

Notes on Numerical Analysis

Alejandro Cantarero

This set of notes covers topics that most commonly show up on the Numerical Analysis qualifying exam in the Mathematics department at UCLA. Each section covers a specific area of numerical analysis. At the end of each section, we provide a list of key theorems. If the proof of the theorem is directly relevant to the qualifying exam, it will be included in this document. Otherwise, we simply include the statement of the theorem. At the end of each section, we include a listing of problems on old qualifying exams that fall under the given topic.

Contents

1	Root Finding	3
1.1	Bisection Method	3
1.2	Fixed Point Iteration	3
1.3	Newton's Method	4
1.4	Solving Systems of Non-Linear Equations	4
1.5	Qual Problems	5
2	Interpolation Theory	5
2.1	Polynomial Interpolation	5
2.1.1	Lagrange Polynomials	5
2.1.2	Newton Divided Differences	5
2.1.3	Hermite Interpolation	6
2.2	Trigonometric Interpolation	6
2.3	Key Theorems	7
2.4	Qual Problems	7
3	Numerical Differentiation	7
3.1	Method of Undetermined Coefficients	8
3.2	Common Formulae	8
3.3	Richardson Extrapolation	9
3.4	Qual Problems	10
4	Numerical Integration	10
4.1	Newton-Cotes Formulas	10
4.2	Composite Integration	10
4.3	Gaussian Quadrature	10
4.4	Key Theorems	10
4.5	Qual Problems	10
5	Iterative Methods	11
5.1	Relaxation Methods	11
5.1.1	Key Theorems	11
5.2	Qual Problems	12
6	Direct Methods for Solving Linear Systems	12
6.1	LU Factorization	12
6.2	Tridiagonal Systems	12
6.3	Qual Problems	13
7	Approximation of Functions	13
7.1	Least Squares	13
7.2	Qual Problems	14
8	Numerical Methods for Ordinary Differential Equations	14
8.1	One Step Methods	15
8.1.1	Stability Analysis	15
8.1.2	Common Methods and Their Properties	15
8.2	Linear Multistep Methods	16
8.3	Predictor-Corrector Methods	17
8.4	Qual Problems	17

9	Finite Differences for PDEs	17
9.1	Order of Accuracy	18
9.1.1	Using Symbols	18
9.2	Stability Analysis	18
9.3	Well-Posedness of Equations	18
9.4	Hyperbolic PDEs	19
9.4.1	Common Methods	19
9.5	Parabolic PDEs	20
9.5.1	Common Methods	20
9.5.2	Variable Diffusion Coefficients	21
9.6	Key Theorems	21
9.7	Qual Problems	22
10	Finite Element Methods	22
10.1	Weak Formulations	22
10.2	Well-Posedness of the Weak Form	24
10.3	Formulating a Finite Element Approximation	25
10.4	The Stiffness Matrix	26
10.5	Error Estimates	26
10.6	Key Theorems	27
10.7	Qual Problems	28
11	References	28

1 Root Finding

In this section, we discuss methods used to find the roots of an equation

$$f(x) = 0. \quad (1.1)$$

The following definition will give us some information on how fast a rootfinding method converges.

Definition 1.1. A sequence of iterates, $\{x_n\}$ converges with order $p \geq 1$ to α if

$$|\alpha - x_{n+1}| \leq c|\alpha - x_n|^p \quad (1.2)$$

for some $c > 0$.

1.1 Bisection Method

We begin by assuming that $f(x)$ is continuous on $[a, b]$ and $f(a)f(b) < 0$ (this guarantees the existence of a solution to (1.1)). The bisection method will also require a tolerance, ϵ . The method is then given by

Algorithm 1 Bisection Method

Require: f, a, b, ϵ

```

repeat
   $c \leftarrow (a + b)/2$ 
3: if  $\text{sign}(f(b)) \text{sign}(f(c)) \leq 0$  then
   $a \leftarrow c$ 
  else
6:  $b \leftarrow c$ 
  end if
  until  $b - c \leq \epsilon$ 
9: return  $c$ 

```

Note that the bisection algorithm will always converge to a root in $[a, b]$. Let α be the root that the bisection method is converging to and let c_n be the n th value of c produced by the method. Then we have

$$|\alpha - c_n| \leq \frac{(b - a)}{2^n}. \quad (1.3)$$

Note that in definition 1.1, if $p = 1$ we can write

$$|\alpha - x_n| \leq c^n |\alpha - x_0|. \quad (1.4)$$

So the bisection method has linear convergence with rate $\frac{1}{2}$, given by c .

1.2 Fixed Point Iteration

Now we consider solving a problem of the form $x = g(x)$ where the fixed point iteration is given by

$$x_{n+1} = g(x_n). \quad (1.5)$$

Note that the rootfinding problem, equation (1.1), can be reformulated as a fixed point problem.

Lemma 1.1. (Existence) Let $g(x)$ be continuous on $[a, b]$ and assume that $g([a, b]) \subset [a, b]$. Then $x = g(x)$ has at least one solution in $[a, b]$.

Lemma 1.2. (Uniqueness I) Let $g(x)$ be continuous on $[a, b]$ and assume that $g([a, b]) \subset [a, b]$. Further, assume there is a $\lambda \in (0, 1)$ such that

$$|g(x) - g(y)| \leq \lambda|x - y|, \quad \forall x, y \in [a, b]. \quad (1.6)$$

Then $x = g(x)$ has a unique solution $p \in [a, b]$ and the iterates will converge for any $x_0 \in [a, b]$ with

$$|p - x_n| \leq \frac{\lambda^n}{1 - \lambda} |x_1 - x_0|. \quad (1.7)$$

Theorem 1.1. Let $g(x) \in C^1([a, b])$ such that $g([a, b]) \subset [a, b]$ and $\max_{a \leq x \leq b} |g'(x)| = \lambda < 1$ then

(i) $x = g(x)$ has a unique solution $p \in [a, b]$.

(ii) The iteration (1.5) converges for any $x_0 \in [a, b]$.

(iii)

$$|p - x_n| \leq \lambda^n |p - x_0| \leq \frac{\lambda^n}{1 - \lambda} |x_1 - x_0| \quad (1.8)$$

$$\lim_{n \rightarrow \infty} \frac{p - x_{n+1}}{p - x_n} = g'(p) \quad (1.9)$$

1.3 Newton's Method

Newton's method is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (1.10)$$

We can derive Newton's method by considering a Taylor expansion of f about x_n .

$$f(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{(x - x_n)^2}{2} f''(\xi) \quad (1.11)$$

where ξ between x and x_n . Let p be our root. Then we find

$$0 = f(x_n) + (p - x_n)f'(x_n) + \frac{(p - x_n)^2}{2} f''(\xi) \quad (1.12)$$

$$p = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(p - x_n)^2}{2} \frac{f''(\xi)}{f'(x_n)}. \quad (1.13)$$

Dropping the error term, we find an approximation to the root, p , that is indeed Newton's method.

1.4 Solving Systems of Non-Linear Equations

Newton's method for systems of non-linear equations can be derived in a similar method as shown in §1.3. Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $x, x_n, p \in \mathbb{R}^n$. If f_i is the i^{th} equation of F , we can write the Taylor expansion for F about x_n as

$$f_i(x) = f_i(x_n) + \sum_{j=1}^n (x_j - x_{j,n}) \frac{\partial f_i(x_n)}{\partial x_j} + H.O.T. \quad (1.14)$$

If p is a root of f_i and G is the Jacobian matrix for F , then if we drop the higher order terms in (1.14) we find

$$0 = F(x_n) + G(x_n)(p - x_n) \quad (1.15)$$

$$-G(x_n)p = F(x_n) - G(x_n)x_n \quad (1.16)$$

$$p = x_n - G^{-1}(x_n)F(x_n). \quad (1.17)$$

1.5 Qual Problems

Winter 2003, #3

Fall 2004, #1

Fall 2005, #1

Newton's Method

Fall 2000, #2

Winter 2002, #1

Spring 2002, #1

Bisection

Winter 2005, #1

Fall 2006, #1

Fixed Point Iteration

Spring 2001, #2

Fall 2001, #1

Winter 2004, #1

Spring 2006, #1

2 Interpolation Theory

Given a set of data points, $\{(x_i, y_i)\}_{i=1}^n$, select a function $f(x)$ from some class of functions such that $y_i = f(x_i)$ for $i = 1, \dots, n$.

2.1 Polynomial Interpolation

For these problems, we choose our class of functions to be polynomials. These are the most common interpolation problems found on the numerical qualifying exams.

2.1.1 Lagrange Polynomials

A simple formula for constructing interpolating polynomials is given by Lagrange's formula.

$$p_n(x) = \sum_{i=0}^n y_i l_i(x), \quad l_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \quad (2.1)$$

2.1.2 Newton Divided Differences

Newton divided differences are useful because they provide an easy method for generating interpolating polynomials. The polynomial is computed from

$$p_n(x) = p_{n-1}(x) + (x - x_0) \cdots (x - x_{n-1}) f[x_0, \dots, x_n] \quad (2.2)$$

where

$$p_0(x) = f(x_0) \quad (2.3)$$

and

$$f[x_0, x_1, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}. \quad (2.4)$$

To compute interpolating polynomials using divided differences, we build the table shown below.

x_i	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}] \cdots$
x_0	f_0	$f[x_0, x_1]$	$f[x_0, x_1, x_2] \cdots$
x_1	f_1	$f[x_1, x_2]$	$f[x_1, x_2, x_3] \cdots$
x_2	f_2	$f[x_2, x_3]$	$f[x_2, x_3, x_4] \cdots$
x_3	f_3	$f[x_3, x_4]$	
x_4	f_4		
\vdots	\vdots	\vdots	\vdots

Now let's look at the error in the interpolating polynomial. Let $t \in \mathbb{R}$ be distinct from the node points x_0, x_1, \dots, x_n . Using divided differences to interpolate $f(x)$ at the node points and t , we find

$$p_{n+1}(x) = p_n(x) + (x - x_0) \cdots (x - x_n) f[x_0, \dots, x_n, t]. \tag{2.5}$$

We have that $p_{n+1}(t) = f(t)$, so let $x = t$ and we find

$$f(t) - p_n(t) = (t - x_0) \cdots (t - x_n) f[x_0, \dots, x_n, t]. \tag{2.6}$$

From the formula given in Theorem 2.2, we can conclude that

$$f[x_0, \dots, x_n, t] = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{2.7}$$

2.1.3 Hermite Interpolation

Hermite polynomials are used when we wish to interpolate both the function ($p(x)$ interpolates $f(x)$) and its first derivative ($p'(x)$ interpolates $f'(x)$). The general problem here is given by: Find a polynomial $p(x)$ such that

$$\begin{aligned} p^{(i)}(x_1) &= y_1^{(i)} & i &= 0, 1, \dots, \alpha_1 - 1 \\ &\vdots & & \\ p^{(i)}(x_n) &= y_n^{(i)} & i &= 0, 1, \dots, \alpha_n - 1 \end{aligned} \tag{2.8}$$

Here the $y_j^{(i)}$ are given data and α_j are the number of conditions imposed on $p(x)$ at x_j for $j = 1, \dots, n$. If we define $N = \alpha_1 + \dots + \alpha_n$ then we have a unique polynomial of degree at most $N - 1$ that satisfies the above equations. To show this, we basically follow the proof for Theorem 2.1, noting that if a polynomial $q(x)$ has its first α derivatives equal to zero at x_j , where x_j is a root, then x_j is a root of order $\alpha - 1$ and we can write

$$q(x) = r(x)(x - x_j)^{\alpha-1}. \tag{2.9}$$

2.2 Trigonometric Interpolation

[Incomplete]

2.3 Key Theorems

Theorem 2.1. Given $n + 1$ distinct points x_0, \dots, x_n and associated function values y_0, \dots, y_n , there is a polynomial, $p(x)$, of degree at most n that interpolates y_i at x_i . Further, this polynomial is unique.

Proof. The existence of a polynomial, $p(x)$, of degree at most n is given by the construction in equation (2.1). To show uniqueness, let $p_1(x)$ and $p_2(x)$ be such that $p_j(x_i) = y_i$ for $j = 1, 2$ and $i = 0, \dots, n$ (that is p_1 and p_2 are both interpolating polynomials). Let $r(x) = p_1(x) - p_2(x)$. Clearly, $r(x)$ has degree less than or equal to n . Further, $r(x_i) = p_1(x_i) - p_2(x_i) = y_i - y_i = 0$. So, $r(x)$ has $n + 1$ zeros which implies that $r(x) \equiv 0 \Rightarrow p_1(x) = p_2(x)$ and hence, the interpolating polynomial is unique. \square

Theorem 2.2. Let x_0, x_1, \dots, x_n be distinct real numbers contained in the interval I and let $f \in C^{n+1}(I)$. Then, for each $x \in I$, there exists $\xi \in I$ such that

$$f(x) - p_n(x) = \frac{(x - x_0) \cdots (x - x_n)}{(n + 1)!} f^{(n+1)}(\xi). \quad (2.10)$$

2.4 Qual Problems

Polynomial Interpolation

Spring 2002, #2

Fall 2003, #3

Winter 2004, #3

Spring 2006, #3

Trigonometric Interpolation

Fall 2001, #2

3 Numerical Differentiation

Numerical differentiation formulas can be found by first constructing an interpolating polynomial $p_n(x)$ for your function $f(x)$ at a set of node points and then computing the derivative of $p_n(x)$. Recall from §2.1.1, the interpolating polynomial is given by

$$p_n(x) = \sum_{i=0}^n f(x_i) l_i(x). \quad (3.1)$$

Now

$$p'_n(x) = \sum_{i=0}^n f(x_i) l'_i(x) \equiv D_h f(x). \quad (3.2)$$

Further, recall the formula for the error of the interpolating polynomial

$$f(x) - p_n(x) = \psi_n(x) f[x_0, \dots, x_n, x]. \quad (3.3)$$

Then

$$f'(x) - D_h f(x) = \psi'_n(x) f[x_0, \dots, x_n, x] + \psi_n(x) f[x_0, \dots, x_n, x, x] \quad (3.4)$$

$$= \psi'_n(x) \frac{f^{(n+1)}(\xi_1)}{(n + 1)!} + \psi_n(x) \frac{f^{(n+2)}(\xi_2)}{(n + 2)!} \quad (3.5)$$

To obtain equations for higher order derivatives, we simply continue to differentiate the above equations. To find higher order formulas, we use a higher order interpolating polynomial.

3.1 Method of Undetermined Coefficients

This method provides both a simple way to derive formulas for numerical differentiation as well as a simple method for finding the error for a difference formula. The main idea of this method is to Taylor expand all terms about the node at which the derivative is being evaluated. We illustrate with an example. Say we wish to find a formula to approximate $f''(x)$ using function values at $x - h, x$, and $x + h$. Then, we write our method as

$$D_h^{(2)} f(x) = c_1 f(x + h) + c_2 f(x) + c_3 f(x - h). \quad (3.6)$$

Now, we need Taylor expansions for $f(x + h)$ and $f(x - h)$, given by

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2} f''(x) \pm \frac{h^3}{6} f^{(3)}(x) + \frac{h^4}{24} f^{(4)}(\xi_{\pm}) \quad (3.7)$$

with $\xi_{\pm} \in [x \pm h, x]$. Plugging these expansions into (3.6) and grouping terms by powers of h , we find

$$\begin{aligned} c_1 f(x + h) + c_2 f(x) + c_3 f(x - h) = \\ (c_1 + c_2 + c_3)f(x) + (c_1 - c_3)hf'(x) + (c_1 + c_3)\frac{h^2}{2}f''(x) + (c_1 - c_3)\frac{h^3}{6}f^{(3)}(x) \\ + \frac{h^4}{24}[c_1 f^{(4)}(\xi_+) + c_3 f^{(4)}(\xi_-)]. \end{aligned} \quad (3.8)$$

Note that in order to obtain $f''(x)$, we need

$$c_1 + c_2 + c_3 = 0, \quad c_1 - c_3 = 0, \quad c_1 + c_3 = \frac{2}{h^2}. \quad (3.9)$$

Solving for the coefficients, we find that

$$c_1 = \frac{1}{h^2}, \quad c_2 = \frac{-2}{h^2}, \quad c_3 = \frac{1}{h^2}. \quad (3.10)$$

Using the Taylor expansions and the coefficients we found, we can find the error,

$$f''(x) - D_h^{(2)} f(x) = -\frac{h^2}{12} f^{(4)}(\xi) \quad (3.11)$$

with $\xi \in [x - h, x + h]$.

3.2 Common Formulae

In this section, we give some of the most common finite difference equations. Table 1 has the equations for one-sided differences and Table 2 lists equations for centered differences.

Table 1: Weights for common one-sided finite differences. The derivative is at $x = 0$ and the nodes are numbered.

Order of Accuracy	Nodes			
	0	1	2	3
1st Derivative				
1	-1	1		
2	$-\frac{3}{2}$	2	$-\frac{1}{2}$	
3	$-\frac{11}{6}$	3	$-\frac{3}{2}$	$\frac{1}{3}$
2nd Derivative				
1	1	-2	1	
2	2	-5	4	-1
3rd Derivative				
1	-1	3	-3	1

Table 2: Weights for common centered finite differences. The derivative is at $x = 0$ and the nodes are numbered.

Order of Accuracy	Nodes				
	-2	-1	0	1	2
1st Derivative					
2		$-\frac{1}{2}$	0	$\frac{1}{2}$	
4	$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$
2nd Derivative					
2		1	-2	1	
4	$-\frac{1}{12}$	$\frac{16}{12}$	$-\frac{30}{12}$	$\frac{16}{12}$	$-\frac{1}{12}$
3rd Derivative					
2	$-\frac{1}{2}$	1	0	-1	$\frac{1}{2}$

3.3 Richardson Extrapolation

Richardson extrapolation can be used any time where we have a formula $N(h)$ that approximates an unknown value, M , where the truncation error has the form

$$M - N(h) = K_1 h + K_2 h^2 + K_3 h^3 + \dots, \quad (3.12)$$

K_i constants. The idea here is simple. Find a way to combine lower-order approximations in order to obtain a formula with a higher-order truncation error. This process is most easily illustrated with a simple example. Consider the approximation of M given by

$$M = N(h) + K_1 h + K_2 h^2 + K_3 h^3 + \dots \quad (3.13)$$

and

$$M = N\left(\frac{h}{2}\right) + K_1 \frac{h}{2} + K_2 \frac{h^2}{4} + K_3 \frac{h^3}{8} + \dots \quad (3.14)$$

Now, look at $2 \times (3.14) - (3.13)$. Letting $N_1(h) \equiv N(h)$, we find

$$M = N_1\left(\frac{h}{2}\right) + \left(N_1\left(\frac{h}{2}\right) - N_1(h)\right) + K_2 \left(\frac{h^2}{2} - h^2\right) + \dots \quad (3.15)$$

If we define

$$N_2(h) = N_1\left(\frac{h}{2}\right) + \left(N_1\left(\frac{h}{2}\right) - N_1(h)\right) \quad (3.16)$$

we now have an $O(h^2)$ approximation for M given by $N_2(h)$. This process can be continued to generate an approximation of any order. In general, for $j = 2, 3, \dots, m$ we have an $O(h^j)$ approximation given by

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{2^{j-1} - 1}. \quad (3.17)$$

3.4 Qual Problems

Fall 2000, #1
 Spring 2001, #3
 Winter 2002, #2
 Winter 2003, #1
 Winter 2004, #2
 Fall 2005, #2

Extrapolation
 Spring 2002, #3

4 Numerical Integration

(Incomplete)

4.1 Newton-Cotes Formulas

4.2 Composite Integration

4.3 Gaussian Quadrature

Gauss-Legendre

Linear change of variables

$$\int_a^b f(t)dt = \left(\frac{b-a}{2}\right) \int_{-1}^1 f\left(\frac{a+b+x(b-a)}{2}\right) dx \quad (4.1)$$

4.4 Key Theorems

4.5 Qual Problems

Gaussian Quadrature

Spring 2001, #1 (Gauss-Legendre)
 Fall 2004, #2
 Fall 2006, #2 (Gauss-Laguerre)

Newton-Cotes

Fall 2002, #1 (Composite)
 Fall 2003, #1 (Composite)

Fall 2002, #2 (Extrapolation)

5 Iterative Methods

In this section, we are interested in solving $Ax = b$ by generating a sequence of approximate solutions, $\{x^{(k)}\}$, that converge to the solution x . A few useful quantities for analyzing the convergence of iterative methods include the *residual*, $r = b - A\hat{x}$ where \hat{x} is a numerical approximation to the solution x , and the *condition number* of a matrix, $\text{cond}(A) = \|A\| \|A^{-1}\|$.

5.1 Relaxation Methods

These methods depend on a splitting, $A = M - N$, of the matrix A . The iteration is given by

$$Mx^{(k+1)} = Nx^{(k)} + b. \quad (5.1)$$

We start by writing $A = D + U + L$ where D, U , and L are the diagonal, strictly upper triangular, and strictly lower triangular parts of the matrix A . The most common splittings are:

$$\text{Jacobi} \quad M_J = D \quad N_J = -(L + U)$$

$$\text{Gauss-Seidel} \quad M_G = (D + L) \quad N_G = -U$$

$$\text{Successive Over-Relaxation} \quad M_\omega = D + \omega L \quad N_\omega = (1 - \omega)D - \omega U$$

Note that for SOR we have the addition of the relaxation parameter $\omega \in \mathbb{R}$. We also have a different characterization of these methods which may provide some insight.

$$\text{Jacobi} \quad x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) \quad (5.2)$$

$$\text{Gauss-Seidel} \quad x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) \quad (5.3)$$

$$\text{SOR} \quad x_i^{(k+1)} = \frac{1}{a_{ii}} \omega \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) + (1 - \omega)x_i^{(k)} \quad (5.4)$$

Note that the difference between Jacobi and the other methods is simply using the most recently available values for the x_i .

5.1.1 Key Theorems

For these theorems, we assume that $b \in \mathbb{R}^n$, $A = M - N \in \mathbb{R}^{n \times n}$, and A, M are non-singular. The spectral radius of A is denoted by $\rho(A)$.

Theorem 5.1. *If $\rho(M^{-1}N) < 1$ then the iterates $x^{(k)}$ defined by (5.1) converge to $x = A^{-1}b$ for any initial guess, $x^{(0)}$.*

Proof. We start by looking at the error found by subtracting the iterative scheme from the actual solution,

$$\begin{aligned} M(x - x^{(k+1)}) &= N(x - x^{(k)}) + b - b \\ Me^{(k+1)} &= Ne^{(k)} \\ e^{(k+1)} &= M^{-1}Ne^{(k)}. \end{aligned}$$

Now, consider

$$\begin{aligned}\|e^{(k+1)}\| &= \|(M^{-1}N)e^{(k)}\| \\ &= \|(M^{-1}N)^{k+1}e^{(0)}\| \\ &\leq \|M^{-1}N\|^{k+1}\|e^{(0)}\|.\end{aligned}$$

If $\rho(M^{-1}N) < 1$ then $\|M^{-1}N\|^k \rightarrow 0$ as $k \rightarrow \infty$ and so $\|e^{(k)}\| \rightarrow 0$ and the method converges. \square

Theorem 5.2. *If A is symmetric positive definite, then the Gauss-Seidel iteration converges for any $x^{(0)}$.*

5.2 Qual Problems

Relaxation Methods

Fall 2001, #3

Fall 2002, #3

Fall 2003, #2

Winter 2005, #3

Error Analysis

Spring 2006, #2

6 Direct Methods for Solving Linear Systems

Here, we are again interested in solving linear systems $Ax = b$.

6.1 LU Factorization

Often, a matrix can be factored into the form

$$A = LU \tag{6.1}$$

where L is a lower triangular matrix and U is an upper triangular matrix. This is commonly done to solve linear systems numerically. Below, we give conditions on the existence of this factorization.

Theorem 6.1. *$A \in \mathbb{R}^{n \times n}$ has an LU factorization if $\det(A(1:k, 1:k)) \neq 0$ for $k = 1, \dots, n-1$. If the LU factorization exists and A is nonsingular, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Theorem 6.2. *If A^T is strictly diagonally dominant, then A has an LU factorization and $|l_{ij}| \leq 1$.*

We then solve the system of equations first with a forward substitution step, $Ly = b$, and then back-substitution $Ux = y$.

6.2 Tridiagonal Systems

Special algorithms exist for solving tridiagonal systems in $O(n)$ operations. As before, we will perform an LU factorization of the matrix A and are again given the choice of which matrix has ones on the diagonal. The two factorizations, $A = LU$, are illustrated below.

$$\begin{pmatrix} a_1 & c_1 & & & & \\ b_2 & a_2 & c_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & b_{n-1} & a_{n-1} & c_{n-1} & \\ & & & b_n & a_n & \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ d_2 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & d_n & 1 & \end{pmatrix} \begin{pmatrix} e_1 & c_1 & & & & \\ e_2 & c_2 & & & & \\ & \ddots & \ddots & & & \\ & & e_{n-1} & c_{n-1} & & \\ & & & e_n & & \end{pmatrix} \quad (6.2)$$

$$= \begin{pmatrix} \alpha_1 & & & & & \\ b_2 & \alpha_2 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & b_n & \alpha_n & \end{pmatrix} \begin{pmatrix} 1 & \gamma_2 & & & & \\ & 1 & \gamma_3 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & \gamma_n & \\ & & & & 1 & \end{pmatrix} \quad (6.3)$$

For each of these factorizations, we can obtain a recurrence relation to determine the unknown values by simply multiplying the L and U matrices together. Note that e_1 and α_1 must be handled differently. The resulting relations for (6.2) are then

$$e_1 = a_1, \quad d_i = \frac{e_{i-1}}{b_i}, \quad e_i = a_i - d_i c_{i-1} \quad (6.4)$$

for $i = 2, \dots, n$. Now we need to solve $LUx = f$ by solving $Ly = f$ and then $Ux = y$. In this case, we end up with the following equations

$$y_1 = f_1, \quad y_i = f_i - d_i y_{i-1} \quad (6.5)$$

$$x_n = \frac{y_n}{e_n}, \quad x_j = \frac{y_j - c_j x_{j+1}}{e_j} \quad (6.6)$$

for $i = 2, \dots, n$ and $j = n-1, \dots, 1$. Note the operation counts in this problem. We have $3n-3$ floating point operations in finding L and U . The forward solve contains other $2n-2$ operations and $4n-4$ operations for the back-substitution. Note that we have obtained a solution using order n floating point operations.

6.3 Qual Problems

LU Factorization

Fall 2000, #3

Tri-Diagonal Systems

Winter 2002, #3

Fall 2005, #3

7 Approximation of Functions

(Incomplete)

7.1 Least Squares

Here we are given a set of data points $\{(x_i, y_i) | i = 1, \dots, n\}$ and wish to fit some function f to the data such that it minimizes

$$E = \left(\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 \right)^{\frac{1}{2}}. \quad (7.1)$$

We are only going to consider the case where $f = P_n(x)$, a polynomial of degree n . To find this minimum, we need to compute the normal equations, found by differentiating (7.1) with respect to each of the unknown coefficients of $P_n(x)$ and setting these equal to zero.

Normal equations

$$A^T Ax = A^T b \tag{7.2}$$

7.2 Qual Problems

Winter 2003, #2

Fall 2006, #3

8 Numerical Methods for Ordinary Differential Equations

In this section we look at methods for solving

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0. \tag{8.1}$$

The techniques developed in this section can also be applied to systems of ODEs,

$$\vec{y}' = \vec{f}(t, \vec{y}), \quad \vec{y}(x_0) = \vec{y}_0. \tag{8.2}$$

This is of particular interest since we can easily convert higher order equations into systems of first order equations, as seen in the following example:

$$y^{(m)} = f(t, y, y', \dots, y^{(m-1)}) \tag{8.3}$$

$$y(t_0) = y_0, \dots, y^{(m-1)}(t_0) = y_0^{(m-1)} \tag{8.4}$$

Now let

$$y_1 = y, \quad y_2 = y', \quad \dots, \quad y_m = y^{(m-1)}. \tag{8.5}$$

Then we have

$$\begin{array}{ll} y_1' = y_2 & y_1(t_0) = y_0 \\ y_2' = y_3 & y_2(t_0) = y_0' \\ \vdots & \vdots \\ y_{m-1}' = y_m & y_{m-1}(t_0) = y_0^{(m-2)} \\ y_m' = f(t, y_1, \dots, y_m) & y_m(t_0) = y_0^{(m-1)}. \end{array} \tag{8.6}$$

Also, by using systems we can convert non-autonomous systems into autonomous systems. An autonomous system has no explicit t dependence in the right-hand side,

$$\vec{y}' = \vec{f}(\vec{y}), \quad \vec{y}(x_0) = y_0. \tag{8.7}$$

Suppose we have an n -dimensional non-autonomous system, $\vec{y}' = \vec{f}(\vec{y}, t)$. Define a new vector, $\vec{v} = (y_1, \dots, y_n, t)^t$. Then

$$\frac{d\vec{v}}{dt} = \begin{pmatrix} f(v_1, \dots, v_{n+1}) \\ 1 \end{pmatrix} \tag{8.8}$$

with initial conditions $v_1 = y_0^1, \dots, v_n = y_0^n, v_{n+1} = t_0$ is an autonomous system of dimension $n + 1$.

8.1 One Step Methods

A general explicit one step method (Euler, Runge-Kutta) can be written in the form

$$y_{k+1} = y_k + h\Phi(y_k, h). \quad (8.9)$$

Now let $y_k = y(t_k)$ be values of the exact solution to the differential equation. The local truncation error, τ_k , is then given by

$$\tau_k = y_{k+1} - y_k - h\Phi(y_k, h). \quad (8.10)$$

Specifically, we are interested in finding the leading order term of the local truncation error,

$$\tau_k = c(t_k)h^{p+1} + O(h^{p+2}) \quad (8.11)$$

where the method is said to be *order p*. To find the local truncation error, we Taylor expand all terms about y_k and cancel terms until we can determine the order of the method.

Definition 8.1. (*Consistency*) A method is said to be consistent if

$$|\tau_k| \rightarrow 0 \text{ as } h \rightarrow 0. \quad (8.12)$$

8.1.1 Stability Analysis

In this section, we concern ourselves with determining the *region of absolute stability* of an ODE method. To find the region of absolute stability, we consider the model equation

$$\frac{dy}{dt} = \lambda y. \quad (8.13)$$

Solutions to (8.13) go to zero for all initial conditions when $\lambda < 0$. In the following definition, h is the step size in the method.

Definition 8.2. (*Region of Absolute Stability*) The region of absolute stability is the set of all $h\lambda \in \mathbb{C}$ such that the numerical solution $\{y_n\} \rightarrow 0$ as $t_n \rightarrow \infty$ when applied to equation (8.13).

Definition 8.3. (*A-Stability*) A method is called A-Stable when the region of absolute stability includes the entire left-half (complex) plane.

8.1.2 Common Methods and Their Properties

We use the notation S to denote the interval of absolute stability in this section.

Forward Euler (Explicit, Order 1, $S = (-2, 0)$)

$$y_{k+1} = y_k + hf(t_k, y_k) \quad (8.14)$$

Backward Euler (Implicit, Order 1, $S = (-\infty, 0) \cup (2, \infty)$)

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1}) \quad (8.15)$$

Trapezoidal Method (Implicit, Order 2, $S = (-\infty, 0)$)

$$y_{k+1} = y_k + \frac{1}{2}h[f(t_n, y_n) + f(t_{n+1}, y_{n+1})] \quad (8.16)$$

Theta Method

$$y_{k+1} = y_k + h[\theta f(t_k, y_k) + (1 - \theta)f(t_{k+1}, y_{k+1})] \quad (8.17)$$

order	θ	method
2	1/2	Trapezoidal
1	0	Backward Euler
1	1	Forward Euler
1	otherwise	—

Runge-Kutta 2 (Explicit, Maximum Order 2, $S = (-2, 0)$)

$$y^* = y_k + \alpha h f(y_k) \quad (8.18)$$

$$y_{k+1} = y_k + h\beta_1 f(y_k) + h\beta_2 f(y^*) \quad (8.19)$$

Second order if $\alpha = \beta_1 = \beta_2 = 1/2$. The interval of absolute stability depends on the choice of coefficients.

Runge-Kutta 3 (Explicit, Order 3)

$$k_1 = f(t_k, y_k), \quad (8.20)$$

$$k_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right), \quad (8.21)$$

$$k_3 = f(t_k + h, y_k + h(-k_1 + 2k_2)), \quad (8.22)$$

$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 4k_2 + k_3) \quad (8.23)$$

8.2 Linear Multistep Methods

The general form of a linear multistep method (LMM) is

$$y_{n+1} = \sum_{j=0}^p \alpha_j y_{n-j} + h \sum_{j=-1}^p \beta_j f_{n-j}. \quad (8.24)$$

One way to derive these methods is via numerical integration. Here, the general idea is to reformulate the differential equation by integrating over some interval, $[t_{n-r}, t_{n+1}]$ yielding

$$y(t_{n+1}) = y(t_{n-r}) + \int_{t_{n-r}}^{t_{n+1}} f(t, y(t)) dt \quad (8.25)$$

Now we can construct an interpolant for the integrand $f(t, y(t))$ and integrate it over the interval. If we drop the error term from the numerical integration, we are then left with a LMM. For an example of deriving a LMM using the trapezoidal rule to approximate the integral, see the solutions set, Spring 2006, #4.

As with single step methods, we are now interested in looking at the local truncation error and the stability of the method. The local truncation error analysis proceeds exactly as it did for one step methods, although now we will generally choose to Taylor expand about t_{n+1} rather than t_n in order to minimize the number of Taylor expansions. The following theorem gives us an easy method for characterizing the consistency of a LMM.

Theorem 8.1. *A linear multistep method is consistent if and only if*

$$\sum_{j=0}^p \alpha_j = 1, \quad -\sum_{j=0}^p j\alpha_j + \sum_{j=-1}^p \beta_j = 1. \quad (8.26)$$

For stability analysis, we look at solving the problem

$$y' \equiv 0, \quad y(0) = 1. \quad (8.27)$$

Consider the polynomial

$$\rho(r) = r^{p+1} - \sum_{j=0}^p \alpha_j r^{p-j} \quad (8.28)$$

and let r_0, \dots, r_p be the roots of $\rho(r)$.

Definition 8.4. (*Root Condition*) A linear multistep method satisfies the root condition if

1. $|r_j| \leq 1, \quad j = 0, 1, \dots, p$
2. if $|r_j| = 1 \Rightarrow \rho'(r_j) \neq 0$ (i.e. the root is simple)

Consistency and the root condition allow us to discuss the stability and convergence of the method through the following two theorems.

Theorem 8.2. (*Stability for LMMs*) Assume the consistency condition. Then the multistep method is stable if and only if the root condition is satisfied.

Theorem 8.3. (*Convergence of LMMs*) A consistent multistep method is convergent if and only if the root condition is satisfied.

8.3 Predictor-Corrector Methods

A *predictor-corrector* method consists of two steps. First, we use an explicit *predictor* formula and then an implicit *corrector* step. It is very common to use an Adams-Bashforth formula as a predictor and an Adams-Moulton formula as the corrector. The simplest case (and most likely to occur on a qualifying exam) would be using AB1 (Forward Euler) as the predictor and AM2 (the Trapezoidal Rule) as the corrector,

$$y_{i+1}^* = y_i + hF_i \quad (8.29)$$

$$y_{i+1} = y_i + \frac{h}{2}(F_i + F_i^*). \quad (8.30)$$

Note that in this case, we have actually obtained an RK2 method.

8.4 Qual Problems

There is one ODE question on each numerical analysis qualifying exam.

9 Finite Differences for PDEs

In this section we are primarily concerned with proposing finite difference methods for solving various PDEs and then verifying certain properties of those methods, namely the order of accuracy and stability. By finding the order of accuracy, we will be able to verify whether the method is consistent and stability will be verified through Theorem 9.2. Then via the Lax-Richtmyer Equivalence Theorem (Theorem 9.1), we will have that the method is convergent.

Definition 9.1. (*Consistency*) Let $Pu = f$ be a partial differential equation and $P_{k,h}v = f$ be a finite difference scheme for solving the PDE. We say that the finite difference scheme is consistent with the PDE if for any smooth function $\phi(t, x)$

$$P\phi - P_{k,h}\phi \rightarrow 0 \text{ as } k, h \rightarrow 0. \quad (9.1)$$

9.1 Order of Accuracy

There are two approaches we can use to find the order of accuracy for finite difference schemes for PDEs. The first is to expand all the terms in Taylor series about a common point and cancel terms in the truncation error until we find the leading order terms. This can also be done through symbols of the difference operator and the differential operator, which we will describe in more detail here. In this section, we write our differential equation as $Pu = f$ and the difference method in the form $P_{k,h}v = R_{k,h}f$. Now we have that

Definition 9.2. *A scheme $P_{k,h}v = R_{k,h}f$ that is consistent with the PDE $Pu = f$ is accurate of order p in time and q in space if for any smooth function $\phi(t, x)$,*

$$P_{k,h}\phi - R_{k,h}P\phi = O(k^p) + O(h^q). \quad (9.2)$$

We say the method is accurate of order (p, q) .

9.1.1 Using Symbols

We begin by defining the symbols of the difference operator and the differential operator.

Definition 9.3. *(Symbol of the Difference Operator) The symbol $p_{k,h}(s, \xi)$ of a difference operator $P_{k,h}$ is given by*

$$p_{k,h}(s, \xi) = \frac{P_{k,h}(e^{skn}e^{imh\xi})}{e^{skn}e^{imh\xi}}. \quad (9.3)$$

Definition 9.4. *(Symbol of the Differential Operator) The symbol of the differential operator P is given by*

$$p(s, \xi) = \frac{P(e^{st}e^{i\xi x})}{e^{st}e^{i\xi x}}. \quad (9.4)$$

Now from Theorem 9.3, we look at

$$p_{k,h}(s, \xi) - r_{k,h}(s, \xi)p(s, \xi) = O(k^p) + O(h^q). \quad (9.5)$$

We then Taylor expand as needed on the left-hand side and cancel terms until we find the appropriate order. For a specific example, see the solutions set.

9.2 Stability Analysis

To determine the stability of the method, we look at the homogeneous equation (that is, we are only interested in the difference operator). To carry out this analysis, we make the substitution $g^n e^{im\theta}$ for v_m^n where $\theta = h\xi$ in the homogeneous equation. We then solve for g , the *amplification factor*. In particular, if we have found the symbol of the difference scheme, $p_{k,h}(s, \xi)$, we can simply substitute $e^{sk} = g$. We then take the resulting equation and manipulate it to obtain a form such that we can use Theorem 9.2. Examples of this analysis are included in the solutions set. There are two useful trigonometric identities when working with these equations that we note here.

$$1 - \cos \theta = 2 \sin^2 \frac{1}{2} \theta \quad (9.6)$$

$$\sin \theta = 2 \sin \frac{1}{2} \theta \cos \frac{1}{2} \theta \quad (9.7)$$

9.3 Well-Posedness of Equations

Through the symbol of the differential operator, we have an easy method for showing whether or not a problem is well posed. We look at $p(s, \xi, \eta) = 0$ and find the roots of the symbol in terms of s . We then have a theorem that states the problem is well posed as long as the real part of the roots is bounded for all values of ξ and η . For an example, see Fall 2006, #5 in the solutions set.

9.4 Hyperbolic PDEs

9.4.1 Common Methods

In this section, we give some methods, and their properties, for solving hyperbolic PDEs, using the one-dimensional transport equation as our prototypical example,

$$u_t + au_x = f. \quad (9.8)$$

After listing some of the common methods, we will give the derivation of some of these methods which is needed to adapt it to problems of a slightly different form than (9.8) and which will also help with remembering the method.

Upwind differencing is used on the spatial derivative, au_x .

Upwind Differencing (Order 1 in space)

$$\begin{cases} \frac{1}{h}(v_m^n - v_{m-1}^n), & \text{when } a > 0 \\ \frac{1}{h}(v_{m+1}^n - v_m^n), & \text{if } a < 0 \end{cases} \quad (9.9)$$

Lax-Friedrichs (Order $O(k) + O(h^2) + O(k^{-1}h^2)$, Stable for $|a\lambda| \leq 1$)

$$\frac{v_m^{n+1} - \frac{1}{2}(v_{m+1}^n + v_{m-1}^n)}{k} + a \frac{v_{m+1}^n - v_{m-1}^n}{2h} = f_m^n \quad (9.10)$$

Lax-Wendroff (Order (2,2), Stable if $|a\lambda| \leq 1$)

$$\frac{v_m^{n+1} - v_m^n}{k} + a \frac{v_{m+1}^n - v_{m-1}^n}{2h} - \frac{a^2 k}{2} \frac{v_{m+1}^n - 2v_m^n + v_{m-1}^n}{h^2} = \frac{1}{2}(f_m^{n+1} + f_m^n) - \frac{ak}{4h}(f_{m+1}^n - f_{m-1}^n) \quad (9.11)$$

Crank-Nicolson (Order (2,2), Unconditionally Stable)

$$\frac{v_m^{n+1} - v_m^n}{k} + a \frac{v_{m+1}^{n+1} - v_{m-1}^{n+1} + v_{m+1}^n - v_{m-1}^n}{4h} = \frac{f_m^{n+1} + f_m^n}{2} \quad (9.12)$$

Box Scheme (Order (2,2), Unconditionally Stable)

$$\frac{1}{2k} [(v_m^{n+1} + v_{m+1}^{n+1}) - (v_m^n + v_{m+1}^n)] + \frac{a}{2h} [(v_{m+1}^{n+1} - v_m^{n+1}) + (v_{m+1}^n - v_m^n)] = \frac{1}{4}(f_{m+1}^{n+1} + f_m^{n+1} + f_{m+1}^n + f_m^n) \quad (9.13)$$

We now give a derivation of the Lax-Wendroff method for equation (9.8). We start by Taylor expanding the solution in time

$$u(x, t+k) = u(x, t) + ku_t(x, t) + \frac{k^2}{2}u_{tt} + O(k^3). \quad (9.14)$$

Now we use the following relations

$$u_t = -au_x + f \quad (9.15)$$

$$u_{tt} = -a(u_t)_x + f_t \quad (9.16)$$

$$= -a(-au_{xx} + f_x) + f_t \quad (9.17)$$

$$= a^2u_{xx} - af_x + f_t \quad (9.18)$$

Now, we plug these into the Taylor expansion

$$u(x, t + k) = u(x, t) + k(-au_x + f) + \frac{k^2}{2}(a^2u_{xx} - af_x + f_t) + O(k^3) \quad (9.19)$$

Here we switch to difference notation and replace all the derivatives with second order accurate centered finite difference approximations in space and a forward difference in time, yielding

$$u_m^{n+1} = u_m^n - \frac{ak}{2h}(u_{m+1}^n - u_{m-1}^n) + \frac{a^2k^2}{2h^2}(u_{m+1}^n - 2u_m^n + u_{m-1}^n) + kf_m^n - \frac{ak^2}{4h}(f_{m+1}^n - f_{m-1}^n) + \frac{k}{2}(f_m^{n+1} - f_m^n) \quad (9.20)$$

$$= u_m^n - \frac{ak}{2h}(u_{m+1}^n - u_{m-1}^n) + \frac{a^2k^2}{2h^2}(u_{m+1}^n - 2u_m^n + u_{m-1}^n) - \frac{ak^2}{4h}(f_{m+1}^n - f_{m-1}^n) + \frac{k}{2}(f_m^{n+1} + f_m^n) \quad (9.21)$$

9.5 Parabolic PDEs

9.5.1 Common Methods

We give the form of these methods for solving the one-dimensional non-homogeneous heat equation,

$$u_t = bu_{xx} + f(t, x). \quad (9.22)$$

Adpating them to other methods will hopefully be straightforward.

Backward-Time Central-Space (Order (1,2), Implicit, Unconditionally Stable)

$$\frac{v_m^{n+1} - v_m^n}{k} = b \frac{v_{m+1}^{n+1} - 2v_m^{n+1} + v_{m-1}^{n+1}}{h^2} + f_m^{n+1} \quad (9.23)$$

Crank-Nicolson (Order (2,2), Implicit, Unconditionally Stable)

$$\frac{v_m^{n+1} - v_m^n}{k} = \frac{1}{2}b \frac{v_{m+1}^{n+1} - 2v_m^{n+1} + v_{m-1}^{n+1}}{h^2} + \frac{1}{2}b \frac{v_{m+1}^n - 2v_m^n + v_{m-1}^n}{h^2} + \frac{1}{2}(f_m^{n+1} + f_m^n) \quad (9.24)$$

Du Fort-Frankel (Order $O(h^2) + O(k^2) + O(k^2h^{-2})$, Explicit, Unconditionally Stable)

$$\frac{v_m^{n+1} - v_m^{n-1}}{2k} = b \frac{v_{m+1}^n - (v_m^{n+1} + v_m^{n-1}) + v_{m-1}^n}{h^2} + f_m^n \quad (9.25)$$

ADI (Peaceman-Rachford) (Order (2,2), Implicit, Unconditionally Stable)

ADI methods have the advantage that we are only inverting one-dimensional operators. In two dimensions, the method is given by:

$$\left(I - \frac{k}{2}A_{1h} \right) \tilde{v}^{n+1/2} = \left(I + \frac{k}{2}A_{2h} \right) v^n \quad (9.26)$$

$$\left(I - \frac{k}{2}A_{2h} \right) v^{n+1} = \left(I + \frac{k}{2}A_{1h} \right) \tilde{v}^{n+1/2} \quad (9.27)$$

We can also write this method as a single equation

$$\left(I - \frac{k}{2}A_{1h}\right) \left(I - \frac{k}{2}A_{2h}\right) v^{n+1} = \left(I + \frac{k}{2}A_{1h}\right) \left(I + \frac{k}{2}A_{2h}\right) v^n. \quad (9.28)$$

Note that for this method, we have assumed a homogeneous problem. The Peaceman-Rachford method also works in three dimensions. In this case the method is given by

$$\left(I - \frac{k}{2}A_{1h}\right) \tilde{v}^{n+1/3} = \left(I + \frac{k}{2}A_{3h}\right) v^n \quad (9.29)$$

$$\left(I - \frac{k}{2}A_{2h}\right) \tilde{v}^{n+2/3} = \left(I + \frac{k}{2}A_{2h}\right) \tilde{v}^{n+1/3} \quad (9.30)$$

$$\left(I - \frac{k}{2}A_{3h}\right) v^{n+1} = \left(I + \frac{k}{2}A_{1h}\right) \tilde{v}^{n+2/3} \quad (9.31)$$

which can also be written as a single equation,

$$\left(I - \frac{k}{2}A_{1h}\right) \left(I - \frac{k}{2}A_{2h}\right) \left(I - \frac{k}{2}A_{3h}\right) v^{n+1} = \left(I + \frac{k}{2}A_{1h}\right) \left(I + \frac{k}{2}A_{2h}\right) \left(I + \frac{k}{2}A_{3h}\right) v^n \quad (9.32)$$

assuming that A_{1h} and A_{2h} commute.

9.5.2 Variable Diffusion Coefficients

When the diffusivity is a function of time or space, the discretization in space is slightly different. Generally, a term of this form is written

$$(b(t, x)u_x)_x. \quad (9.33)$$

A second order (in space) discretization of this term is given by

$$\frac{b(t_n, x_{m+1/2})(v_{m+1}^n - v_m^n) - b(t_n, x_{m-1/2})(v_m^n - v_{m-1}^n)}{h^2}. \quad (9.34)$$

9.6 Key Theorems

Theorem 9.1. (*Lax-Richtmyer Equivalence Theorem*) *A consistent finite difference scheme for a partial differential equation for which the initial value problem is well-posed is convergent if and only if it is stable.*

Theorem 9.2. (*Stability*) *A one step finite difference scheme (with constant coefficients) is stable in a stability region Λ if and only if there is a constant K (independent of θ , k , and h) such that*

$$|g(\theta, k, h)| \leq 1 + Kk \quad (9.35)$$

with $(k, h) \in \Lambda$. If $g(\theta, k, h)$ is independent of k and h , we have

$$|g(\theta)| \leq 1. \quad (9.36)$$

Theorem 9.3. (*Symbols and Order of Accuracy*) *A scheme $P_{k,h}v = R_{k,h}f$ that is consistent with $Pu = f$ is accurate of order (p, q) if and only if for each value of s and ξ we have*

$$p_{k,h}(s, \xi) - r_{k,h}(s, \xi)p(s, \xi) = O(k^p) + O(h^q). \quad (9.37)$$

9.7 Qual Problems

There are two finite difference problems for PDEs on each qualifying exam. Here we list a general category for the problems along with some specific information of the problem to give an idea of topics that have been asked in recent years. Generally, for every problem we either have to give a method or demonstrate that the given method is of a certain order of accuracy and is stable. Problems marked with (bc's) indicate the question asked about well-posedness in relation to the choice of boundary conditions.

Convection-Diffusion Equation

Fall 2006, #6, (Non-linear)
 Spring 2006, #6 (Max Norm)
 Fall 2004, #6
 Fall 2002, #6 (Max Principle)
 Spring 2001, #7 (L^2 and max norms)

Heat Equation

Fall 2005, #5 (Varying Diffusivity, ADI)
 Winter 2004, #6 (Fourth-Order)
 Fall 2003, #6 (Du Fort-Frankel)
 Spring 2002, #7
 Winter 2002, #6

Wave Equation with Mixed Partial Term

Fall 2006, #5 (Well-Posedness)
 Spring 2006, #5 (Well-Posedness)

Wave Equation plus First Order Terms

Winter 2005, #5 (bc's)
 Winter 2002, #5

General 2nd Order Equations

Fall 2004, #5 (Well-Posedness)
 Fall 2002, #5 (Well-Posedness)

Transport Equation

Fall 2005, #6 (Lax-Wendroff, bc's)
 Winter 2004, #5 (2D, bc's)
 Winter 2003, #5 (Box Scheme)
 Spring 2002, #6 (Non-linear)
 Fall 2001, #5 (2D)

Heat Equation with Mixed Partial Term

Winter 2003, #6 (Well-Posedness)
 Fall 2001, #6 (Well-Posedness)

Wave Equation

Spring 2001, #6 (bc's)

10 Finite Element Methods

In this section, we look at developing finite element methods for scalar elliptic PDEs. We now describe the general form of these problems. Let Ω be a domain in \mathbb{R}^d and let \mathcal{L} be a differential operator of the form

$$\mathcal{L}u = -\nabla \cdot (\sigma \cdot \nabla u) + \beta \cdot \nabla u + \mu u \quad (10.1)$$

where σ, β and μ are functions defined over Ω taking values in $\mathbb{R}^{d,d}$, \mathbb{R}^d and \mathbb{R} , respectively. Let f be a function, $f: \Omega \rightarrow \mathbb{R}$. We wish to find a function $u: \Omega \rightarrow \mathbb{R}$ such that

$$\begin{cases} \mathcal{L}u = f & \text{in } \Omega, \\ \mathcal{B}u = g & \text{on } \partial\Omega \end{cases} \quad (10.2)$$

where \mathcal{B} is an operator that describes the boundary conditions for the problem. In the following sections we will derive the weak formulations of this problem corresponding to different boundary conditions, construct a finite element method, and demonstrate the existence of solutions.

10.1 Weak Formulations

To find a weak formulation of the problem, we multiply the equation by a test function v , chosen in an appropriate space, and then integrate over Ω . Using Green Formula (Theorem 10.1), we will then obtain a weak formulation of the problem.

Dirichlet Boundary Conditions We start with homogeneous Dirichlet boundary conditions. In this case, the equation (10.2) becomes

$$\begin{cases} \mathcal{L}u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \quad (10.3)$$

We choose our space of test functions to be $H_0^1(\Omega)$ since we have the condition $u = 0$ on $\partial\Omega$. Now we have

$$\int_{\Omega} -\nabla \cdot (\sigma \cdot \nabla u)v + v(\beta \cdot \nabla u) + \mu uv = \int_{\Omega} fv \quad (10.4)$$

$$\int_{\Omega} \nabla v \cdot \sigma \cdot \nabla u + v(\beta \cdot \nabla u) + \mu uv - \int_{\partial\Omega} (n \cdot \sigma \cdot \nabla u)v = \int_{\Omega} fv. \quad (10.5)$$

Let's define the following bilinear form

$$a(u, v) = \int_{\Omega} \nabla v \cdot \sigma \cdot \nabla u + v(\beta \cdot \nabla u) + \mu uv. \quad (10.6)$$

The weak formulation is then given by

$$\begin{cases} \text{Seek } u \in H_0^1(\Omega) \text{ such that} \\ a(u, v) = \int_{\Omega} fv, \quad \forall v \in H_0^1(\Omega). \end{cases} \quad (10.7)$$

In the case where the Dirichlet boundary conditions are not homogeneous, say we have $u = g$ on $\partial\Omega$ where $g: \partial\Omega \rightarrow \mathbb{R}$, we have to handle the equation slightly differently. We first assume g is sufficiently smooth so that there exists a lifting, u_g of u in $H^1(\Omega)$. That is, $u_g \in H^1(\Omega)$ and $u_g = g$ on $\partial\Omega$. Then, we can write $u = u_g + \phi$ where $\phi \in H_0^1(\Omega)$. Following the same method as above, we obtain the weak form

$$\begin{cases} \text{Seek } u \in H^1(\Omega) \text{ such that} \\ u = u_g + \phi, \quad \phi \in H_0^1(\Omega), \\ a(\phi, v) = \int_{\Omega} fv - a(u_g, v), \quad \forall v \in H_0^1(\Omega). \end{cases} \quad (10.8)$$

Neumann Boundary Conditions Now we consider boundary conditions of the form $n \cdot \sigma \cdot \nabla u = g$ on $\partial\Omega$ where $g: \partial\Omega \rightarrow \mathbb{R}$. Now, we take our test functions in $H^1(\Omega)$ since we no longer are looking for functions that are zero on $\partial\Omega$. Deriving the weak form, we find

$$\int_{\Omega} -\nabla \cdot (\sigma \cdot \nabla u) + \beta \cdot \nabla u + \mu u = \int_{\Omega} fv \quad (10.9)$$

$$\int_{\Omega} \nabla v \cdot \sigma \cdot \nabla u + v(\beta \cdot \nabla u) + \mu uv - \int_{\partial\Omega} (n \cdot \sigma \cdot \nabla u)v = \int_{\Omega} fv \quad (10.10)$$

$$\int_{\Omega} \nabla v \cdot \sigma \cdot \nabla u + v(\beta \cdot \nabla u) + \mu uv = \int_{\Omega} fv + \int_{\partial\Omega} gv \quad (10.11)$$

giving

$$\begin{cases} \text{Seek } u \in H^1(\Omega) \text{ such that} \\ a(u, v) = \int_{\Omega} fv + \int_{\partial\Omega} gv, \quad \forall v \in H^1(\Omega). \end{cases} \quad (10.12)$$

Mixed Dirichlet-Neumann Boundary Conditions Suppose we have a partition of our boundary, $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, where we impose a Dirichlet boundary condition on $\partial\Omega_D$ and a Neumann condition on $\partial\Omega_N$,

$$\begin{cases} u = 0, & \text{on } \partial\Omega_D \\ n \cdot \sigma \cdot \nabla u = g, & \text{on } \partial\Omega_N \end{cases} \quad (10.13)$$

Table 3: Weak formulations for different boundary conditions for second-order, scalar, elliptic PDEs (10.2).

Problem	V	$a(u, v)$	$f(v)$
Homogeneous Dirichlet	$H_0^1(\Omega)$	$a(u, v)$	$\int_{\Omega} f v$
Neumann	$H^1(\Omega)$	$a(u, v)$	$\int_{\Omega} f v + \int_{\partial\Omega} g v$
Dirichlet-Neumann	$H_{\partial\Omega_D}^1(\Omega)$	$a(u, v)$	$\int_{\Omega} f v + \int_{\partial\Omega_N} g v$
Robin	$H^1(\Omega)$	$a(u, v) + \int_{\partial\Omega} \gamma u v$	$\int_{\Omega} f v + \int_{\partial\Omega} g v$

for a function $g: \partial\Omega_N \rightarrow \mathbb{R}$. The appropriate space for the test functions is now given by

$$H_{\partial\Omega_D}^1(\Omega) = \{u \in H^1(\Omega) : u = 0 \text{ on } \partial\Omega_D\}. \quad (10.14)$$

When finding the weak form, we now split the integral on $\partial\Omega$ into two parts on the different portions of the boundary, yielding the following weak form

$$\begin{cases} \text{Seek } u \in H_{\partial\Omega_D}^1(\Omega) \text{ such that} \\ a(u, v) = \int_{\Omega} f v + \int_{\partial\Omega_N} g v, \quad \forall v \in H_{\partial\Omega_D}^1(\Omega). \end{cases} \quad (10.15)$$

Robin Boundary Conditions A Robin boundary condition has the form $\gamma u + n \cdot \sigma \cdot \nabla u = g$ on $\partial\Omega$ where $\gamma, g: \partial\Omega \rightarrow \mathbb{R}$. If we compute the integral in equation (10.4), we find the weak formulation

$$\begin{cases} \text{Seek } u \in H^1(\Omega) \text{ such that} \\ a(u, v) + \int_{\partial\Omega} \gamma u v = \int_{\Omega} f v + \int_{\partial\Omega} g v, \quad \forall v \in H^1(\Omega). \end{cases} \quad (10.16)$$

General Form We can summarize these results very simply in a table (given in Table 3). All the weak forms we have considered fit in this framework except for the non-homogeneous Dirichlet problem. The generic form of these methods is given by

$$\begin{cases} \text{Seek } u \in V \text{ such that} \\ a(u, v) = f(v), \quad \forall v \in V, \end{cases} \quad (10.17)$$

where V is a Hilbert space satisfying

$$H_0^1(\Omega) \subset V \subset H^1(\Omega). \quad (10.18)$$

We have a bilinear form $a(\cdot, \cdot)$ on $V \times V$ and a linear form, $f(\cdot)$ defined on V .

10.2 Well-Posedness of the Weak Form

Here we consider the well-posedness of the problem (10.17). We begin with some definitions.

Definition 10.1. (H^1 -Norm) For $u \in H^1(\Omega)$, its norm is given by

$$\|u\|_{H^1(\Omega)} = \left(\int_{\Omega} u^2 + |\nabla u|^2 \right)^{\frac{1}{2}}. \quad (10.19)$$

We specifically note that from this definition, we can conclude that $\|\nabla u\|_{L^2(\Omega)} \leq \|u\|_{H^1(\Omega)}$. This will be useful in proving the following properties:

Definition 10.2. (*Coercivity*) The bilinear form $a(\cdot, \cdot)$ is coercive if $\exists \alpha > 0$ such that

$$a(u, u) \geq \alpha \|u\|_V^2, \quad \forall u \in V. \quad (10.20)$$

Definition 10.3. (*Continuity of the bilinear form*) $a(\cdot, \cdot)$ is continuous if $\exists \gamma > 0$ such that

$$|a(u, v)| \leq \gamma \|u\|_V \|v\|_V, \quad \forall u, v \in V. \quad (10.21)$$

Definition 10.4. (*Continuity of the linear form*) $f(\cdot)$ is continuous if $\exists \delta > 0$ such that

$$|f(v)| \leq \delta \|v\|_V, \quad \forall v \in V. \quad (10.22)$$

In order to demonstrate the well-posedness of the weak variational form, we need to verify the conditions in the Lax-Milgram Lemma (Lemma 10.1), i.e. the bilinear form is continuous (bounded) and coercive and that the linear form is continuous (bounded). In §10.6, we give several inequalities that are useful for finding the bounds needed to verify the Lax-Milgram Lemma. For specific examples, see the solutions set, available as a separate document.

10.3 Formulating a Finite Element Approximation

To begin, we need to divide our computational domain into elements. In one dimension, these elements will be (disjoint) intervals and in two dimensions we generally choose to use a set of triangles, K_i . We call the set of non-overlapping triangles, \mathcal{T}_h , a *triangulation* of the domain $\Omega = K_1 \cup K_2 \cup \dots \cup K_m$. The triangles are chosen such that no vertex of one triangle lies on the edge of another.

Now, given a triangulation of the domain Ω , we construct the subspace on which our solution exists. Generally, we construct a solution that is a piecewise-polynomial on each element. As an example, we can define $V_h \subset V$ such that $V_h = \{v : v \text{ is continuous, } v|_K \text{ is linear } \forall K \in \mathcal{T}_h\}$. We adopt the following notation for $r \in \mathbb{N}$

$$P_r(K) = \{v : v \text{ is a polynomial of degree } \leq r \text{ on } K\}. \quad (10.23)$$

Note that $q = \dim P_r(K) = \frac{1}{2}(r+1)(r+2)$. Let $n_i, i = 1, \dots, N$ be the set of nodes for \mathcal{T}_h . Our solution, $v \in V_h$ will then be described by the values $v(n_i)$. These are the *global degrees of freedom*. We also have *element degrees of freedom*, associated with each node belonging to a single element, $K \in \mathcal{T}_h$. Next we construct a set of nodal basis functions on each element, $\phi_i \in P_r(K)$ such that

$$\phi_i(n_j) = \delta_{ij} \quad (10.24)$$

for n_j a node belonging to K giving us the representation

$$v(x) = \sum_{i=1}^q v(n_i) \phi_i(x), \quad x \in K \quad (10.25)$$

for $v(x) \in P_r(K)$. Now, we have $V_h = \{v : v|_K \in P_r(K) \forall K \in \mathcal{T}_h \text{ with } v \text{ continuous at the nodes}\}$. Functions $v \in V_h$ will be such that $v \in C^0(\bar{\Omega})$. Note that to construct higher order polynomials on the elements, it will be necessary to add additional element degrees of freedom. We can specify additional values at mid-points of the elements or add derivative information at the nodes we already have, resulting in different basis functions. Further, we can construct a basis for the space V_h , $\{\varphi_i\}$, from the nodal basis functions. The important observation here is that φ_i is only non-zero on elements K such that $n_i \in K$. Our finite element method is then given by

$$\begin{cases} \text{Find } u_h \in V_h \text{ such that} \\ a(u_h, v) = (f, v) \quad \forall v \in V_h. \end{cases} \quad (10.26)$$

10.4 The Stiffness Matrix

Now we briefly look at actually solving (10.26). Let $\{\varphi_1, \dots, \varphi_M\}$ be a basis for V_h . Then for $v \in V_h$ we can write

$$v = \sum_{i=1}^M \eta_i \varphi_i, \quad \eta_i \in \mathbb{R}. \quad (10.27)$$

Now, we can write

$$a(u_h, \varphi_i) = L(\varphi_i), \quad i = 1, \dots, M \quad (10.28)$$

where $L(\cdot)$ is our linear form, (f, \cdot) . Given that the solution can be written as

$$u_h = \sum_{i=1}^M \xi_i \varphi_i, \quad \xi_i \in \mathbb{R} \quad (10.29)$$

we can write (10.26) as

$$\sum_{i=1}^M a(\varphi_i, \varphi_j) \xi_i = L(\varphi_j), \quad j = 1, \dots, M \quad (10.30)$$

which can be written in matrix form

$$A\xi = b \quad (10.31)$$

where $\xi, b \in \mathbb{R}^M$ with $a_{ij} = a(\varphi_i, \varphi_j)$ and $b_i = L(\varphi_i)$. Now we make two important observations. Using equation (10.27) we have

$$a(v, v) = a\left(\sum_{i=1}^M \eta_i \varphi_i, \sum_{j=1}^M \eta_j \varphi_j\right) = \sum_{i,j=1}^M \eta_i a(\varphi_i, \varphi_j) \eta_j = \eta \cdot A\eta. \quad (10.32)$$

Since the bilinear form is coercive, we obtain

$$\eta \cdot A\eta = a(v, v) \geq \alpha \|v\|_V^2 > 0 \quad (10.33)$$

showing that the matrix is positive definite. Further, we often have that the bilinear form is symmetric $a(\varphi_i, \varphi_j) = a(\varphi_j, \varphi_i)$. This property is easy to verify if it is indeed true for our problem. If it is, then we have that A is a symmetric positive definite matrix which implies that the matrix problem has a unique solution (and in some sense, is easy to solve). Further, we can see that the matrix A will be sparse since $a(\varphi_i, \varphi_j) = 0$ unless n_i and n_j both belong to the same triangle K . By selecting the proper ordering of the nodes, we can ensure that A is both banded and sparse.

10.5 Error Estimates

In order to obtain error estimates, we need to enforce a condition on the triangulation of the domain. Particularly, we cannot allow the triangles to become arbitrarily thin. Let h_K be the diameter of K (the length of the longest side of K) and ρ_K be the diameter of the largest circle we can inscribe in K . If $h = \max_{K \in \mathcal{T}_h} h_K$, let $\beta > 0$ be a constant independent of h . Then if

$$\frac{\rho_K}{h_K} \geq \beta \quad \forall K \in \mathcal{T}_h \quad (10.34)$$

we can apply some simple error estimates. Now, let $\pi_h u \in V_h$ be an interpolant of u in the appropriate finite element space. We then have the following estimates where the interpolant consist of piecewise polynomials of degree $r \geq 1$.

$$\|u - \pi_h u\|_{L^2(\Omega)} \leq Ch^{r+1} |u|_{H^{r+1}(\Omega)}, \quad (10.35)$$

$$|u - \pi_h u|_{H^1(\Omega)} \leq Ch^r |u|_{H^{r+1}(\Omega)}. \quad (10.36)$$

Now, for elliptic problems, we have the following abstract error estimate,

$$\|u - u_h\|_V \leq C \|u - v\|_V \quad \forall v \in V_h. \quad (10.37)$$

Letting $v = \pi_h u \in V_h$, we can then bound the finite element error by the interpolant error

$$\|u - u_h\|_V \leq C \|u - \pi_h u\|_V. \quad (10.38)$$

10.6 Key Theorems

Theorem 10.1. (*Green's Formula*)

$$-\int_{\Omega} \nabla \cdot (\sigma \nabla u) v = \int_{\Omega} \nabla v \cdot \sigma \cdot \nabla u - \int_{\partial\Omega} (n \cdot \sigma \cdot \nabla u) v \quad (10.39)$$

Theorem 10.2. (*Poincaré's Inequality*) *Let $1 \leq p < \infty$ and let Ω be a bounded open set. Then there exists $c_{p,\Omega} > 0$ such that $\forall v \in W_0^{1,p}(\Omega)$*

$$c_{p,\Omega} \|v\|_{L^p(\Omega)} \leq \|\nabla v\|_{L^p(\Omega)}. \quad (10.40)$$

Theorem 10.3. *Let Ω be a bounded connected open set in \mathbb{R}^n , with sufficiently regular boundary. Then we have for $u \in H^1(\Omega)$, such that $\int_{\Omega} u(x) dx = 0$,*

$$\|u\|_{L^2(\Omega)}^2 \leq P(\Omega) \|\nabla u\|_{L^2(\Omega)}^2. \quad (10.41)$$

Corollary 10.1. $|u|_{H^1(\Omega)} = \|\nabla u\|_{L^2(\Omega)}$ *is a norm equivalent with the norm $\|u\|_{H^1(\Omega)}$ on the subspace V_0 (closed in $H^1(\Omega)$) defined by*

$$V_0 = \{u \in H^1(\Omega) : \int_{\Omega} u(x) dx = 0\}. \quad (10.42)$$

Corollary 10.2. *Let Ω be a bounded connected open set in \mathbb{R}^n with sufficiently regular boundary Γ . Suppose $\Gamma = \Gamma_1 \cup \Gamma_2$ with length (area) of $\Gamma_2 > 0$. Let*

$$V_{\Gamma_2} = \{u \in H^1(\Omega) : u|_{\Gamma_2} = 0\}. \quad (10.43)$$

Then V_{Γ_2} is a closed subspace of $H^1(\Omega)$ and $|u|_{H^1(\Omega)} = \|\nabla u\|_{L^2(\Omega)}$ is a norm equivalent with the norm $\|u\|_{H^1(\Omega)}$ on the subspace V_{Γ_2} .

Corollary 10.3. *Let Ω be an open and bounded domain with Lipschitz-continuous boundary $\Gamma = \partial\Omega$. Then there is a positive constant C such that*

$$\|u|_{\Gamma}\|_{L^2(\Gamma)} \leq C \|u\|_{H^1(\Omega)}. \quad (10.44)$$

Lemma 10.1. (*Lax-Milgram*) *Let a be a coercive, bounded bilinear form on a Hilbert space V and f be a bounded linear form on V , then there exists a unique $u \in V$ such that*

$$a(u, v) = f(v) \quad \forall v \in V. \quad (10.45)$$

Further, for all $f \in V$, we have

$$\|u\|_V \leq \frac{1}{\alpha} \|f\|_V \quad (10.46)$$

where α is the constant from the coercivity bound.

10.7 Qual Problems

There is one finite element problem on every qualifying exam. Generally it is the last problem, #7.

11 References

The material for these notes was taken primarily from the sources listed below.

General Numerical Analysis

K. E. ATKINSON, *An Introduction to Numerical Analysis*, John Wiley & Sons, second ed., 1989.

R. L. BURDEN AND J. D. FAIRES, *Numerical Analysis*, Brooks/Cole, seventh ed., 2001.

Numerical Linear Algebra

G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, third ed., 1996.

Numerical Differentiation

B. FORNBERG, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, 2005.

Numerical Methods for ODEs

K. E. ATKINSON, *An Introduction to Numerical Analysis*, John Wiley & Sons, second ed., 1989.

L. SHAMPINE, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, 1994.

Finite Differences for PDEs

J. STRIKWERDA, *Finite Difference Schemes and Partial Differential Equations*, SIAM, second ed., 2004.

Finite Element Method

A. ERN AND J.-L. GUERMOND, *Theory and Practice of Finite Elements*, Springer-Verlag, 2004.

C. JOHNSON, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Studentlitteratur, 1987.