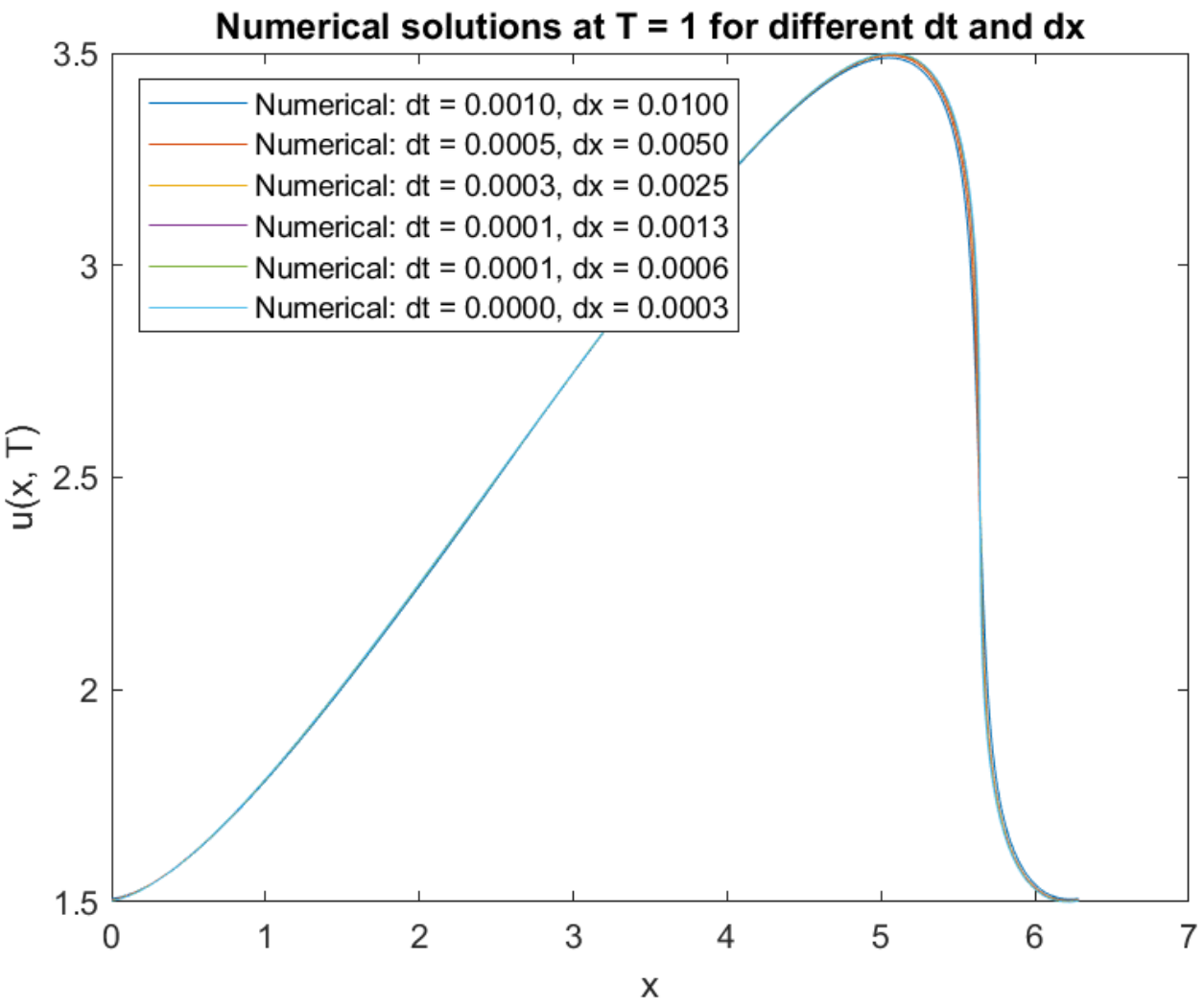
```
% Define domain and initial parameters
L = 2*pi;          % Domain length
T = 1;             % Final time
dx = .01;         % Initial spatial step size
dt = 0.001;          % Initial time step size
N = round(L/dx);   % Number of spatial grid points
M = round(T/dt);   % Number of time grid points
figure;
% Loop over different values of dt and dx
for i = 1:6
   % Initialize solution matrix
   u = zeros(N+1, M+1);   % u(i,j) represents u at x = (i-1)*dx and t =
(j-1)*dt
   % Set initial condition
   for j = 1:N+1
       x = (j-1)*dx;
       u(j,1) = 2.5 + sin(x);
   end
   % boundary condition
   u(1,:) = u(end-1,:);
   u(end,:) = u(2,:);
   % Time-stepping loop
   for j = 1:M
       for k = 2:N
           if u(k,j) >= 0
               u(k,j+1) = u(k,j) - dt*u(k,j)*(u(k,j) - u(k-1,j))/dx;
           else
               u(k,j+1) = u(k,j) - dt*u(k,j)*(u(k+1,j) - u(k,j))/dx;
           end
       end
       % boundary condition
       u(1,j+1) = u(end-1,j+1);
       u(end,j+1) = u(2,j+1);
   end
   % solution at the final time level
   u_final = u(:,end);
   % Analytical solution
   x = linspace(0, L, N+1);
   u_analytical = 2.5 + sin(x - T); % Analytical solution
   % Compute error and order of accuracy
   error = norm(u_final-u_analytical,inf);
   errors(i) = error;
   if i > 1
       order = log(errors(i-1)/errors(i))/log(2);
       orders(i) = order;
   end
   % Plot
   plot(x, u_final, 'DisplayName', sprintf('Numerical: dt = %.4f, dx = %.4f',
dt, dx));
```

```
    hold on;

    % Halve dt and dx for the next iteration
    dt = dt/2;
    dx = dx/2;
    N = round(L/dx);
    M = round(T/dt);
end
xlabel('x');
ylabel('u(x, T)');
title('Numerical solutions at T = 1 for different dt and dx');
legend('show');
legend('Location','northwest');
hold off;
% Output errors and orders of accuracy
fprintf('dt\t\t dx\t\t Error\t\t Order\n');
for i = 1:6
    fprintf('%.4e\t%.4e\t%.4e\t%.4f\n', 0.001/(2^(i-1)), 0.01/(2^(i-1)),
errors(i), abs(orders(i)));
end
```

dt	dx	Error	Order
1.0000e-03	1.0000e-02	6.2431e+02	0.0000
5.0000e-04	5.0000e-03	1.2533e+03	1.0054
2.5000e-04	2.5000e-03	2.5092e+03	1.0016
1.2500e-04	1.2500e-03	5.0232e+03	1.0014
6.2500e-05	6.2500e-04	1.0049e+04	1.0004
3.1250e-05	3.1250e-04	2.0102e+04	1.0003


Numerical solutions at T = 1 for different dt and dx
Numerical: dt = 0.0010, dx = 0.0100
Numerical: dt = 0.0005, dx = 0.0050
Numerical: dt = 0.0003, dx = 0.0025
Numerical: dt = 0.0001, dx = 0.0013
Numerical: dt = 0.0001, dx = 0.0006
Numerical: dt = 0.0000, dx = 0.0003
u(x, T)
x
3.5
3
2.5
2
1.5
0
1
2
3
4
5
6
7