MATH 128A, SUMMER 2009: HOMEWORK 6 SOLUTIONS

Example code to solve the numerical problems:

```
% Create the table of solutions (by 8 methods) to the IVP from HW6.
% t(i+1):   Value of t_i
% w(i+1,j): Value of w at t_i using method #j
% names{j}: Name 7-letter-long name for method #j

% Set up the IVP:
f = @(t,y) 1+y./t;
h = 0.25;
t = 1:h:2;
n = length(t)-1;
w(1,1:8) = 2;

for i = 1:n
names{1} = ' Exact ';
w(i+1,1) = t(i+1).*(log(t(i+1))+2);

names{2} = 'Taylor2';
df = @(t,y) 1./t; % Total derivative of f(t,y(t))
w(i+1,2) = w(i,2) + h*f(t(i),w(i,2)) + h^2/2*df(t(i),w(i,2));

names{3} = 'M.Euler';
k1 = h*f(t(i)  ,w(i,3)   );
k2 = h*f(t(i)+h,w(i,3)+k1);
w(i+1,3) = w(i,3) + 1/2*k1 + 1/2*k2;

names{4} = 'A-B (2)';
if i == 1
w(i+1,4) = w(i+1,1); % Use exact value for w_1.
else
w(i+1,4) = w(i,4) + h/2*(3*f(t(i),w(i,4)) - f(t(i-1),w(i-1,4)));
end

names{5} = 'A-M (1)';
% Algebraic solution to w = a + b*f(t,w):
% w = a + b + (b/t)w ==> (1-b/t)*w = a+b
sol = @(a,b,t) (a+b) ./ (1-b./t);
w(i+1,5) = sol(w(i,5) + h/2*f(t(i),w(i,5)), h/2, t(i+1));

names{6} = 'APC 1,1';
wpredict = w(i,6) + h*f(t(i),w(i));
w(i+1,6) = w(i,6) + h/2*(f(t(i+1),wpredict) + f(t(i),w(i,6)));
end
```

Date: Thursday 7/30.

```
names{7} = '  RK4  ';
[trk4,wrk4] = rk4(f, 1, 2, 2, n);
w(:,7) = wrk4;

[trkf,wrkf] = rkf(f, 1, 2, 2, 1e-4, 0.05, 0.25);

% Print the table: Start with first 7 columns, which use the same t values.
fprintf(' t  ');
for j=1:7
fprintf('|%s', names{j});
end
fprintf('\n');
fprintf('----+-------+-------+-------+-------+-------+-------+-------\n');

for i = 1:n+1
fprintf('%4.2f', t(i));
for j=1:7
fprintf('| %5.3f ', w(i,j));
end
fprintf('\n');
end
% RKF gets its own table, below.
fprintf('\n');
fprintf(' t  |  RKF  \n');
fprintf('----+-------\n');

for i = 1:length(trkf)
fprintf('%4.2f| %5.3f \n', trkf(i), wrkf(i));
end
```

Output:

```
 t  | Exact |Taylor2|M.Euler|A-B (2)|A-M (1)|APC 1,1|  RK4
----+-------+-------+-------+-------+-------+-------+-------
1.00| 2.000 | 2.000 | 2.000 | 2.000 | 2.000 | 2.000 | 2.000
1.25| 2.779 | 2.781 | 2.775 | 2.779 | 2.778 | 2.775 | 2.779
1.50| 3.608 | 3.612 | 3.601 | 3.613 | 3.606 | 3.601 | 3.608
1.75| 4.479 | 4.485 | 4.469 | 4.488 | 4.476 | 4.469 | 4.479
2.00| 5.386 | 5.394 | 5.373 | 5.399 | 5.382 | 5.373 | 5.386
 t  |  RKF
----+-------
1.00| 2.000
1.25| 2.779
1.50| 3.608
1.75| 4.479
2.00| 5.386
```

It's also possible to put these into a single table, since the RKF method was able to meet the given tolerance without lowering h.

RK problem: Compute Taylor terms...

...for $y(t_i + h)$ (using series for $y(t)$ centered at t_i):

$$
\begin{aligned}
y(t) &= y(t) \\
&\to \boxed{w_i} \\
y'(t) &= g(y(t)) \\
&\to \boxed{g(w_i)h} \\
y''(t) &= g'(y(t))g(y(t)) \\
&\to \boxed{[g'(w_i)g(w_i)]\frac{h^2}{2!}} \\
y'''(t) &= g''(y(t))g(y(t))^2 + g'(y(t))^2 g(y(t)) \\
&\to \boxed{\left[g''(w_i)g(w_i)^2 + g'(w_i)^2 g(w_i)\right]\frac{h^3}{3!}}
\end{aligned}
$$

...for w_{i+1} (as a function of h):

$$
\begin{aligned}
w_{i+1} &= w_i + b_1 h g(w_i) + b_2 h g(w_i + ahg(w_i)) \\
&\to \boxed{w_i} \\
w'_{i+1} &= b_1 g(w_i) + b_2 g(w_i + ahg(w_i)) \\
&\quad + b_2 h g'(w_i + ahg(w_i)) a g(w_i) \\
&\to \boxed{[(b_1 + b_2)g(w_i)]h} \\
w''_{i+1} &= 2b_2 g'(w_i + ahg(w_i)) a g(w_i) \\
&\quad + b_2 h g''(w_i + ahg(w_i))[ag(w_i)]^2 \\
&\to \boxed{[2b_2 a g'(w_i)g(w_i)]\frac{h^2}{2!}} \\
w'''_{i+1} &= 3b_2 g''(w_i + ahg(w_i))[ag(w_i)]^2 \\
&\quad + b_2 h g'''(w_i + ahg(w_i))[ag(w_i)]^3 \\
&\to \boxed{[3b_2 a^2 g''(w_i)g(w_i)^2]\frac{h^3}{3!}}
\end{aligned}
$$

By this point in the calculation we have three equations for the coefficients, just from getting as much agreement between these Taylor terms as possible:

$$
\begin{aligned}
1 &= b_1 + b_2 \\
1 &= 2b_2 a \\
1 &= 3b_2 a^2
\end{aligned}
$$

So $a = 2/3$, $b_2 = 3/4$, $b_1 = 1/4$.

Even with this choice the $O(h^3)$ Taylor coefficients don't match, so $y_{i+1} - w_{i+1} = O(h^3)$ and $\tau_{i+1} = O(h^2)$.

Multistep Method Problem:

Let $y(t) = c_0 + c_1(t - t_i)^1 + c_2(t - t_i)^2 + c_3(t - t_i)^3$.

Get $y'(t) = c_1(t - t_i)^0 + 2c_2(t - t_i)^1 + 3c_3(t - t_i)^2$.

Then the goal is to make these power series match to the highest order possible:

$$
\begin{array}{rlccccccccccl}
 & & c_0 & + & c_1 h^1 & + & c_2 h^2 & + & c_3 h^3 & + & \cdots & \\
\approx & a_1[& c_0 & & & & & & & & &] \\
 & +a_0[& c_0 & - & c_1 h^1 & + & c_2 h^2 & - & c_2 h^3 & + & \cdots &] \\
 & +b_1[& & & c_1 h^1 & & & & & & &] \\
 & +b_0[& & & c_1 h^1 & - & 2c_2 h^2 & + & 3c_2 h^2 & - & \cdots &]
\end{array}
$$

Indeed, we can make all of the terms written if we solve the linear system

$$
\begin{array}{cccccccccc}
1 & = & a_1 & + & a_0 & & & & \\
1 & = & & - & a_0 & + & b_1 & + & b_0 \\
1 & = & & & a_0 & & & - & 2b_0 \\
1 & = & & - & a_0 & & & + & 3b_0
\end{array}
$$

The unique solution is $(a_1, a_0; b_1, b_0) = (-4, 5; 4, 2)$.

Taking the power series one term farther would show that $\tau_{i+1} = O(h^3)$. But it doesn't matter anyway, as this method's instability makes it useless.