

```

% programa diferencas finitas
clc
% domínio espacial [xi,xf]x[xi,xf]
xi = 0.;
xf = 2.;

% domínio temporal [ti,tf]

ti = 0.;
tf = 0.5;

%numero de repeticoes para convergencia
n=1;

for k=1:n

    % numero de elementos
    nel = 4^(k) % numero de nos = nel + 1

    % dimensao do elemento
    h = (xf-xi)/nel;
    h2 = h*h;
    dt = h;

    % coordenadas
    x = zeros(nel+1,1);
    y = zeros(nel+1,1);
    y(1)=xi;
    x(1)=xi;
    for i=2:nel+1
        x(i)= (i-1)*h;
        y(i)= (i-1)*h;
    end

    sig = dt/h2;

    % definição do vetor de variaveis
    ux = zeros(nel+1,nel+1);

    % condicao inicial
    uy = sin(pi*x(:)/2.)*sin(pi*y(:)/2.)';

    t = ti;
    while t < tf

        t = t + dt/2.;
        % primeiro passo solução em x
        for j=2:nel
            % montagem da matriz
            a = (-sig/2.)*ones(1,nel-1); % diagonal inferior
            b = (1. + sig)*ones(1,nel-1); % diagonal principal
            c = (-sig/2.)*ones(1,nel-1); % diagonal superior

            % montagem do vetor fonte
            f = (1-sig)*uy(2:nel,j) + (sig/2.)*(uy(2:nel,j+1) +
uy(2:nel,j-1));

```

```

        % condicoes de contorno
        ux(1,j) = exp(-
pi*pi*t/2.)*sin(pi*x(1)/2.)*sin(pi*y(j)/2.);
        ux(nel+1,j) = exp(-
pi*pi*t/2.)*sin(pi*x(nel+1)/2.)*sin(pi*y(j)/2.);

        f(1) = f(1) + ux(1,j)*sig/2.;
        f(nel-1) = f(nel-1) + ux(nel+1,j)*sig/2.;

        % resolucao do sistema pelo algoritmo de Thomas
        ux(2:nel,j) = TDMAsolver(a,b,c,f);
    end

    t = t + dt/2.;
    % segundo passo solucao em y
    for i=2:nel
        % montagem da matriz
        a = (-sig/2.)*ones(1,nel-1); % diagonal inferior
        b = (1. + sig)*ones(1,nel-1); % diagonal principal
        c = (-sig/2.)*ones(1,nel-1); % diagonal superior

        % montagem do vetor fonte
        f = (1-sig)*ux(i,2:nel) + (sig/2.)*(ux(i+1,2:nel) + ux(i-
1,2:nel));

        % condicoes de contorno
        uy(i,1) = exp(-
pi*pi*t/2.)*sin(pi*x(i)/2.)*sin(pi*y(1)/2.);
        uy(i,nel+1) = exp(-
pi*pi*t/2.)*sin(pi*x(i)/2.)*sin(pi*y(nel+1)/2.);

        f(1) = f(1) + uy(i,1)*sig/2.;
        f(nel-1) = f(nel-1) + uy(i,nel+1)*sig/2.;

        % resolucao do sistema pelo algoritmo de Thomas
        uy(i,2:nel) = TDMAsolver(a,b,c,f);
    end

end

erro(k)=max(max(abs(uy(:, :)-exp(-
pi*pi*t/2.)*sin(pi*x(:)/2.)*sin(pi*y(:)/2.))));
dh(k)=h;
end

plot(-log(dh),log(erro));

```