Math 337 Homework 03 Rewrite

Andy Reagan

February 13, 2014

1. Using Taylor expansions of y'_{i-1} and y'_{i-2} about $x = x_i$, verify that

$$y_i''' = \frac{y'_i - 2y'_{i-1} + y'_{i-2}}{h^2} + O(h).$$

Solution: Using the Taylor expansions, we have

$$y'_{i-1} = y'_i - hy''_i + \frac{h^2}{2}y'''_i + O(h^3) \tag{1}$$
$$-2y'_{i-1} = -2y'_i + 2hy''_i - h^2y'''_i + O(h^3) \tag{2}$$
$$y'_{i-2} = y'_i - 2hy''_i + 2h^2y'''_i + O(h^3) \tag{3}$$

Using these expansions, notice that the y_i and hy_i terms cancel as follows

$$-2y'_{i-1} + y'_{i-2} = -y_i + h^2y'''_i + O(h^3) \tag{4}$$

such that

$$\frac{y'_i - 2y'_{i-1} + y'_{i-2}}{h^2} = \frac{h^2y'''_i + O(h^3)}{h^2} = y'''_i + O(h), \tag{5}$$

as desired.

4 Use the Predictor-Corrector method given by Eqs 3.33, 3.39, and 3.40 to solve

$$y' = \sin y, \qquad y(0) = 1, \qquad x \in [0, \pi]. \tag{6}$$

Select the step size so that the local truncation error, given by 3.39, be at most $\epsilon_{\text{loc}} = 10^{-4}$. Provide an explanation for your choice of the step size.

Solution: Using formula 1.28, the third derivative of y is

$$y'''(x) = f_{xx} + f_xf_y + 2ff_{xy} + f(f_y)^2 + f^2f_{yy} \tag{7}$$

I compute the respective derivatives as

$$\begin{aligned}
f_x &= \cos(y(x))\sin(y(x)) && (8)\\
f_{xx} &= \cos(y(x))\cos(y(x))\sin(y(x)) + \sin(y(x))(-\sin(y(x))\sin(y(x))) && (9)\\
&= \cos^2(y(x))\sin(y(x)) - \sin^3(y(x)) && (10)\\
f_y &= \cos(y(x)) && (11)\\
f_{yy} &= -\sin(y(x)) && (12)\\
f_{xy} &= \cos^2(y(x)) - \sin^2(y(x)) && (13)
\end{aligned}$$

Therefore we have

$$\begin{aligned}
y'''(x) &= \cos^2(y(x))\sin(y(x)) - \sin^3(y(x)) + \cos^2(y(x))\sin(y(x)) && (14)\\
&\quad + 2\sin(y(x))(\cos^2(y(x)) - \sin^2(y(x))) + \sin(y(x))\cos^2(y(x)) - \sin^2(y(x))\sin(y(x)) && (15)\\
&= 5\cos^2(y(x))\sin(y(x)) - 3\sin^3(y(x)) && (16)
\end{aligned}$$

I'm not confident in this equation, so I set the max of y''' to 1 per our discussion. With the error given by Equation 3.39, we compute h for $\epsilon_{\text{loc}} = 10^{-4}$ and choose the max of $y'''(x)$ on $[0, \pi]$. For the max of y''' set to 1, we find $h = 0.1062$.

In Figure 1, we plot the numerical solution. We test whether our timestep controlled the error in Figure 2. We see that the error estimate which we made was sufficient to contain the error within $\epsilon_{\text{loc}} = 10^{-4}$.

The step size computation, shown in the code, is equivalent to solving 3.37 with our estimate of y''' given above:

$$|\epsilon_{i+1}| \approx 1/12 \cdot h^3 |y'''_{i+1}| \qquad \Rightarrow \qquad h \approx \sqrt[3]{12|\epsilon_{\text{loc}}|/|y'''_{i+1}|}.$$

```
% prb4_2.m
% solve the ODE
%    y' = sin(y), y(0) = 1
% using a multi-step method

% IC
y0 = 1;

% define the function
dy = @(t,y,params) sin(y);

% parameters (there are none)
params = [];

% find the max of y''' on [0,pi]
x = linspace(0,pi,100);
% y'''
ytp = 5.*cos(x).^2.*sin(x)-3.*sin(x).^3;
% take a quick look
figure;
plot(x,ytp);
```

```
hold on; plot(x(find(ytp==max(ytp),1)),max(ytp),'s','MarkerSize',5,'MarkerFaceColor','
    k');
xlabel('x','FontSize',16);
ylabel('y','FontSize',16);
saveas(gcf,'andy_hw03_prb4_ytp.png'); hold off;

% solve for h per equation 3.37
ytp_max = 1; % max(abs(ytp))
h = ((12e-4)/ytp_max)^(1/3);
fprintf('h is %.6f\n',h);

% set up the interval
x = 0:h:pi;
% initialize arrays
y = zeros(1,length(x)); y(1) = y0;
error = zeros(1,length(x));

% make a first guess using order O(h^2) method
% here I first chose ME for 2nd order

k1 = h*dy(x(1),y(1),params);
k2 = h*dy(x(1)+h,y(1)+k1);
y(2) = y(1)+1/2*(k1+k2);

y_p_vec(2) = y(2);

% loop over interval
for i=3:length(x)
    % grab current point
    x_i = x(i);
    % predictor
    y_p = y(i-1) + 0.5*h*(3*dy(x(i-1),y(i-1),params)-dy(x(i-2),y(i-2),params));
    % corrector
    y_c = y(i-1) + 0.5*h*(dy(x(i-1),y(i-1))+dy(x(i),y_p));
    % note the (-) sign because we subtract 3.39 from Y_c
    y(i) = 1/6*(y_p+5*y_c);
    % error estimate
    error(i) = abs(y_p- y_c)/6;
end

% make lots of plots, save them all
% first, the numerical solution

% cell of the data
yplot={y,error};
% cell of the legends
legendcell={'numerical','error estimate'};
figname={'numerical','just_error'};
for i=1:length(yplot)
    figure; hold on;
    plot(x,yplot{i},'-');
    xlabel('x','FontSize',16);
    legend(legendcell{i});
    ylabel('y','FontSize',16);
    saveas(gcf,sprintf('andy_hw03_prb4_%s.png',figname{i}));
end

% print those out
fprintf('max estimated error is %.5f\n',max(error));
```

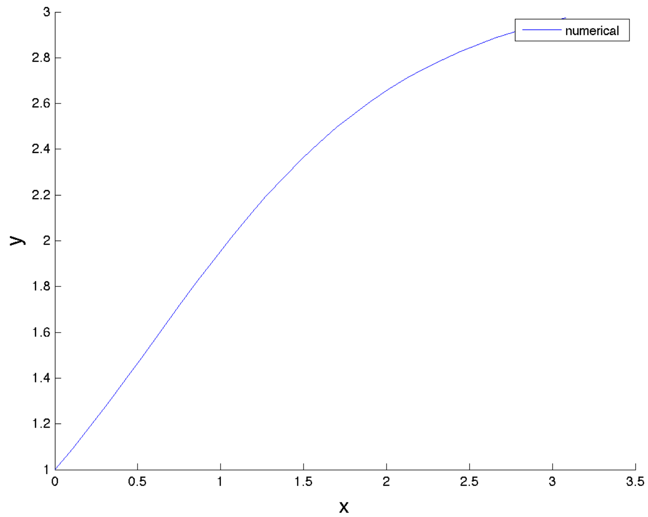

Figure 1: The numerical solution.

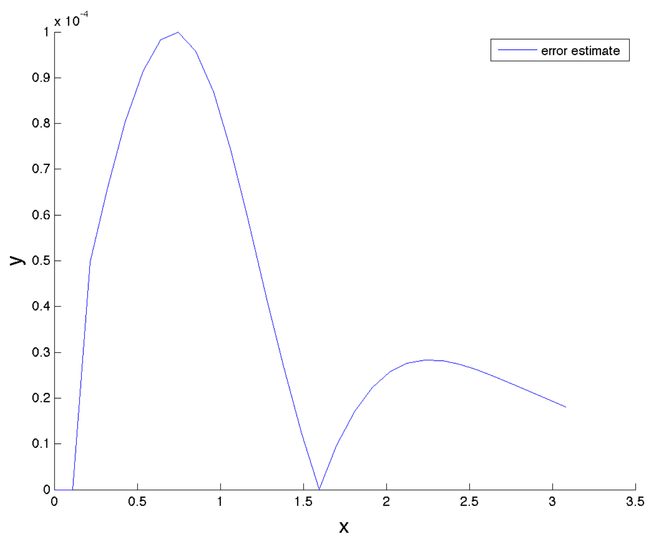

Figure 2: The estimated error at each time step.