Mini-Matlab Lesson 10: Runge-Kutta 4

Contents

The Runge-Kutta formula involves a weighted average of values of f(t,y) at different points in the interval between t_n and t_{n+1}. It is given by

$$y_{n+1} = y_n + \frac{h}{6}(k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4})$$

The second term (divided by h) can be interpreted as an average slope (or if you like, the approximation of the integral equation [see Improved Euler's Method] using Simpson's rule. The slope components are

$k_{n1} = f(t_n, y_n)$ (use slope at left end)

$k_{n2} = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_{n1})$ (use slope at midpoint via Euler's formula and k_{n1})

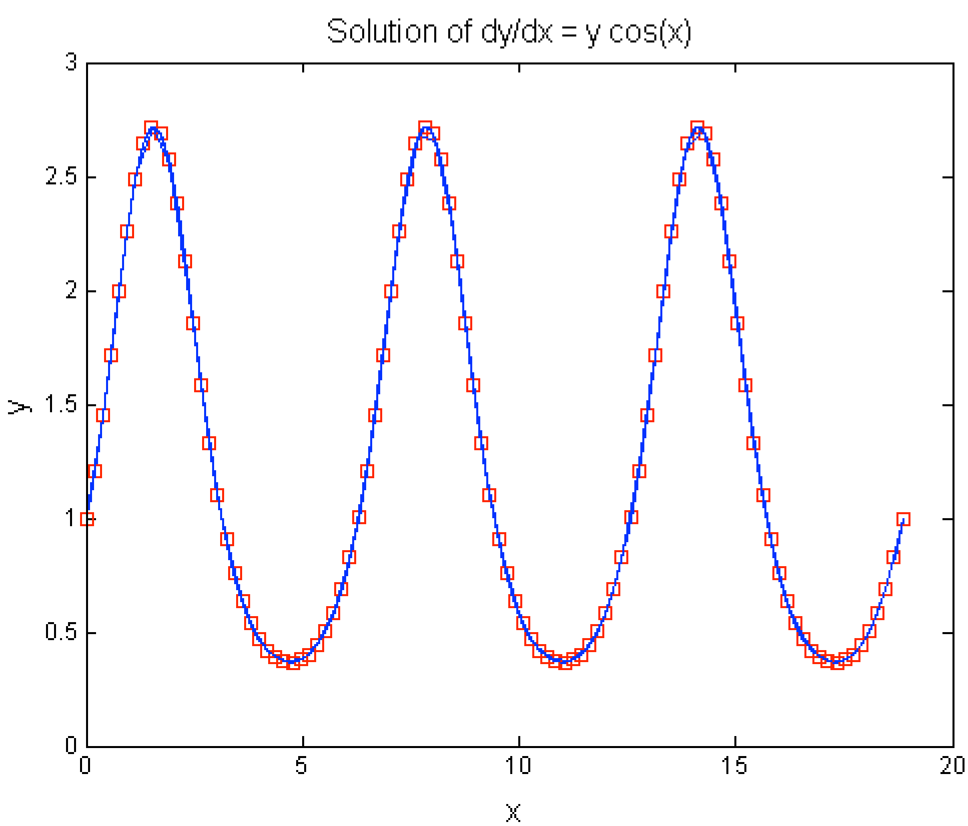
$k_{n3} = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_{n2})$ (use slope at midpoint via Euler's formula and k_{n2})

$k_{n4} = f(t_n + h, y_n + hk_{n3})$ (slope at right endpoint using Euler's formula and k_{n3})

```
clear;
close all;

nmat = [50 100 200 400];

f = @(x,y) y*cos(x);
g = @(x)exp(sin(x));

xfine = linspace(0, 6*pi, 100);     % A nice, fine mesh

figure(1);
plot(xfine, g(xfine), 'rs-');
hold on;

for j = 1:length(nmat)

    tic

    n = nmat(j);

    x = linspace(0, 6*pi, n);
    h = x(2) - x(1);               % Discretization size

    y(1) = 1;                      % Initial value

    for k = 2:n

        k1 = f(x(k-1), y(k-1));
        k2 = f(x(k-1) + h/2, y(k-1) + h/2*k1);
```

```
        k3 = f(x(k-1) + h/2, y(k-1) + h/2*k2);
        k4 = f(x(k-1) + h, y(k-1) + h*k3);

        y(k) = y(k-1) + h/6*(k1 + 2*k2 + 2*k3 + k4);
    end

    plot(x, y, 'b');
    drawnow;

    err(j) = norm(g(x) - y, inf);
    disp(['When n = ', num2str(n), ' the error is ', num2str(err(j))]);

    time(j) = toc;
end

hold off;
xlabel('x', 'FontSize', 16);
ylabel('y', 'FontSize', 16);
title('Solution of dy/dx = y cos(x)', 'Fontsize', 16);
```

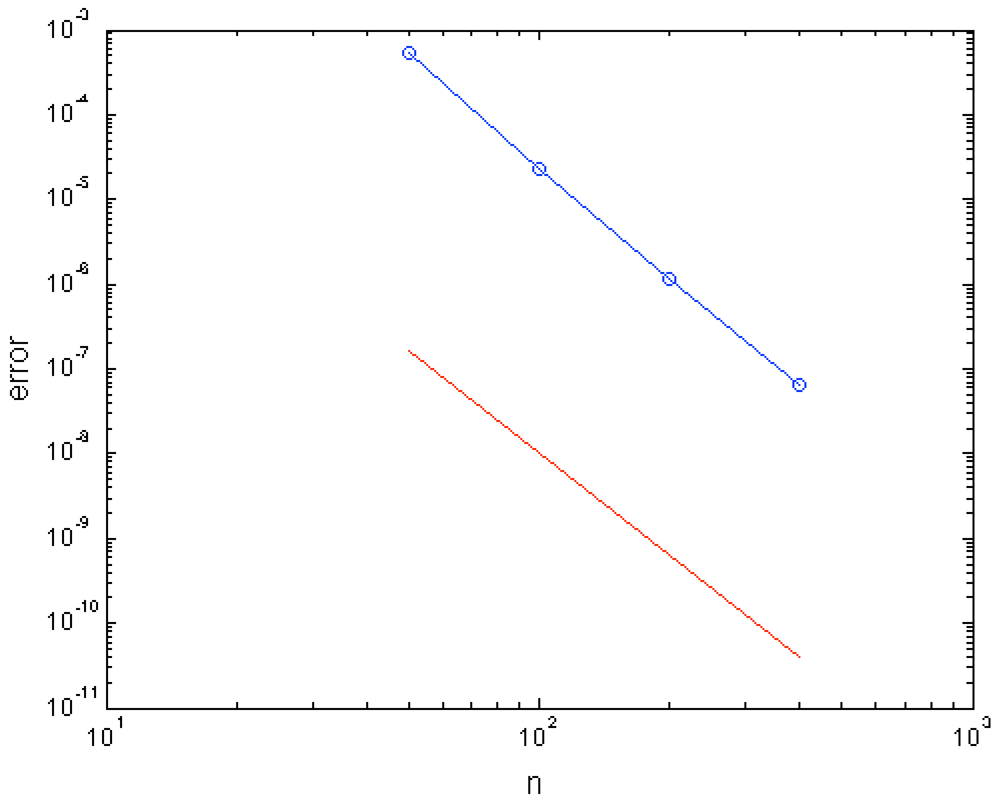
```
When n = 50 the error is 0.00053384
When n = 100 the error is 2.2864e-05
When n = 200 the error is 1.1471e-06
When n = 400 the error is 6.3251e-08
```



We can plot the error involved on a loglog plot

We can check that the error is indeed $O(h^4)$.

```
figure(2)
loglog(nmat, err, 'bo-');
hold on
loglog(nmat, 1./nmat.^4, 'r');
hold off
xlabel('n', 'FontSize', 16);
ylabel('error', 'FontSize', 16);
```



Published with MATLAB® 7.12