**Math 128a - Final - Spring 2002**

This exam is open book, open notes, open calculator (you shouldn't need one). The total score is 130 points. The number of points approximately indicates the number of minutes you should spend on the problem.

1) (35 points) In this problem we explore how ODE solvers are designed.

**Part A.** (15 points) Use the method of undetermined coefficients to derive an Adams-Moulton method of the form

$$x_{n+1} = x_n + h * [A \cdot f_{n+1} + B \cdot f_n + C \cdot f_{n-1}]$$

Here the notation is that $t_n = n * h, x_n = x(t_n)$ and $f_n = f(t_n, x_n)$. Compute the values of $A$, $B$ and $C$; show your work. What is the value of $k$ such that the LTE $= O(h^k)$?

**Answer:** As we saw in the text and in the homework, one way to determine the coefficients $A$, $B$, and $C$ is to demand that the approximate integration formula

$$\int_0^1 f(t, x(t)) \approx Af(1, x(1)) + Bf(0, x(0)) + Cf(-1, x(-1))$$

(which arises in the derivation of multistep methods) is exact when $f(t, x(t))$ is a polynomial of degree at most 2. Thus, we define the three polynomials

$$p_0(t) = 1, \quad p_1(t) = t - 1, \quad p_2(t) = (t - 1)t,$$

and require that for all $n$,

$$\int_0^1 p_n(t)\, dt = Ap_n(1) + Bp_n(0) + Cp_n(-1).$$

This gives us the three equations

$$\begin{aligned} A + B + C &= 1, \\ -B - 2C &= -1/2, \\ 2C &= -1/6. \end{aligned}$$

Therefore, $C = -1/12$, $B = 2/3$, and $A = 5/12$, so the formula is

$$x_{n+1} = x_n + \frac{h}{12}[5f_{n+1} + 8f_n - f_{n-1}].$$

From the lecture notes, we know that this Adams-Moulton formula will have LTE $O(h^4)$, so this is a third-order method.

**Part B.** (10 points) Use the method of undetermined coefficients to derive an Adams-Bashforth method of the form

$$x_{n+1} = x_n + h * [D \cdot f_n + E \cdot f_{n-1}]$$

Compute the values of $D$ and $E$; show your work. What is the value of $k$ such that the LTE $= O(h^k)$?

**Answer:** We proceed as above. We choose $D$ and $E$ such that the approximate integration formula

$$\int_0^1 f(t, x(t)) \approx Df(0, x(0)) + Ef(-1, x(-1))$$

(which arises in the derivation of multistep methods) is exact when $f(t, x(t))$ is a polynomial of degree at most 1. Thus, we define the two polynomials

$$p_0(t) = 1, \quad p_1(t) = t$$

and require that for all $n$,

$$\int_0^1 p_n(t)\, dt = Dp_n(0) + Ep_n(-1).$$

This gives us the two equations

$$\begin{aligned} D + E &= 1, \\ -E &= 1/2, \end{aligned}$$

Therefore, $E = -1/2$, $D = 3/2$, so the formula is

$$x_{n+1} = x_n + \frac{h}{2}[3f_n - f_{n-1}].$$

From the lecture notes, we know that this Adams-Bashforth formula will have LTE $O(h^3)$, so this is a third-order method.

**Part C.** (10 points) Describe an algorithm (with pseudocode) that uses the methods in Part A and B with fixed step size $h$ to solve the ODE $x'(t) = f(t, x(t))$, starting at $x(0) = x_0$ up to time $t_{final}$. You may assume that $t_{final}$ is an integer multiple of $h$. Make sure to describe how to monitor the LTE (but not to change $h$).

**Answer:** At the first step, when $n = 0$ and we want to find $x_1$, $f_{-1}$ is unknown because $x_{-1}$ is unknown. Therefore, we cannot use the formula from Part A. Instead, we initialize the procedure with a third-order Runge-Kutta method, denoted RK3 in the pseudocode below. (If we wished to monitor the LTE at this initialization step, we would want to use a second-order Runge-Kutta method as well; if we denoted this second-order method as RK2, then the LTE at this first step would be smaller than $|\text{RK2}(f, x_0, h) - \text{RK3}(f, x_0, h)|$.)

We then predict $x_{n+1}$ with the Adams-Bashforth method from Part B, use this to predict $f_{n+1}$, and plug this into the formula from Part A to correct our guess of $x_{n+1}$. This formula from Part B will also allow us to monitor the LTE. We are choosing to be conservative with our LTE estimates by using second-order methods to predict the error while using third-order methods to generate the output.

Note that as the algorithm runs, we need only store two values of $f_i$.

Pseudocode for this algorithm appears below:

$x_1 = \text{RK3}(f, x_0, h)$
$f_0 = f(0, x_0)$
$f_1 = f(h, x_1)$
for $i = 1$ to $f_{final}/h - 1$
$\quad xp_{i+1} = x_i + \frac{h}{2}[3f_i - f_{i-1}]$
$\quad f_{i+1} = f((i+1)h, xp_{i+1})$
$\quad x_{i+1} = x_i + \frac{h}{12}[5f_{i+1} + 8f_i - f_{i-1}]$
$\quad \text{LTE} = xp_{i+1} - x_{i+1}$
$\quad f_{i+1} = f((i+1)h, x_{i+1})$
end

2) (35 points) In this problem we explore how to efficiently solve linear systems of equations $Ax = b$, when $A$ is *banded*, i.e. only has nonzero entries near the diagonal. We say that $A$ has *lower bandwidth lbw* if $a_{ij} = 0$ whenever $i > j + lbw$, and that $A$ has *upper bandwidth ubw* if $a_{ij} = 0$ whenever $j > i + ubw$. For example, the 8-by-8 matrix below has $lbw = 2$ and $ubw = 3$.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & 0 & 0 \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} \\ 0 & 0 & 0 & 0 & a_{75} & a_{76} & a_{77} & a_{78} \\ 0 & 0 & 0 & 0 & 0 & a_{86} & a_{87} & a_{88} \end{bmatrix}$$

We call the part of the $A$ that may be nonzero the *band* of $A$.

**Part A.** (10 points) Assume the $n$-by-$n$ matrix band matrix $A$ with lower bandwidth $lbw$ and upper bandwidth $ubw$ is stored in memory exactly as shown above. Give pseudocode for Gaussian elimination with no pivoting (GENP) that performs no arithmetic on the zero entries outside the band. (Your code should just compute the entries of $L$ and $U$ so that $A = L * U$).

**Answer:** The observation is that neither assignment statement in GENP creates a nonzero entry outside the original band of $A$. We take the GENP code, and only have to change the limits on the loops:

```
for i = 1 to n - 1
    for j = i + 1 to min(n, i + lbw)
        A(i, j) = A(i, j)/A(i, i)
    for j = i + 1 to min(n, i + lbw)
        for k = i + 1 to min(n, i + ubw)
            A(j, k) = A(j, k) - A(j, i) * A(i, k)
```

**Part B.** (5 points) Considering $L$ from Part A as a band matrix, what are its lower and upper bandwidths? Considering $U$ from Part A as a band matrix, what are its lower and upper bandwidths?

**Answer:** $L$ is lower triangular, and so has upper bandwidth 0. $L$ has no nonzero entries outside the original band of $A$ and so has the same lower bandwidth as $A$, namely $lbw$.
$U$ is upper triangular, and so has lower bandwidth 0. $U$ has no nonzero entries outside the original band of $A$ and so has the same upper bandwidth as $A$, namely $ubw$.

**Part C.** (10 points) Band matrices are used when $lbw$ and $ubw$ are both much smaller than $n$, because the algorithm in Part A does much less work than plain Gaussian elimination. How many arithmetic operations does your algorithm from Part A do? Count additions, subtractions, multiplications and divisions each as 1 operation. Give your answer in the

form $c_1 \cdot lbw \cdot ubw \cdot n + c_2 \cdot lbw \cdot n + c_3 \cdot ubw \cdot n + O(1)$, where you supply the constants $c_1$, $c_2$ and $c_3$. The $O(1)$ term is independent of $n$, but can depend on $lbw$ and $ubw$, which we are assuming are small. Show how you determined these constants.

**Answer:** The first loop (to compute $A(i,j)$) does $lbw$ divisions as long as $i \leq n - lbw$, and fewer after that, for an exact count of $lbw \cdot (n - lbw) + lbw \cdot (lbw - 1)/2 = lbw \cdot n + O(1)$. In fact $lbw \cdot n$ is an upper bound.

As in GENP, nearly all the work is done by the double loop over $j$ and $k$. Except near the end of the matrix (when $i$ is close to $n$) the loops go from $j = i + 1$ to $i + lbw$ and from $k = i + 1$ to $i + ubw$, or $lbw \cdot ubw$ iterations in all. Since we do 1 multiplication and 1 add in the last line, the work is bounded above by $2 \cdot lbw \cdot ubw \cdot n$. A lower bound is $2 \cdot lbw \cdot ubw \cdot (n - \max(lbw, ubw))$, which differs from the upper bound by something depending only on $lbw$ and $ubw$, or $O(1)$. Thus the number of operations can be written $2 \cdot lbw \cdot ubw \cdot n + O(1)$.

In total, the number of operations is bounded by $2 \cdot lbw \cdot ubw \cdot n + 1 \cdot lbw \cdot n + O(1)$ (in fact, omitting the $O(1)$ yields an upper bound). Thus $c_1 = 2$, $c_2 = 1$ and $c_3 = 0$.

**Part D.** (5 points) Give pseudocode for solving $Ax = b$ using the $L$ and $U$ factors computed from Part A, doing no arithmetic on the zero entries of $L$ and $U$.

**Answer:**

```
Solving Ly = b for y
    for i = 1 to n
        y(i) = b(i)
        for j = max(i - lbw, 1) to i - 1
            y(i) = y(i) - L(i,j) * y(j)
        y(i) = y(i)/L(i,i)

Solving Ux = y for x = A^{-1}b
    for i = n down to 1
        x(i) = y(i)
        for j = i + 1 to min(n, i + ubw)
            x(i) = x(i) - U(i,j) * x(j)
        x(i) = x(i)/U(i,i)
```

**Part E.** (5 points) How many arithmetic operations does your algorithm from Part D do? Follow the same advice as for part C. Give your answer in the form $d \cdot lbw \cdot n + e \cdot ubw \cdot n + f \cdot n + O(1)$, where you supply the constants $d$, $e$ and $f$. Show how you determined these constants.

**Answer:** For solving $Ly = b$, the work is $2 \cdot lbw \cdot n + n + O(1)$. For solving $Ux = y$, the work is $2 \cdot ubw \cdot n + n + O(1)$. In total, the work is $2 \cdot lbw \cdot n + 2 \cdot ubw \cdot n + 2 \cdot n + O(1)$, so $d = e = f = 2$.

3) (35 points) In this problem we investigate the accuracy of ODE solvers. Consider the implicit second order integration formula for $x'(t) = f(x(t))$: $x_{n+1} = x_n + hf(x_{n+1})$, $h > 0$, where $x_n$ is the approximate solution of the ODE at $t = h \cdot n$. Consider applying this formula to the differential equation $x'(t) = \mu x(t)$, where $\mu$ is a constant and $x(0) \neq 0$ is given. $\mu$ may be any complex number $\mu = \mu_r + i \cdot \mu_i$, where $i = \sqrt{-1}$ and $\mu_r$ and $\mu_i$ are real.

**Part A.** (5 points.) Write down an explicit expression for $x_n$ (the numerical solution from the formula) in terms of $x_0 = x(0)$, $n$, $h$ and $\mu$.

**Answer:** $x_n = x_0/(1 - h\mu)^n$.

**Part B.** (5 points.) Write down an explicit expression for $x(t)$ (the true solution) in terms of $x(0)$, $\mu$ and $t$.

**Answer:** $x(t) = e^{\mu t}x(0)$.

**Part C.** (5 points.) Under what conditions on $\mu$ does $\lim_{t\to\infty} |x(t)| = 0$ for any $x(0) \neq 0$?

**Answer:** $|x(t)| = |e^{\mu t}x(0)| = e^{\mu_r t}|x(0)| \to 0$ as $t \to \infty$ if and only if $\mu_r < 0$.

**Part D.** (5 points.) Under what conditions on $\mu$ does $\lim_{t\to\infty} |x(t)| = \infty$ for any $x(0) \neq 0$?

**Answer:** $|x(t)| = |e^{\mu t}x(0)| = e^{\mu_r t}|x(0)| \to \infty$ as $t \to \infty$ if and only if $\mu_r > 0$.

**Part E.** (5 points.) Under what conditions on $\mu$ and $h$ does $\lim_{n\to\infty} |x_n| = 0$ for any $x_0 \neq 0$? Give your answer in the form "The limit is 0 if and only if the complex number $\mu \cdot h$ lies in region $C$ of the complex plane, where $C$ is precisely described as follows ..."

**Answer:** $|x_n| = |x_0|/|1 - \mu \cdot h|^n \to 0$ as $n \to \infty$ if and only if $|1 - \mu \cdot h| > 1$, i.e. the distance from 1 to $\mu \cdot h$ exceeds 1, or $\mu \cdot h$ lies outside a circle centered at 1 with radius 1 in the complex plane. Thus $C$ is the exterior of this circle.

**Part F.** (5 points.) Under what conditions on $\mu$ and $h$ does $\lim_{n\to\infty} |x_n| = \infty$ for any $x_0 \neq 0$? Give your answer in the form "The limit is infinite if and only if the complex number $\mu \cdot h$ lies in region $D$ of the complex plane, where $D$ is precisely described as follows ..."

**Answer:** $|x_n| = |x_0|/|1 - \mu \cdot h|^n \to \infty$ as $n \to \infty$ if and only if $|1 - \mu \cdot h| < 1$, i.e. the distance from 1 to $\mu \cdot h$ is less than 1, or $\mu \cdot h$ lies inside a circle centered at 1 with radius 1 in the complex plane. Thus $D$ is the interior of this circle.

**Part G.** (5 points.) Assume $x(0) = x_0 \neq 0$. Complete the following sentence and explain why it is true: "$\lim_{t\to\infty} |x(t)| = \lim_{n\to\infty} |x_n|$ if and only if the complex number $\mu \cdot h$ lies in region $E$ of the complex plane, where $E$ is precisely described as follows..."

**Answer:** From earlier parts,

- $\lim_{t\to\infty} |x(t)| = 0$ if $\mu_r < 0$, or equivalently $\mu \cdot h$ is in the open left half plane;
- $\lim_{t\to\infty} |x(t)| = \infty$ if $\mu_r > 0$, or equivalently $\mu \cdot h$ is in the open right half plane;
- $|x(t)| = |e^{i\mu_i}x(0)| = |x(0)|$ so $\lim_{t\to\infty} |x(t)| = |x(0)|$ if $\mu_r = 0$, or equivalently $\mu \cdot h$ is pure imaginary.

Also from earlier parts

- $\lim_{n\to\infty} |x_n| = 0$ if $\mu \cdot h$ is outside the unit circle centered at 1;
- $\lim_{n\to\infty} |x_n| = \infty$ if $\mu \cdot h$ is inside the unit circle centered at 1;
- $|1 - \mu \cdot h| = 1$ so $|x_n| = |x_0/(1 - \mu \cdot h)^n| = |x_0|$ and $\lim_{n\to\infty} |x_n| = |x_0|$ when $\mu \cdot h$ is on the unit circle centered at 1.

Thus the three possible limiting values of $\lim_{t\to\infty} |x(t)|$ and $\lim_{n\to\infty} |x_n|$ are 0, $\infty$, and $|x_0| = |x(0)|$. The limits are the same when

- $\mu \cdot h$ is in the open left half plane (and the common limit is 0),
- $\mu \cdot h$ is inside the unit circle centered at 1 (and the common limit is $\infty$),
- $\mu \cdot h = 0$ (and the common limit is $|x(0)| = |x_0|$),

4) (25 points) In class we talked about *Least Squares Problems:* Let $\|r\|_2 = \sqrt{\sum_{i=1}^{n} r_i^2}$ be the length of the vector $r$. Then if $A$ is an $m$-by-$n$ matrix with $m > n$, $b$ is an $m$-by-1 vector, the vector $s$ that minimizes $\|A \cdot s - b\|_2$ is given by $s = (A^T A)^{-1} A^T b$.

We will use this fact to solve the following approximation problem: Suppose we are given $m$ points in $\mathbf{R}^3$: $(x_1, y_1, z_1)$, ... , $(x_m, y_m, z_m)$. Using this data, we want to find a simple function $f(\cdot, \cdot)$ of two variables such that $z_i \approx f(x_i, y_i)$, i.e. $f(x, y)$ is a good approximation of $z$ in the sense that $\sqrt{\sum_{i=1}^{m} (f(x_i, y_i) - z_i)^2}$ is minimized.

**Part A.** (10 points) Suppose we want $f$ to be a linear function: $f(x, y) = s_1 \cdot x + s_2 \cdot y + s_3$. For what matrix $A$ and vector $b$ is the solution given by

$$s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = (A^T A)^{-1} A^T b$$

**Answer:**

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & 1 \end{bmatrix}, \quad b = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}$$

**Part B.** (10 points) Suppose we want $f$ to be a quadratic function:

$$f(x, y) = s_1 \cdot x^2 + s_2 \cdot x \cdot y + s_3 \cdot y^2 + s_4 \cdot x + s_5 \cdot y + s_6$$

For what matrix $A$ and vector $b$ is the solution given by

$$s = \begin{bmatrix} s_1 \\ \vdots \\ s_6 \end{bmatrix} = (A^T A)^{-1} A^T b$$

**Answer:**

$$A = \begin{bmatrix} x_1^2 & x_1 \cdot y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 \cdot y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^2 & x_m \cdot y_m & y_m^2 & x_m & y_m & 1 \end{bmatrix}, \quad b = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}$$

**Part C.** (5 points) Suppose that you compute $z_i$ with the program

```
for i = 1 to m
      z_i = 37x_i^2 - 22x_i y_i + 18y_i + 10 + r_i
end
```

where $r_i$ is a random number in the range $[-1, 1]$. Suppose you then compute $A$, $b$ and $s$ as described in Part B. Give a guaranteed upper bound on the error $\|As - b\|_2$.

**Answer:** The choice $s_1 = 37$, $s_2 = -22$, $s_3 = 0$, $s_4 = 0$, $s_5 = 18$, $s_6 = 10$ makes $|f(x_i, y_i) - z_i| = |r_i| \leq 1$, so $\|As - b\|_2 \leq \sqrt{m}$. The precise solution of the least squares problem can only make the value of $\|As - b\|_2$ get smaller than $\sqrt{m}$.