MATLAB Practical II: Solving ODEs etc.

This practical concentrates on quadrature and solving ordinary differential equations in MATLAB. Try some of the examples on the handout "MATLAB Practical II: Solving ODEs etc." to satisfy yourself that no black magic was involved. Then answer the following questions.

1. Consider the function $g(x) = -\log|\cos(x)|$.

   Write an m-file `g.m` that defines $g(x)$ (like `f.m` on this morning's handout). Compute the integral $\int_0^1 g(x)\,\mathrm{d}x$ using the function `quad` or `quadl`. Adjust the tolerance to convince yourself that you've computed the first eight digits accurately, and copy them down here. [ ]

2. Solve the initial value problem $y'(x) = xy$, $y(0) = 1$ on the interval $x \in [0, 2]$ using both `ode23` and `ode45`. (You will need to write your own m-file, similar to `f1.m` on this morning's handout.)

   Compare the numerical results to the exact solution (which you can find via separation of variables) at the point $x = 2$.

   How many digits after the decimal point are correct? [ode23: ode45:]

3. Type `help ode23` and `help odeset` to learn more about the parameter choices for MATLAB's ODE solvers.

   Which property "displays computational cost statistics"? [ ]

   Adjust the `reltol` and `abstol` parameters to force the steplength to be shorter. For example, to set the relative tolerance in the example using `f1.m` in this morning's handout:

   ```
   options = odeset('reltol', 1e-12)
   [x,y] = ode23('f1', [0 10], 0, options)
   ```

   How many digits of accuracy can you now obtain for $y(2)$ in Problem 2? [ ]

please turn over...

4. Produce a `quiver` plot of the phase plane for the nonlinear system of equations

$$\begin{aligned} y_1' &= y_1 + y_2 - y_1\,(y_1^2 + y_2^2) \\ y_2' &= -y_1 + y_2 - y_2\,(y_1^2 + y_2^2) \end{aligned}$$

for $y_1 \in [-2, 2]$, $y_2 \in [-2, 2]$.
Type in the `plottraj.m` script from the lecture handout and plot a few solutions. What limiting behaviour is observed in this phase plane?

For more information on this example and those shown in the lecture, see Chapter 4 of *Advanced Mathematical Methods for Scientists and Engineers* by Bender and Orszag, a book you may find useful for your Mathematical Methods lectures.

5. Use `help bvp4c` and `help bvpinit` to learn how to solve boundary value ODEs in MATLAB. We will now solve a boundary value problem for the ODE specified in `f2.m` in the lecture sheeet, $y'' = -y$. (You will need to type in that function.) Now define the function `f2bc.m` to contain:

```
function bc = f2bc(ya,yb);
 bc = [ya(1)+1; yb(1)-2];
```

This file defines the boundary conditions $y(a) = -1$, $y(b) = 2$. We shall take $a = 0$ and $b = 1$. To solve the boundary value problem using `bvp4c`, type:

```
solinit = bvpinit([0:.1:1],[1 0]);
sol = bvp4c('f2','f2bc',solinit);
plot(sol.x, sol.y(1,:));
```

Figure out how to change the second boundary condition to $y'(1) = 0$. (Hint: Study the parameters to the `f2bc` function.) Solve this equation with the new boundary conditions. What is the approximate value of $y(x)$ at $x = 0.5$? [ ]

(Please do this using `bvp4c`, not by solving the equation analytically!)