

## Solutions to Problem Set 2, Math 228A, Fall 2005

1. Here we need to find the error of implicit Euler.

Follow the convergence proof of explicit Euler.

We begin by finding the local truncation error (this is the ‘local residual’, which can be thought of as the additional error acquired from step  $u_n$  to  $u_{n+1}$ ). Local truncation error is very different from the true error, since the ‘localizing assumption’ is in place. In other words, to calculate the local truncation error we assume that all of our previously computed values are exactly correct. That is, that all of our previously computed data points lie exactly on the solution curve. Then, the local truncation error (for the current step) is the difference between our next computed result, and the exact solution.

$$\begin{aligned}\tau_n &= (\text{exact solution } y_{n+1} = y(t_{n+1})) \\ &\quad - (\text{whatever we get by using our method on } y_n = y(t_n) \text{ to produce } y_{n+1}) \\ &= y_{n+1} - (y_n + hf(t_{n+1}, y_{n+1})) = (y_{n+1} - y_n) - hy'_{n+1} \\ &= (y_{n+1} - y(t_n)) - hy'_{n+1} = (y_{n+1} - y(t_{n+1} - h)) - hy'_{n+1} \\ &= (y_{n+1} - (y_{n+1} - hy'_{n+1} + \frac{h^2}{2}y''_{n+1} + O(h^3))) - hy'_{n+1} \\ &= -\frac{h^2}{2}y''_{n+1} + O(h^3)\end{aligned}$$

(*Note the proof of consistency is not very trivial.*) Now we can approximate the local truncation error for each step ( $\tau_n$ ). Note that the approximation depends on the solution curve  $y$ . (so it would be most appropriate to write  $\tau_n$  as a function  $\tau_n(y)$  which depends on the  $y$  used in the localizing assumption). In other words, our local truncation error will be different depending on what solution curve we are currently lying on. So one can think of the numerical solution as follows.

We start at  $y_0$ , the initial condition. The exact solution curve  $y$  passes through  $y_0$ . We are interested in finding  $y_1 = y(t_1)$ , and so we use our scheme (implicit euler) to find  $u_1$ . Unfortunately, we make a local error of  $\tau_0 = \frac{h^2}{2}y''_1 + O(h^3)$  (depends on 2nd derivative of  $y$ ), and end up on some neighboring solution curve  $\hat{y}$ . Now we would like to find  $y_2 = y(t_2)$ , and so we use our scheme on  $u_1$  to find  $u_2$ . But now things get even worse, since  $u_1$  lies on  $\hat{y}$ , and  $u_2$  isn't even the correct value for  $\hat{y}_2$ . In fact, since we make yet another error of  $\tau_1 = \frac{h^2}{2}\hat{y}''_2 + O(h^3)$  (depends on 2nd derivative of  $\hat{y}$ ), we end up on yet another solution curve  $\hat{\hat{y}}$  (perhaps even farther away from  $y$ ). So by now we could be way off, since  $\hat{y}$  might drift away from  $y$ , and  $\hat{\hat{y}}$

might drift away from  $\hat{y}$ , and so forth (and things might get really bad really fast). Fortunately, there are nice algebraic tricks that work for 1-step methods that allow you to obtain an upper bound for the true error ( $e_n$ ). Note  $u_n = u_{n-1} + hf(t_n, u_n)$ , **not**  $u_n = u_{n-1} + hf(t_{n-1}, u_{n-1})$ .

$$\begin{aligned} e_n &= y_n - u_n \\ &= (y_{n-1} + hf(t_n, y_n) + \tau_{n-1}) - (u_{n-1} + hf(t_n, u_n)) \\ &= e_{n-1} + h(f(t_n, y_n) - f(t_n, u_n)) + \tau_{n-1} \\ &= (y_{n-1} - u_{n-1}) + h(f(t_n, y_n) - f(t_n, u_n)) + \tau_{n-1} \\ \|e_n\| &= \|e_{n-1}\| + \|h(f(t_n, y_n) - f(t_n, u_n))\| + \|\tau_{n-1}\| \end{aligned}$$

Now here we need to assume  $f$  is Lipschitz with Lipschitz constant  $L$ .

$$\begin{aligned} \|e_n\| &\leq \|e_{n-1}\| + hL\|y_n - u_n\| + \|\tau_{n-1}\| \\ &\leq \|e_{n-1}\| + hL\|e_n\| + \|\tau_{n-1}\| \\ &\leq \left(\frac{1}{1-hL}\right)\|e_{n-1}\| + \left(\frac{1}{1-hL}\right)\|\tau_{n-1}\| \end{aligned}$$

*Many people made mistakes here.* So now we have a handy recurrence relation, and we can sift the indices all the way down to 0.

$$\begin{aligned} \|e_n\| &\leq \left(\frac{1}{1-hL}\right)^n \|e_0\| + \left(\frac{1}{1-hL}\right)\|\tau_{n-1}\| + \left(\frac{1}{1-hL}\right)^2\|\tau_{n-2}\| + \left(\frac{1}{1-hL}\right)^3\|\tau_{n-3}\| + \dots \\ &\leq \left(\frac{1}{1-hL}\right)^n \|e_0\| + \sum_{j=1}^n \left(\frac{1}{1-hL}\right)^j \|\tau_{n-j}\| \end{aligned}$$

Now we have to figure out what the leading coefficients are.

$$\frac{1}{1-hL} = 1 + hL + (hL)^2 + (hL)^3 + \dots \leq e^{2hL} \quad \text{when } |hL| \leq 0.1$$

and so

$$\left(\frac{1}{1-hL}\right)^n \leq e^{2nhL} = e^{2TL} \quad \text{when } |hL| \leq 0.1$$

**(The above step is very important! Unfortunately many people forgot it.  $T$  should be involved here, instead of  $n$ . The same applies to the next problem.)** And now we assume that every  $\|\tau_{n-j}\|$  is bounded by some  $\tau$  which is  $O(h^2)$ . (this is a valid assumption in this case, since we are assuming that  $y$  is  $C^2$ , and hence  $y''$  exists and is bounded).

$$\begin{aligned} \sum_{j=1}^n \left(\frac{1}{1-hL}\right)^j \|\tau_{n-j}\| &\leq \sum_{j=1}^n \left(\frac{1}{1-hL}\right)^j \tau \leq \tau \sum_{j=1}^n \left(\frac{1}{1-hL}\right)^j \\ &\leq \tau \frac{\left(\frac{1}{1-hL}\right)^{n+1} - 1}{\left(\frac{1}{1-hL}\right) - 1} \leq \frac{e^{2(n+1)hL} - 1}{hL} \tau \end{aligned}$$

And so we have

$$\|e_n\| \leq e^{TL}\|e_0\| + O\left(\frac{\tau}{hL}\right) + O(h^2)$$

And now if we can assume  $e_0 = O(h^2)$ , then  $\|e_n\|$  will be  $O(h)$ .

**2.** Here we need to find the error for the method  $u_{n+1} = u_n + hf(t_n, u_n) + \frac{h^2}{2}f'(t_n, u_n)$ . (let  $f'(t, y) = f_t(t, y) + Df(t, y) \cdot f(t, y)$  for convenience).

This problem has a solution very similar to that of problem 1. The same reasoning and technique applies (since we are dealing with a 1-step method). The main difficulty is that we need to come up with estimates for  $h\|f(t_n, y_n) - f(t_n, u_n)\|$  as well as  $\frac{h^2}{2}\|f'(t_n, y_n) - f'(t_n, u_n)\|$ .

The first estimate follows from the Lipschitz condition on  $f$ :

$$h\|f(t_n, y_n) - f(t_n, u_n)\| \leq hL\|y_n - u_n\| \leq hL\|e_n\|$$

And the second estimate follows from the Lipschitz condition on  $f'$ :

$$\frac{h^2}{2}\|f'(t_n, y_n) - f'(t_n, u_n)\| \leq \frac{h^2}{2}\hat{L}\|y_n - u_n\| \leq \frac{h^2}{2}\hat{L}\|e_n\|$$

Note that we need some Lipschitz constant  $\hat{L}$  for  $f'$ . This is not guaranteed, and needs to be assumed. (however, since  $f$  is  $C^2$ , we know that  $f'$  is  $C^1$  and so if we are dealing with a convex domain, we can use problem 1 to get a Lipschitz constant for  $f'$ ). We may also assume WLOG that  $L = \hat{L}$  by taking the max of them. One last fact we may need to use is

$$1 + hL + \frac{h^2}{2}L \leq 1 + 2hL$$

for small enough  $h$ . Then we use the usual technique as in the example of lecture notes.

**4.** Write down the explicit form of  $F$  in terms of  $f$ .

$F$  is in the form  $F(u_{n+1}, \dots, u_{n-k+1}; t_{n+1}, h, f)$ . For the  $t$  terms we won't be so strict to write only in terms of  $t_{n+1}$  and  $h$ .

Forward Euler	$F = f(t_n, u_n)$
Backward Euler	$F = f(t_{n+1}, u_{n+1})$
Trapezoidal	$F = \frac{1}{2}[f(t_n, u_n) + f(t_{n+1}, u_{n+1})]$
Midpoint w/ forward Euler predictor	$F = f(t_n + \frac{h}{2}, u_n + \frac{h}{2}f(t_n, u_n))$
3-step implicit Adams	$F = \frac{9}{24}f_{n+1} + \frac{19}{24}f_n - \frac{5}{24}f_{n-1} + \frac{1}{24}f_{n-2}$
BDF	$F = b_{-1}f(t_{n+1}, u_{n+1})$

Here we only verify that the assumptions are satisfied for the midpoint rule with forward Euler predictor

$$u_{n+1} = u_n + hf(t_n + \frac{h}{2}, u_n + \frac{h}{2}f(t_n, u_n))$$

Assume of course, that  $f$  is Lipschitz in both variables.  $F$  is clearly Lipschitz in its first variable. Now we show  $F$  is Lipschitz in its second variable.

$$\begin{aligned} \|(F(t, u))_n - (F(t, v))_n\| &= \|f(t_{n+\frac{1}{2}}, u_n + \frac{h}{2}f(t_n, u_n)) - f(t_{n+\frac{1}{2}}, v_n + \frac{h}{2}f(t_n, v_n))\| \\ &\leq L\|u_n + \frac{h}{2}f(t_n, u_n) - v_n - \frac{h}{2}f(t_n, v_n)\| \\ &\leq L\|u_n - v_n\| + \frac{h}{2}L\|f(t_n, u_n) - f(t_n, v_n)\| \\ &\leq L\|u_n - v_n\| + \frac{h}{2}L^2\|u_n - v_n\| = (L + \frac{h}{2}L^2)\|u_n - v_n\| \end{aligned}$$

### 5. Consistency of the BDF.

Just Taylor expansion at  $t_n$ .

### 6. Consistency of the implicit midpoint rule.

Here we have to find the local truncation error for the implicit midpoint rule. We have to be very careful though, since the implicit midpoint rule is rather tricky. The following solution is *invalid*

$$\begin{aligned} y_{n+1} &= y_n + hf(t_n + \frac{h}{2}, y_n + \frac{y_{n+1} - y_n}{2}) + \tau_n \\ y_{n+1} - y_n &= hf(t_n + \frac{h}{2}, y_n + \frac{y_{n+1} - y_n}{2}) + \tau_n \\ (y + hy' + \frac{h^2}{2}y'' + O(h^3)) - y &= h(y' + \frac{h}{2}y'' + O(h^2)) + \tau_n \\ O(h^3) &= \tau_n \end{aligned}$$

The reason that the above solution doesn't quite work is we lost track of the indices. That is, we didn't really keep track of where we evaluate  $y$ , and where we evaluate  $f$ . On closer inspection, we see that

$$\begin{aligned} y_{n+1} - y_n &= hy'_n + \frac{h^2}{2}y''_n + O(h^3) \\ &= hf(t_n, y_n) + \frac{h^2}{2}(f_t(t_n, y_n) + Df(t_n, y_n) \cdot f(t_n, y_n)) + O(h^3) \end{aligned}$$

And that

$$\begin{aligned}
& f\left(t_n + \frac{h}{2}, y_n + \frac{y_{n+1} - y_n}{2}\right) \\
&= f(t_n, y_n) + \frac{h}{2} f_t\left(t_n, y_n + \frac{y_{n+1} - y_n}{2}\right) \\
&+ Df\left(t_n + \frac{h}{2}, y_n\right) \cdot \left(\frac{y_{n+1} - y_n}{2}\right) + O(h^2) + O\left(\frac{y_{n+1} - y_n}{2}\right) \\
&= f(t_n, y_n) + \frac{h}{2} f_t(t_n, y_n) + \frac{h}{2} Df_t(t_n, y_n) \cdot \left(\frac{y_{n+1} - y_n}{2}\right) + Df(t_n, y_n) \cdot \left(\frac{y_{n+1} - y_n}{2}\right) \\
&+ \frac{h}{2} Df_t(t_n, y_n) \cdot \left(\frac{y_{n+1} - y_n}{2}\right) + O(h^2) + O\left(\frac{y_{n+1} - y_n}{2}\right) + O\left(h \frac{y_{n+1} - y_n}{2}\right)
\end{aligned}$$

(as you can see, the problem here is that we end up evaluating  $f_t$  and  $Df$  at the wrong points.) We can substitute using  $\frac{y_{n+1} - y_n}{2} = \frac{hf(t_n + \frac{h}{2}, y_n + \frac{y_{n+1} - y_n}{2})}{2}$ , and continue chugging away. We can continue the problem this way, and (after sufficient algebra) crank out a workable solution. However, the problem can be done more easily by noting the following:

$$y_n + \frac{y_{n+1} - y_n}{2} = \frac{y_{n+1} + y_n}{2} = \text{an approximation to } y_{n+\frac{1}{2}}$$

This strongly suggests that we expand our expressions around  $y_{n+\frac{1}{2}}$  instead of  $y_n$ .

$$\begin{aligned}
y_{n+1} &= y_{n+\frac{1}{2}+\frac{1}{2}} = y_{n+\frac{1}{2}} + \frac{h}{2} y'_{n+\frac{1}{2}} + \frac{h^2}{8} y''_{n+\frac{1}{2}} + O(h^3) \\
y_n &= y_{n+\frac{1}{2}-\frac{1}{2}} = y_{n+\frac{1}{2}} - \frac{h}{2} y'_{n+\frac{1}{2}} + \frac{h^2}{8} y''_{n+\frac{1}{2}} + O(h^3) \\
y_{n+1} - y_n &= h y'_{n+\frac{1}{2}} + O(h^3)
\end{aligned}$$

(So in particular, the difference quotient  $\frac{y_{n+1} - y_n}{h}$  is an  $O(h^2)$  approximation to the derivative at the midpoint).

$$\frac{y_{n+1} + y_n}{2} = \frac{2y_{n+\frac{1}{2}} + \frac{h^2}{4} y''_{n+\frac{1}{2}} + O(h^3)}{2} = y_{n+\frac{1}{2}} + \frac{h^2}{8} y''_{n+\frac{1}{2}} + O(h^3)$$

(So in particular, the average  $\frac{y_{n+1} + y_n}{2}$  is an  $O(h^2)$  approximation to the midpoint).

$$\begin{aligned}
f\left(t_n + \frac{h}{2}, y_n + \frac{y_{n+1} - y_n}{2}\right) &= f\left(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}} + O(h^2)\right) \\
&= f\left(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}\right) + Df\left(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}\right) \cdot O(h^2)
\end{aligned}$$

And so now we can put it all together to get:

$$\begin{aligned}
 y_{n+1} - y_n &= hf(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}} + O(h^2)) + \tau_n \\
 hy'_{n+\frac{1}{2}} + O(h^3) &= h(f(t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + O(h^2)) + \tau_n \\
 &= hy'_{n+\frac{1}{2}} + O(h^3) + \tau_n \\
 \tau_n &= O(h^3)
 \end{aligned}$$

8. we investigate the instability of this 2-step method

$$u_{n+1} + 4u_n - 5u_{n-1} = h(4f_n + 2f_{n-1}).$$

What does it mean for a method to be ‘unstable’? Usually, this means that the method is ‘zero-unstable’, which means that minor (arbitrarily small but nonzero) perturbations (in initial conditions, computation of intermediate values, or the differential system) give rise to arbitrarily large (unbounded) differences in the computed numerical solution.

For now, let’s assume we are solving the simplest of differential systems, namely  $y' = 0$ . This has the exact solution  $y(t) = y_0$  (a constant, say  $y_0 = 1$ ). If we try and apply our numerical method to this problem, we get

$$u_{n+1} = 5u_{n-1} - 4u_n$$

Now if  $u_0 = u_1 = 1$ , then  $u_2$  will equal 1, and  $u_3$  will equal 1 and so on. (our numerical solution will be equal to the exact solution). However, if  $u_0 = 1$  but  $u_1 = 1 + \epsilon$ , then we will get

$$\begin{aligned}
 u_2 &= 5 - 4(1 + \epsilon) = 1 - 4\epsilon \\
 u_3 &= 5(1 + \epsilon) - 4(1 - 4\epsilon) = 1 + 21\epsilon \\
 u_4 &= 5(1 - 4\epsilon) - 4(1 + 21\epsilon) = 1 - 104\epsilon \\
 u_5 &= 5(1 + 21\epsilon) - 4(1 - 104\epsilon) = 1 + 521\epsilon
 \end{aligned}$$

and so the errors grow (very quickly).

To analyze this more precisely, we note that the difference equation

$$u_{n+1} + 4u_n - 5u_{n-1} = 0$$

has characteristic polynomial  $r^2 + 4r - 5$ , with roots  $r = 1$  and  $r = -5$ . Hence the general solution is

$$u_n = \alpha 1^n + \beta (-5)^n$$

So if  $u_0 = 1$  and  $u_1 = 1 + \epsilon$ , then

$$\begin{aligned}1 &= \alpha + \beta, \quad 1 + \epsilon = \alpha - 5\beta \\ \beta &= \frac{-\epsilon}{6}, \quad \alpha = 1 + \frac{\epsilon}{6} \\ u_n &= \left(1 + \frac{\epsilon}{6}\right) - \frac{\epsilon}{6}(-5)^n\end{aligned}$$

So as  $h \rightarrow 0$  and  $n \rightarrow \infty$ ,  $u_n \rightarrow \infty$ .

This of course assumes exact arithmetic at each step. Things get a lot more complicated when there are rounding errors present. (try to apply the method to  $y' = 0$ ,  $y_0 = \sqrt{2}$  with perturbed initial conditions).