Runge-Kutta Method

Math 230

The following *Mathematica* code implements the Runge-Kutta method:

```
f[t_, y_] := your function here
y[0] = your initial value here
Δt = your step size here
t[k_] := k * Δt

Do [
  m = f[ t[k], y[k] ];
  n = f[ t[k + 1/2], y[k] + m * Δt/2 ];
  q = f[ t[k + 1/2], y[k] + n * Δt/2 ];
  p = f[ t[k + 1], y[k] + q * Δt];
  y[k + 1] = y[k] + (m + 2n + 2q + p) * Δt/6
, {k, 0, your number of steps here} ]
```

1. Consider the initial-value problem

$$\frac{dy}{dt} = t^2 + y, \qquad y(0) = 2.$$

(a) Use the Runge-Kutta method to approximate $y(1)$ using the following numbers of steps: $1, 2, 4, 8, \ldots, 2^{10}$

(b) Find the exact value of $y(1)$.

(c) Compute the error for some (or all!) of your approximations. By what factor does the error decrease when you double the number of steps?

2. Consider the initial-value problem:

$$\frac{dy}{dt} = 1 + ty^2, \qquad y(0) = 2.$$

Use the Runge-Kutta method to approximate $y(0.5)$. Use several different step sizes and compare the errors, as in the previous problem.

Since we don't know how to solve this differential equation, you might use `DSolve` or `NDSolve` to find the "exact" value of $y(1)$.