c1 = 2 .* c2;
    for i = 2 : 2 : n1
        P(i) = c1; P(i+1) = c2;
    end
    P(n) = P(1);

    % формування допоміжних матриць AK, BK і вектора FF,
    % де AK(k,i)=Ak(X(i)), BK(k,i)=Bk(X(i)),
    % FF(i)=f(X(i)), i=1,...,n.
    for i = 1 : n
        x = X(i);
        [AK(:, i), BK(:, i)] = kfun(x); FF(i) = ffun(x);
    end

    % формування матриці A і вектора F СЛАР A*C=F,
    % де A(i,j)=if(i=j,1,0)-lambda*SUM(k=1,n, P(k)*AK(j,k)*BK(i,k) ),
    % F(i) =SUM(k=1,n, P(k)*FF(k)*BK(i,k) ), i=1,...,n
    F = zeros(m,1); A = zeros(m,m);
    for i = 1 : m
        for j = 1 : m
            aa = 0;
            for k = 1 : n
                aa = aa + P(k) .* AK(j, k) .* BK(i, k);
            end
            aa = -lambda .* aa;
            if i == j
                aa = 1 + aa;
            end
            A(i,j) = aa;
        end

        aa = 0;

        for k = 1 : n
```

```
            aa = aa + P(k) .* FF(k) .* BK(i, k);
        end
        F(i) = aa;
    end
    % розв'язок СЛАР A*C=F
    C = A\F;
    % розв'язок у вигляді
    % y(x)=f(x)+lambda * SUM(k=1, m, C(k) * Ak(x) )

    for i = 1 : n
        Y(i) = FF(i) + lambda .* dot(C, AK(:, i));
    end
end
```

```
function [y, yp] = frurmk_f(x,u)
    y = 1 - x - u^2;
    yp = -2*u;
end

function [y, yp] = frurmk_k(x, t, u)
    y = x;
    yp = 1;
end

function [X, Y] = frur_mk(a, b, n, k_dkfun, f_dffun, Y, e, mki)
    % Розв'язок нелінійного інтегрального рівняння
    % типу Фредгольма-Урисона методом заміни
    % інтеграла скінченною сумою
    % b
    % ∫ K(x,s,u(s))ds = f(x,u(x)), x ϵ [a,b].
    % a
    % вхідні дані :
    % a - нижня границя інтеграла;
    % b - верхня границя інтеграла;
    % k_dkfun - функція користувача, з описом ядра K(x,s,y)
    % і dK(x,s,u)/du;
    % f_dffun - функція користувача, з описом правої частини f(x,y)
    % і df(x,u)/du;
    % Y - початкове наближення до розв'язку;
    % e - точність (e>0);
    % mki - максимальна кількість кроків ітерацій;
    % вихідні дані:
    % X - масив вузлів сітки;
    % Y - масив наближеного розв'язку.

    % формування вузлів сітки
    F = Y; J = zeros(n,n);
    % формування коефіцієнтів P(i), i=1,...,n
```

```
    % квадратурної формули
    P = ones(1, n);

    X = linspace(-1, 1, n);
    X(9) = 0.91953391; X(2) = - X(9);
    X(8) = 0.73877386; X(3) = - X(8);
    X(7) = 0.47792495; X(4) = - X(7);
    X(6) = 0.16527896; X(5) = - X(6);
    P(10) = 0.02222222; P(1) = P(10);
    P(9) = 0.13330599; P(2) = P(9);
    P(8) = 0.22488934; P(3) = P(8);
    P(7) = 0.29204268; P(4) = P(7);
    P(6) = 0.32753976; P(5) = P(6);
    % transform if [a, b] != [-1, 1]
    if a ~= -1 || b ~= 1
        ab = 0.5 .* (a + b); ba = 0.5 .* (b - a);
        for i = 1 : n
            X(i) = ab + ba .* X(i);  P(i) = ba .* P(i);
        end
    end



    s = 0; Fl = 1;
    while Fl && s < mki
        % знаходження розв'язку нелінійної системи :
        % формування матриці Якобі J і вектора F
        for i = 1:n
            xi = X(i); sk = 0;
            for j = 1:n
                p = P(j);
                [k,dk] = k_dkfun(xi,X(j),Y(j));
                J(i,j) = p.*dk; sk = sk + p.*k;
            end
            [f,df] = f_dffun(xi,Y(i));
            J(i,i) = J(i,i) - df;
            F(i) = sk - f;
        end

        d = J\F;

        Y = Y - d;
        s = s + 1;
        sk = norm(d);
        Fl = sk > e;
    end
    Y = abs(Y);
end
```