

Math 337 Homework 03

Andy Reagan

February 4, 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: From Equation 3.10 we have that

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

Taking the derivative of this we have

$$y'''_i = \frac{y''_i - y''_{i-1}}{h}.$$

Putting equation 3.10 into the above, we have

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

as desired.

2. Obtain the counterpart of the linear system in Equation 3.15 without assuming that $x_i = 0$.

Solution: Without making that assumption, we have the system of equations

$$f = 1 \Rightarrow \int_{x_i}^{x_{i+1}} 1 \, dx = h \cdot (b_0 \cdot 1 + b_1 \cdot 1 + b_2 \cdot 1) \quad (1)$$

$$f = x \Rightarrow \int_{x_i}^{x_{i+1}} x \, dx = h \cdot (b_0 \cdot x_i + b_1 \cdot (x_i - h) + b_2 \cdot (x_i - 2h)) \quad (2)$$

$$f = x^2 \Rightarrow \int_{x_i}^{x_{i+1}} x^2 \, dx = h \cdot (b_0 \cdot x_i^2 + b_1 \cdot (x_i - h)^2 + b_2 \cdot (x_i - 2h)^2) \quad (3)$$

Solving both the integrals and expanding the RHS, this system becomes

$$h = hb_0 + hb_1 + hb_2 \quad (4)$$

$$x_i h + \frac{h^2}{2} = hb_0 x_i + hb_1 x_i - h^2 b_1 + hb_2 x_i - 2b_2 h^2 \quad (5)$$

$$x_i^2 h + h^2 x_i + \frac{h^3}{3} = hb_0 x_i^2 + hb_1 x_i^2 - 2b_1 x_i h^2 + b_1 h^3 + b_2 x_i^2 h - 4b_2 h^2 x_i + 4b_2 h^3 \quad (6)$$

We begin solving this system of three equations and three unknowns by diving by h in Equation 4 to obtain

$$1 = b_0 + b_1 + b_2 \quad (7)$$

and grouping the hx_i terms in Equation 5 for

$$\frac{h^2}{2} + x_i h = hx_i(b_0 + b_1 + b_2) - h^2(b_1 + 2b_2). \quad (8)$$

Using Equation 7 in the above Equation 8, we cancel and divide by h^2 to obtain

$$-\frac{1}{2} = 2b_2 + b_1 \Rightarrow b_1 = -2b_2 - \frac{1}{2}. \quad (9)$$

Similarly, we group terms in $x_i^2 h$ and $h^2 x_i$ in Equation 6, using Equation 7 and 9 to cancel them, respectively, and grouping by powers of h^3 are left with

$$h^3\left(\frac{1}{3} - b_1 + 4b_2\right) = 0. \quad (10)$$

We stop here, having arrived at precisely the form of Equations 3.15 (7,9 and 10 here).

3. Find the coefficients b_{-1}, b_0, b_1 in

$$Y_{i+1} = Y_i + h(b_{-1}f_{i+1} + b_0f_i + b_1f_{i-1})$$

that produce a 3-rd order method.

Solution: Following the derivation in Section 3.2, I approximate the solution as the integral of f , and require the coefficients to create a method that is exact for a third degree polynomial, $y = p_3$. Therefore, we represent f as a second-degree polynomial p_2 have solve the system

$$f = 1 \Rightarrow \int_0^h 1 dx = h \cdot (b_{-1} \cdot 1 + b_0 \cdot 1 + b_1 \cdot 1) \quad (11)$$

$$f = x \Rightarrow \int_0^h x dx = h \cdot (b_{-1} \cdot (x_i + h) + b_0 \cdot x_i + b_1 \cdot (x_i - h)) \quad (12)$$

$$f = x^2 \Rightarrow \int_0^h x^2 dx = h \cdot (b_{-1} \cdot (x_i + h)^2 + b_0 \cdot (x_i)^2 + b_1 \cdot (x_i - h)^2) \quad (13)$$

Equation 11 is the same form as Equation 6 in Problem 2, and so we have a similar result reducing this. Using the assumption that $x_i = 0$ in the Equations 12 and 13 leaves us with the two equations:

$$h^2/2 = h \cdot (b_{-1}h - b_1h) \quad (14)$$

$$h^3/3 = h \cdot (b_{-1}h^2 + b_1h^2) \quad (15)$$

Cancelling the h terms in the above two equations, we substitue to find b_1, b_{-1} and use Equation 6 to obtain b_0 . We find

$$b_1 = 5/12, b_0 = 2/3, \text{ and } b_{-1} = -1/12. \quad (16)$$

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

$$y' = \sin y, \quad y(0) = 1, \quad x \in [0, \pi]. \quad (17)$$

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: First note that using separation of variables, the analytical solution to the ODE given above is

$$y(t) = 2 * \cot^{-1} \left(e^{-t} \cot(1/2) \right). \quad (18)$$

Using formula 1.28, the third derivative of y is

$$y'''(x) = -8 \sin(x) \cos(x) - 2 \sin^2(x) \cos^4(x) - 6 \sin^2(x) \cos(x). \quad (19)$$

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 $[0, \pi]$.

Figure 1 is a plot of y''' , with the max as a big square. We use the maximum to compute the timestep. In Figure 2, we plot the numerical solution and the analytical solution. We test whether our timestep controlled the error in Figure 3. Figure 4 gives us an idea of the improvement of Equation 3.40 over Equations 3.33, by plotting the difference in true error of Y_p and Y_{i+1} . We see that the error estimate which we made was sufficient to contain the error within $\epsilon_{\text{loc}} = 10^{-4}$.

All of these figures are created, saved, and closed individually in the script `andy_hw03_prb4.m`. The step size computation, shown in the code, is equivalent to solving 3.37 with our estimate of y''' given above:

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

I also plot and save the improvement of Equation 3.40 over 3.35, that is, the difference in the actual error of each method. Looking at the code, I took care to make the plots in the a loop. I also loop over the initialization method, using a Modified Euler method for second order accuracy and Kutta's third order method for third order accuracy. To see the results for the ME start, run the code. Here, for brevity, I just include the results starting with RK3.

```

% prb4.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);
ay = @(t,y,params) 2*acot(exp(-t)*cot(1/2));

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

% find the max of y''' on [0,pi]
x = linspace(0,pi,100);
% y'''
ytp = -4.*sin(x).*cos(x)-2.*(sin(x).^2).*cos(x)+(cos(x).^3)-3.*sin(x).^2.*cos(x);
% take a quick look
h = figure;
plot(x,ytp);
hold on; plot(x(find(ytp==max(ytp),1)),max(ytp),'s','MarkerSize',5,'MarkerFacecolor','r');
saveas(h,'andy_hw03_prb4-ytp.png'); close(h); hold off;

methodcell = {'ME','RK3'};
for methodcount=1:2
    method = methodcell{methodcount};
    fprintf('running with initial method %s\n',method);
    % solve for h per equation 3.37
    h = ((12e-4)/max(ytp))^(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;
    y_p_vec = zeros(1,length(x)); y_p_vec(1) = y0;
    error = zeros(1,length(x));
    an = zeros(1,length(x)); an(1) = ay(x(1),y(1),params);

    % make a first guess using order  $O(h^2)$  method
    % here I first chose ME for 2nd order
    % but the modification of the method made it third order
    % so I want a third order method to begin it

```

```

switch method
    case 'ME'
        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);
    case 'RK3'
        k1 = h*dy(x(1),y(1),params);
        k2 = h*dy(x(1)+0.5*h,y(1)+0.5*k1);
        k3 = h*dy(x(1)+h,y(1)-k1+2*k2);
        y(2) = y(1)+1/6*(k1+4*k2+k3);
end
y-p-vec(2) = y(2);
an(2) = ay(x(2),y(2),params);

% 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));
    % save the predictors
    y-p-vec(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(i) - y_p)/6;
    % analytical solution
    an(i) = ay(x_i,y(i),params);
end

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

yplot={y,y,an,error,error,abs(an-y),abs(abs(an-y)-abs(an-y-p-vec))};
legendcell={'','numerical','analytical'},'',',',{ 'estimated_error','actual_error'};
figname={'ay','both','an','error_est','error_both','error_true','improvement'};
for i=1:length(yplot)
    h = figure; hold on;
    plot(x,yplot{i},'-');
    xlabel('x');
    ylabel('y');
    switch i
        case {2,5}

```

```

        plot(x, yplot{i+1}, '—');
        legend(legendcell{i});
    end
    saveas(h, sprintf('andy_hw03_prb4_%s_%s.png', method, figname{i}));
    close(h);
end

% print those out
fprintf('max_estimated_error_is_%.5f\n', max(error));
fprintf('max_true_error_is_%.5f\n', max(abs(an-y)));

end

```

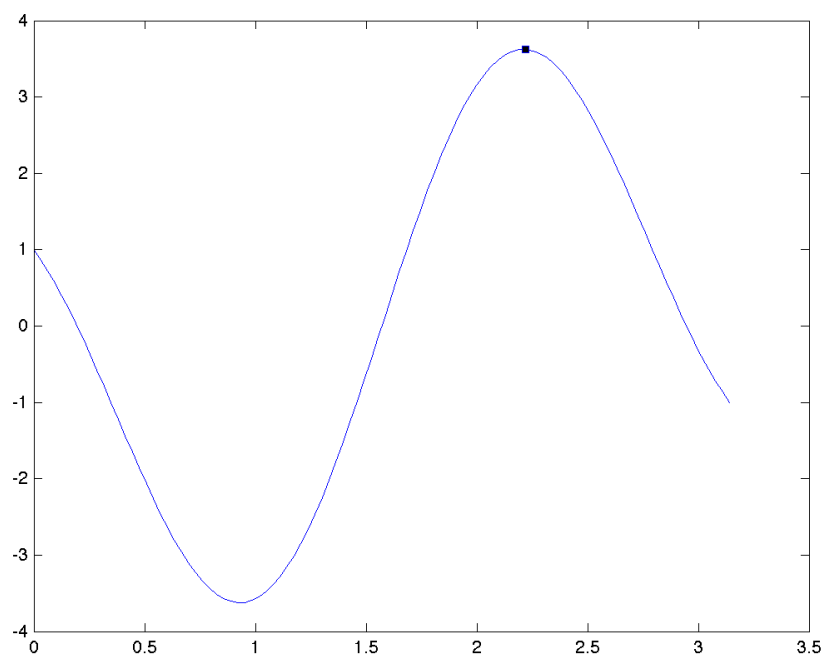


Figure 1: A plot of y''' . The max is labeled by the black box.

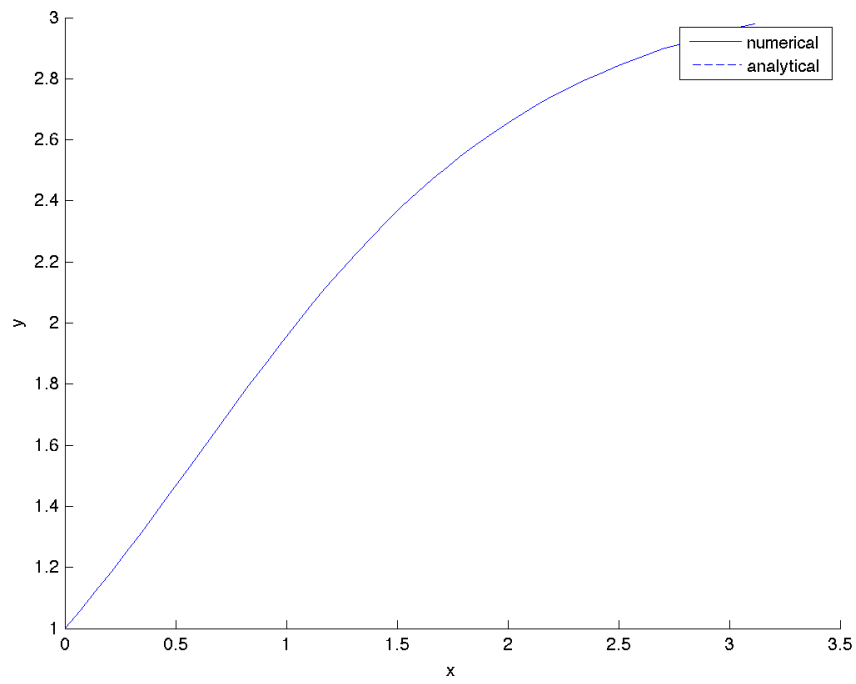


Figure 2: The numerical, and analytical, solution. They are so close that it is difficult to tell them apart.

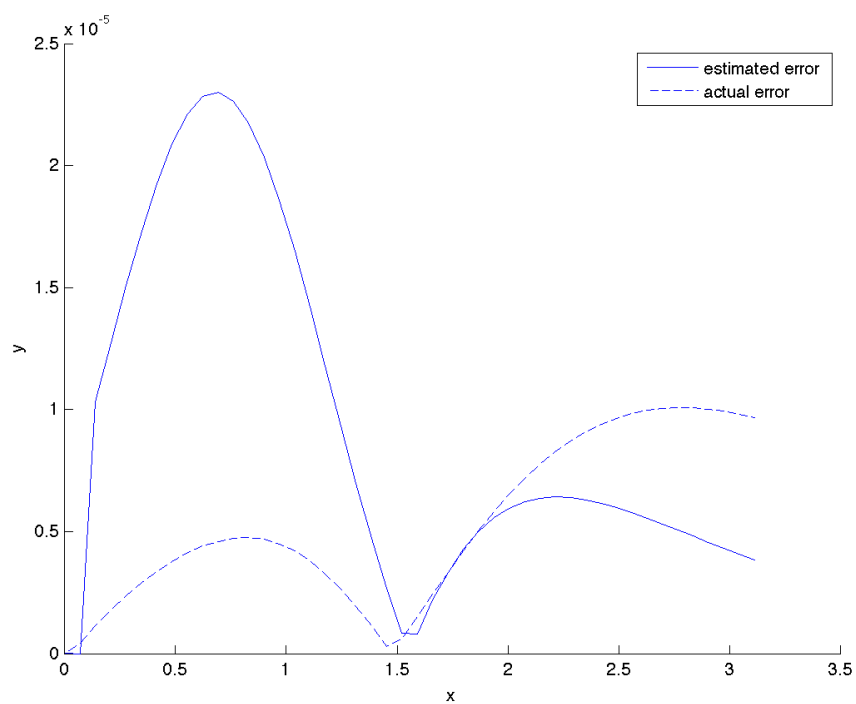


Figure 3: The estimated (dashed) and true error at each timestep of the integration.

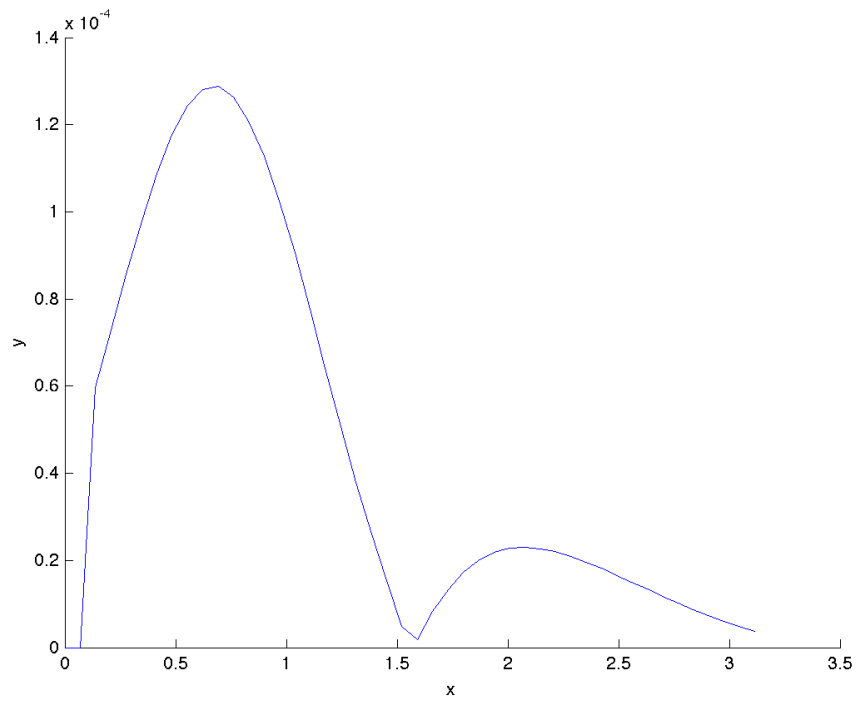


Figure 4: The improvement of Equation 3.40 over Equations 3.33, by plotting the difference in true error of Y_p and Y_{i+1} .