Table of Contents

HW4, CEE6513

Author: Donglai Yang Date: Oct 22, 2023

```
L = 1000;
% Analytical solution
u_func = @(x,L) transpose(-0.5*x.^2 + L*x);
% Numerical solution is defined below
```

iterate mesh sizes

```
hs = logspace(-6,-1,6)*L;
err = zeros(size(hs));
for ii = 1:length(hs)
    u_a = u_func(0:hs(ii):L, L);
    u_n = FDfunc(L,hs(ii));
    % error
    err(ii) = sqrt(mean((u_n - u_a).^2));
end

degree = 1;
coefficients = polyfit(log(hs), log(err), degree);
slope = coefficients(1);
fprintf('The slope is %.1f\n',slope)

% make an error plot
figure;
loglog(hs, err,'-.o','LineWidth',2); hold on;
xlabel('h','FontSize',14)
ylabel('Error','FontSize',14)
```

Function

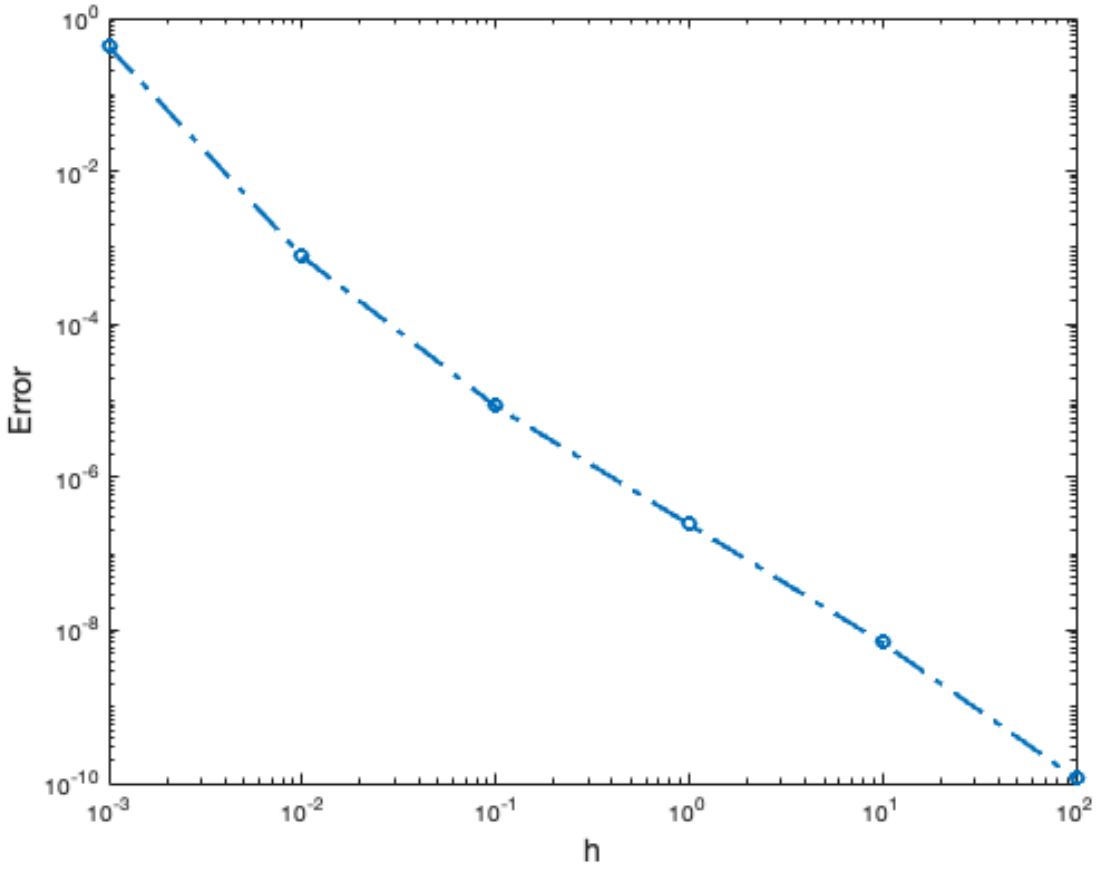
```
function [u, z] = FDfunc(L, h)
%FDFUNC Finite difference calculation of 1D steady 2nd order ODE

    % create coordinate
    z = 0:h:L;
    N = length(z);
    ee = ones(N,1);
    A = spdiags([ee -2*ee ee], -1:1, N, N);
    % Dirichlet BC
    A(1,1:3) = [1,0,0];
```

```
    % second order backward difference for the Neumann BC
    A(end,end-2:end) = [1,-4, 3];
    % b vector
    b = -h^2*ones(N,1);
    b(1) = 0;
    b(end) = 0;
    % inversion
    u = A\b;

end
```

The slope is -1.8



Published with MATLAB® R2022b