

Math 4650 Sample Final

1. Suppose you are given the method

$$w_{i+1} = w_{i-1} + h(af(t_i, w_i) + bf(t_{i-1}, w_{i-1}))$$

for approximating the differential equation  $y'(t) = f(t, y(t))$ .

- (a) What should  $a$  and  $b$  be so that the local truncation error is as small as possible? What will the local truncation error be in that case?

**Solution:** To compute local truncation error, we assume  $w_i = y(t_i)$  and  $w_{i-1} = y(t_{i-1})$ . Then  $f(t_i, w_i) = f(t_i, y(t_i)) = y'(t_i)$  and  $f(t_{i-1}, w_{i-1}) = f(t_{i-1}, y(t_{i-1})) = y'(t_{i-1})$ . So the approximation with these values will be

$$\begin{aligned} w_{i+1} &= y(t_{i-1}) + ah y'(t_i) + bh y'(t_{i-1}) \\ &= y(t_i - h) + ah y'(t_i) + bh y'(t_i - h). \end{aligned}$$

Now we want to expand  $w_{i+1}$  in powers of  $h$ , so that we can easily compare it to  $y(t_{i+1})$ .

$$\begin{aligned} w_{i+1} &= y(t_i) - hy'(t_i) + \frac{h^2}{2}y''(t_i) - \frac{h^3}{6}y'''(t_i) + \dots \\ &\quad + ah y'(t_i) + bh \left( y'(t_i) - hy''(t_i) + \frac{h^2}{2}y'''(t_i) + \dots \right) \\ &= y(t_i) + h(a + b - 1)y'(t_i) + h^2\left(\frac{1}{2} - b\right)y''(t_i) + h^3\left(\frac{b}{2} - \frac{1}{6}\right)y'''(t_i) + \dots \end{aligned}$$

Finally we look at the expansion of  $y(t_{i+1})$ , the exact solution.

$$y(t_{i+1}) = y(t_i + h) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \frac{h^3}{6}y'''(t_i) + \dots$$

Now we want  $w_{i+1}$  and  $y(t_{i+1})$  to be as close as possible for small  $h$ , which means we want to match up as many powers of  $h$  as possible. The  $y(t_i)$  term matches up automatically. Matching the  $y'(t_i)$  term, we get the equation  $a + b - 1 = 1$ . Matching up the  $y''(t_i)$  term, we get the equation  $\frac{1}{2} - b = \frac{1}{2}$ .

These equations imply that  $b = 0$  and  $a = 2$ . So the method is

$$w_{i+1} = w_{i-1} + 2hf(t_i, w_i).$$

As we derived above, when we assume  $w_i = y(t_i)$  and  $w_{i-1} = y(t_{i-1})$ , we get

$$w_{i+1} = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) - \frac{h^3}{6}y'''(t_i) + \dots$$

So the local truncation error is

$$\begin{aligned}
 \tau_{i+1} &= \frac{1}{h} (y(t_{i+1}) - w_{i+1}) \\
 &= \frac{1}{h} \left( (y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \frac{h^3}{6}y'''(t_i) + \dots) \right. \\
 &\quad \left. - (y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) - \frac{h^3}{6}y'''(t_i) + \dots) \right) \\
 &= \frac{1}{h} \left( \frac{h^3}{6}y'''(t_i) + \frac{h^3}{6}y'''(t_i) + \dots \right) \\
 &= \frac{h^2}{3}y'''(t_i) + \dots \\
 &= \frac{h^2}{3}y'''(\xi_i),
 \end{aligned}$$

where the replacement of  $t_i$  with some unknown  $\xi_i$  allows us to make the equation exact.

(b) Is the method stable? Strongly stable?

**Solution:** To test for stability, we set  $f = 0$  (i.e., we test on the simplest possible differential equation  $y' = 0$ ). So the method is  $w_{i+1} = w_{i-1}$ . To solve this difference equation, we try a solution of the form  $w_i = \lambda^i$ . Then  $\lambda^{i+1} = \lambda^{i-1}$ , so that  $\lambda^{i-1}(\lambda^2 - 1) = 0$ . Thus the characteristic equation is

$$\lambda^2 - 1 = 0,$$

with solutions  $\lambda = 1$  and  $\lambda = -1$ .

There are two roots with  $|\lambda| = 1$ , but they both have multiplicity one. So the method is stable (it satisfies the root condition), but not strongly stable ( $\lambda = 1$  is not the only root of size 1).

(c) Is the method convergent? How do you know?

**Solution:** A method is convergent if and only if it is both stable and consistent. Since the local truncation error is  $O(h^2)$ , the method is consistent; since it satisfies the root condition, the method is stable. Therefore it is convergent.

2. Use Romberg integration to estimate  $\int_0^2 x^2 dx$ . How large does  $k$  have to be before  $R_{k,k}$  is the exact answer?

**Solution:** The exact answer is obviously  $\frac{8}{3}$ .

Recall that the first column of the Romberg matrix is the various trapezoid rules, computed inductively by

$$R_{1,1} = \frac{h_1}{2} [f(a) + f(b)], \quad R_{k,1} = \frac{1}{2} \left[ R_{k-1,1} + h_{k-1} \sum_{i=1}^{2^{k-2}} f(a + (2i-1)h_k) \right],$$

where  $h_k = \frac{b-a}{2^{k-1}}$ .

The other columns are constructed from the first column using the error-reduction formula

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}.$$

So we have  $h_1 = 2$  and

$$R_{1,1} = \frac{2}{2}(0^2 + 2^2) = 4.$$

Then we get  $h_2 = 1$  and

$$R_{2,1} = \frac{1}{2}[R_{1,1} + h_1 f(a + h_2)] = \frac{1}{2}(4 + 2f(1)) = 3.$$

Now we get

$$R_{2,2} = R_{2,1} + \frac{R_{2,1} - R_{1,1}}{4 - 1} = 3 + \frac{3 - 4}{3} = \frac{8}{3}.$$

So already at two steps, we get the exact answer. This shouldn't be too surprising: we know that  $R_{2,2}$  is  $O(h^4)$ , and it gets that way by being exact on polynomials up to second-order.

3. For the differential equation  $y' = 2y - 4t$  with  $y(0) = 1$ , verify that the exact solution is  $y(t) = 2t + 1$ .

**Solution:** We check that  $y(0) = 2 \cdot 0 + 1 = 1$ , and that  $y'(t) = 2$ . Also  $2y - 4t = 2(2t + 1) - 4t = 2$ . So the differential equation and initial condition are both satisfied.

- (a) Use Euler's method with step size  $h = \frac{1}{2}$  to estimate  $y(1)$ .

**Solution:** We have

$$w_{i+1} = w_i + hf(t_i, w_i) = (1 + 2h)w_i - 4ht_i.$$

When  $h = \frac{1}{2}$ , we get  $w_{i+1} = 2w_i - 2t_i$ . Therefore

$$w_1 = 2w_0 - 2t_0 = 2,$$

and

$$w_2 = 2(2) - 2(\frac{1}{2}) = 3.$$

Euler's method happens to be exact, since  $y''(\xi) \equiv 0$ .

- (b) Use Heun's method with step size  $h = 1$  to estimate  $y(1)$ .

**Solution:** Heun's method in general is

$$w_{i+1} = w_i + \frac{h}{4}[f(t_i, w_i) + 3f(t_i + \frac{2}{3}h, w_i + \frac{2}{3}hf(t_i, w_i))]$$

We have  $w_i + \frac{2h}{3}f(t_i, w_i) = (1 + \frac{4h}{3})w_i - \frac{8h}{3}t_i$ . Thus

$$\begin{aligned} f(t_i + \frac{2}{3}h, w_i + \frac{2}{3}hf(t_i, w_i)) &= 2((1 + \frac{4h}{3})w_i - \frac{8h}{3}t_i) - 4(t_i + \frac{2h}{3}) \\ &= (2 + \frac{8h}{3})w_i + (-4 - \frac{16h}{3})t_i - \frac{8h}{3}. \end{aligned}$$

So finally we have

$$\begin{aligned}w_{i+1} &= w_i + \frac{h}{4}(2w_i - 4t_i) + \frac{3h}{4}\left((2 + \frac{8h}{3})w_i + (-4 - \frac{16h}{3})t_i - \frac{8h}{3}\right) \\ &= (1 + 2h + 2h^2)w_i + (-4h - 4h^2)t_i - 2h^2.\end{aligned}$$

In particular, when  $h = 1$  we get

$$w_{i+1} = 5w_i - 8t_i - 2.$$

Thus we have  $w_1 = 5 - 0 - 2 = 3$ . Again the method is exact, which makes sense since  $y'''(\xi) \equiv 0$ .

- (c) Use the second-order Taylor method with step size  $h = 1$  to estimate  $y(1)$ .

**Solution:** The second-order Taylor method is obtained from the approximation

$$y(t+h) \approx y(t) + hy'(t) + \frac{h^2}{2}y''(t).$$

In this case, we have  $y'(t) = 2y - 4t$  and so

$$y''(t) = 2y'(t) - 4 = 4y(t) - 8t - 4.$$

The method is therefore

$$w_{i+1} = w_i + h(2w_i - 4t_i) + h^2(2w_i - 4t_i - 2) = (1 + 2h + 2h^2)w_i - (4h + 4h^2)t_i - 2h^2.$$

In fact it is identical to Heun's method in this case. So the result is  $w_1 = 3$ , which is again exact.

Which method comes closest to the exact solution?

**Solution:** All the methods are exact since the solution is just a linear function of time.

4. (a) Suppose  $y'(t) = f(t, y)$  is some differential equation with initial condition  $y(0) = \alpha$  given. If Euler's method with step size  $h$  is used to get  $w_1$ , an approximation to  $y(h)$ , how far off do you expect it to be?

**Solution:** The general error formula for one step of Euler's method is

$$y(h) - w_1 = \frac{h^2}{2}y''(\xi_1).$$

- (b) Now suppose Euler's method with step size  $h/2$  is used to get  $\tilde{w}_2$ , also an approximation to  $y(h)$ . What error do you expect using this method?

**Solution:** We know that since  $\tilde{w}_0 = y(0)$ , we have

$$\tilde{w}_1 = y\left(\frac{h}{2}\right) - \frac{(h/2)^2}{2}y''(\tilde{\xi}_1).$$

If  $\tilde{w}_1$  were exactly equal to  $y\left(\frac{h}{2}\right)$ , we would have by the local truncation error formula that

$$\tilde{w}_2 = y(h) - \frac{(h/2)^2}{2}y''(\tilde{\xi}_2).$$

But it's not, and therefore the total error is the sum of the two errors (up to higher-order terms):

$$\tilde{w}_2 - y(h) = -\frac{h^2}{8}y''(\tilde{\xi}_1) - \frac{h^2}{8}y''(\tilde{\xi}_2).$$

- (c) Show how to estimate the true value of  $y(h)$  from the computed values  $w_1$  and  $\tilde{w}_2$ .

**Solution:** We now have two error formulas:

$$\tilde{w}_2 - y(h) = -\frac{h^2}{8}y''(\tilde{\xi}_1) - \frac{h^2}{8}y''(\tilde{\xi}_2)$$

and

$$y(h) - w_1 = \frac{h^2}{2}y''(\xi_1).$$

To be able to do anything with this, we need to assume that all  $\xi$ 's are the same:  $\xi_1 = \tilde{\xi}_1 = \tilde{\xi}_2$ . Then we have the formulas

$$y(h) = \tilde{w}_2 + \frac{h^2}{4}y''(\xi)$$

and

$$y(h) = w_1 + \frac{h^2}{2}y''(\xi),$$

and we can eliminate the  $y''(\xi)$  term by taking twice the first equation minus the second:

$$y(h) = 2\tilde{w}_2 - w_1.$$

The error in this approximation should be better than  $O(h^2)$ .

- (d) Explain briefly how you could use this to derive an adaptive Euler's method.

**Solution:** Since we have a higher-order approximation to the actual value, we can estimate the error in using step size  $h$  by

$$y(h) - w_1 = 2\tilde{w}_2 - 2w_1.$$

This is correct to within  $O(h^3)$ , which should be a small correction.

So we basically compute the new  $w_{i+1}$  at each step, cut the step size in half and compute  $\tilde{w}_{2i+2}$ , and finally compute the approximate error  $\varepsilon_i = 2|\tilde{w}_{2i+2} - w_{i+1}|$ . If  $\varepsilon_i$  is less than the desired tolerance, we move on to the next time step. If not, we cut  $h$  in half and compute again.

5. Give the  $LU$ -decomposition of the matrix

$$A = \begin{pmatrix} 1 & -2 & 2 \\ 2 & -1 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

Then solve the equation  $A\mathbf{x} = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}$  using the decomposition and back-substitution.

**Solution:**

We first reduce the matrix using Gaussian elimination, and we record the factors we used. Eliminating the first row using  $m_{21} = 2$  and  $m_{31} = -1$ , we get

$$A^{(2)} = \begin{pmatrix} 1 & -2 & 2 \\ 0 & 3 & -4 \\ 0 & -3 & 3 \end{pmatrix}$$

From the new matrix, we get  $m_{32} = -1$ , and the reduced matrix is

$$A^{(3)} = U = \begin{pmatrix} 1 & -2 & 2 \\ 0 & 3 & -4 \\ 0 & 0 & -1 \end{pmatrix}.$$

Therefore the  $LU$  decomposition is

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & -2 & 2 \\ 0 & 3 & -4 \\ 0 & 0 & -1 \end{pmatrix}.$$

To solve the equation  $Ax = b$ , we write  $Ly = b$  and  $Ux = y$ . The solution of

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}$$

can be read off as  $y_1 = 2$ ,  $y_2 = 1 - 2(2) = -3$ , and  $y_3 = -1 + 2 - 3 = -2$ . We then have to solve  $Ux = y$ , i.e.,

$$\begin{pmatrix} 1 & -2 & 2 \\ 0 & 3 & -4 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -3 \\ -2 \end{pmatrix};$$

again we can easily read off the solution as  $x_3 = 2$ ,  $x_2 = \frac{5}{3}$ , and  $x_1 = \frac{4}{3}$ .

6. Suppose you want to solve an equation involving an  $n \times n$  matrix of bandwidth 4 which looks like the following:

$$\begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & \cdots \\ a_{21} & a_{22} & a_{23} & 0 & 0 & \cdots \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & \cdots \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & \cdots \\ 0 & 0 & a_{53} & a_{54} & a_{55} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ \vdots \end{pmatrix}$$

Write an algorithm, generalizing Crout's algorithm, to solve this system. How does the number of operations in your algorithm depend on  $n$ ?

**Solution:** To reduce the first column, we need to clear out two rows. The new matrix is

$$\begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0 & \cdots \\ 0 & a_{22}^{(2)} & a_{23} & 0 & 0 & \cdots \\ 0 & a_{32}^{(2)} & a_{33} & a_{34} & 0 & \cdots \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & \cdots \\ 0 & 0 & a_{53} & a_{54} & a_{55} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(2)} \\ b_4 \\ b_5 \\ \vdots \end{pmatrix}$$

The only changes we make are  $a_{22}^{(2)} = a_{22} - a_{12}a_{21}/a_{11}$ ,  $a_{32}^{(2)} = a_{32} - a_{12}a_{31}/a_{11}$ ,  $b_2^{(2)} = b_2 - b_1a_{21}/a_{11}$ , and  $b_3^{(2)} = b_3 - b_1a_{31}/a_{11}$ .

Similarly in reducing the next column, we will perform the computations  $a_{33}^{(3)} = a_{33} - a_{32}^{(2)}a_{23}/a_{22}^{(2)}$ ,  $a_{43}^{(3)} = a_{43} - a_{42}a_{23}/a_{22}^{(2)}$ ,  $b_3^{(3)} = b_3^{(2)} - b_2^{(2)}a_{32}^{(2)}/a_{22}^{(2)}$ , and  $b_4^{(3)} = b_4 - b_2^{(2)}a_{42}/a_{22}^{(2)}$ .

It's now easy to generalize this: for  $k$  starting at 1, we do

$$\begin{aligned} a[k+1,k+1] &= a[k+1,k+1] - a[k+1,k]*a[k,k+1]/a[k,k] \\ a[k+2,k+1] &= a[k+2,k+1] - a[k+2,k]*a[k,k+1]/a[k,k] \\ b[k+1] &= b[k+1] - b[k]*a[k+1,k]/a[k,k] \\ b[k+2] &= b[k+2] - b[k]*a[k+2,k]/a[k,k] \end{aligned}$$

This loop continues up to  $k = n - 2$ ; then we only do two more computations when  $k = n - 1$  (since there will then be only one nonzero element below the diagonal, rather than two).

So the algorithm to reduce the matrix to upper-triangular form is (storing the values  $a[k+1,k]/a[k,k]$  which get used twice, for efficiency)

```
for k from 1 to n-2 do
    m1 = a[k+1,k]/a[k,k]
    m2 = a[k+2,k]/a[k,k]
    a[k+1,k+1] = a[k+1,k+1] - m1*a[k,k+1]
    a[k+2,k+1] = a[k+2,k+1] - m2*a[k,k+1]
    b[k+1] = b[k+1] - m1*b[k]
    b[k+2] = b[k+2] - m2*b[k]
end do
m1 = a[n,n-1]/a[n-1,n-1]
a[n,n] = a[n,n] - m1*a[n-1,n]
b[n] = b[n] - m1*b[n-1]
```

The number of multiplications and divisions in this algorithm is six for each  $k$ -step, plus three more at the end, so  $6(n-2)+3 = 6n-9$ . Similarly there are  $4(n-2)+2 = 4n-6$  additions/subtractions.

To do the backward-substitution, we observe that the reduced matrix now only has nonzero elements one over from the diagonal. That means the second part of the algorithm is exactly the same as for Crout's method.

```

x[n] = b[n]/a[n,n]
for j from 1 to n-1 do
  x[n-j] = (b[n-j] - a[n-j,n-j+1]*x[n-j+1])/a[n-j,n-j]
end do

```

There are two multiplications/divisions each time through the loop, and one at the beginning, so the total here is  $2(n-1) + 1 = 2n - 1$ . Similarly there are  $n - 1$  additions/subtractions total through this loop.

Adding the two results together, we see that this method requires

$$6n - 9 + 2n - 1 = 8n - 10 \text{ mults/divs and } 4n - 6 + n - 1 = 5n - 7 \text{ adds/subs.}$$

7. The STORMEY algorithm (Second/Third Order Runge-Moulton Egg Yolks) is defined as follows:

$$w_{i+1} = w_i + \frac{h}{4}f(t_i + h, w_{i+1}) + \frac{3h}{4}f(t_i + \frac{h}{3}, w_i + \frac{h}{3}f(t_i, w_i)).$$

For the differential equation  $y'(t) = 2y(t) - t$ , work out what precisely the STORMEY algorithm does. For  $h = 1$  and  $y(0) = 1$ , what is the estimate for  $y(2)$ ?

**Solution:** We first compute  $w_i + \frac{h}{3}f(t_i, w_i) = (1 + \frac{2h}{3})w_i - \frac{h}{3}t_i$ . Plugging this in, we get

$$\begin{aligned} f(t_i + \frac{h}{3}, w_i + \frac{h}{3}f(t_i, w_i)) &= 2((1 + \frac{2h}{3})w_i - \frac{h}{3}t_i) - (t_i + \frac{h}{3}) \\ &= (2 + \frac{4h}{3})w_i - (1 + \frac{2h}{3})t_i - \frac{h}{3}. \end{aligned}$$

The full algorithm is therefore

$$\begin{aligned} w_{i+1} &= w_i + \frac{h}{4}(2w_{i+1} - t_i - h) + \frac{3h}{4}((2 + \frac{4h}{3})w_i - (1 + \frac{2h}{3})t_i - \frac{h}{3}) \\ w_{i+1} &= w_i + \frac{h}{2}w_{i+1} - \frac{ht_i}{4} - \frac{h^2}{4} + (\frac{3h}{2} + h^2)w_i - (\frac{3h}{4} + \frac{h^2}{2})t_i - \frac{h^2}{4} \\ (1 - \frac{h}{2})w_{i+1} &= (1 + \frac{3h}{2} + h^2)w_i - (h + \frac{h^2}{2})t_i - \frac{h^2}{2}. \end{aligned}$$

Finally, solving for  $w_{i+1}$  to make the method explicit, we get

$$w_{i+1} = \frac{(1 + \frac{3h}{2} + h^2)w_i - (h + \frac{h^2}{2})t_i - \frac{h^2}{2}}{1 - \frac{h}{2}}.$$

When  $h = 1$  as in the problem, we have

$$w_{i+1} = \frac{\frac{7}{2}w_i - \frac{3}{2}t_i - \frac{1}{2}}{\frac{1}{2}} = 7w_i - 3t_i - 1.$$

Thus with  $w_0 = 1$ , we get  $w_1 = 7 - 1 = 6$  and  $w_2 = 42 - 3 - 1 = 38$ .