Numerical Methods

April 2, 2012

Notes

Notes

- There will be a problem session on Tuesday at 7pm in MAGC 1.324
- Practice Problems:

19.1	1-19
19.2	1-7, **9-18**

Notes

19.1 Numerical Methods

We have seen, particularly in this course for finding eigenvalues of a $n \times n$ matrix, that it is important in **many** applications to find the roots of a polynomial of degree n.

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$

- As early as 2000 BC we have stone records of Babylonians solving quadratic equations.

Notes

19.1 Numerical Methods

We have seen, particularly in this course for finding eigenvalues of a $n \times n$ matrix, that it is important in **many** applications to find the roots of a polynomial of degree n.

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$

- Indian mathematicians found versions of the quadratic formula in the 8th century BC. (explicit general solution was written down finally by Brahmagupta in 628 AD).
- Chinese mathematicians in 200 BC (via completing the square).
- Euclid (Greece) gave an interpretation of the problem (and solution) using geometry.

Notes

19.1 Numerical Methods

We have seen, particularly in this course for finding eigenvalues of a $n \times n$ matrix, that it is important in **many** applications to find the roots of a polynomial of degree n.

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$

- Babylonians, Egyptians, and Diophantus (Greece) found solutions of some cubic equations.
- Niccol Tartaglia (Italian, 1500-1557) gave a formula for general solutions of cubic equations.
- Formulas were shortly found for giving general solutions of quartic equations. But at that point progress on the question seems to come to a halt.

Notes

19.1 Numerical Methods

We have seen, particularly in this course for finding eigenvalues of a $n \times n$ matrix, that it is important in **many** applications to find the roots of a polynomial of degree n.

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$

- French mathematician Évariste Galois (1811-1832) became very intrigued by the question of why progress on the quintic and higher degree polynomial roots seemed so difficult. His life's work (he only lived to be 21!!) was a mathematical proof that there are **only** formulas for the solutions of general polynomials of degree n for n=1, 2, 3, 4.
- Note – he was a radical in France, and died from wounds suffered in a duel.
- If you are interested in this subject, you should be a math dual-major or minor, and you should take the Modern Algebra course. **see me we can talk**

Notes

19.1 Numerical Methods

We have seen, particularly in this course for finding eigenvalues of a $n \times n$ matrix, that it is important in **many** applications to find the roots of a polynomial of degree n.

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$

- In the 18th and 19th century it was becoming very important for physical and engineering applications to be able to find the roots of a polynomial of large degree.
- There are no formulas for solving them. (Galoise showed this).
- There are rarely rational roots. (Diophantus was showing this circa 100 AD).
- The only practical thing to try is to approximate the roots.
- Using graphs is really not a very practical solution.
- We need methods which are automatic, and algorithmic (can be written in a code in our favorite language).

Notes

Errors

Our computer can store only finitely many decimal digits. Without further modification our personal computers we typically use have a maximum of 15 digits to store for each number.

But here is the problem: If we have two numbers which have already had an error introduced and try to divide them by each other the resulting error could potentially be much larger.

Notes

Example 1

Consider

$$\frac{1}{30} \cong 0.03$$

with an error of $0.0033\bar{3} \cong 3 \times 10^{-3}$.

Notice what happens if we do the following:

$$\begin{aligned} 0 &= 3000 - 3000 \\ &= 100 \times 30 - 3000 \\ &= \frac{100}{1/30} - 3000 \\ &\cong \frac{100}{0.03} - 3000 \\ &\cong 3333.33 - 3000 \\ &= 333.33 \end{aligned}$$

At the first congruent step we introduced an error of size 10^{-3} **in the denominator**, leading to an error of size 10^{3} in the answer.

Notes

Errors

In general you should avoid: dividing by small numbers, multiplying by large numbers, taking the difference of large numbers, and adding together **too** many numbers.

This last point is important to remember: There is going to be a limit to how accurate our numerical methods can be.

The limit is imposed by what will happen if we run the algorithm for too long — errors will accumulate and become significant.

Notes

Stability

A related issue to numerical errors for our algorithms will be **stability**.

An algorithm is **stable** if for small changes to the input problem the output only changes by a small amount.

Unstable algorithms are bad as they imply that if the input conditions have errors we could reach the wrong conclusions.

As most of our algorithms are going to be iterative, unstable algorithms will also allow errors to accumulate rapidly.

Notes

19.2 Solutions of Equations

The absolute most general problem we want to consider is

Solving Equation

Given a function $f(x)$ determine values of x such that

$$f(x) = 0$$

Bonus points for giving an estimate of how close our approximate solution is to the actual solution.

Notes

Bissection Method

This is what I call a **Brute Force** method: Suppose f is a continuous function.
Step 1: Find a and b such that $f(a)$ and $f(b)$ have opposite signs. There must be a zero between a and b.

Step 2: Let c be the midpoint of a and b. If $f(c) = 0$ we are lucky. Otherwise take c and either a or b so that $f(c)$ and $f(a)$ or $f(b)$ have the opposite sign.

Step 3: Continue halving the size of the interval.

This simple idea is almost useless in Engineering applications, but it has been used as a mathematical tool to provide some startling insights into numbers themselves.

Notes

Fixed Point Iteration

Consider the related problem of finding a solution to $g(x) = x$.
Example: For an example take $g(x) = \frac{1}{3}\left[x^2 + 1\right]$.

1. Start by sketching a graph of $y = g(x)$ and $y = x$. What does the equation $g(x) = x$ mean on our graph?

Notes

Fixed Point Iteration

Consider the related problem of finding a solution to $g(x) = x$.
Example: For an example take $g(x) = \frac{1}{3}\left[x^2 + 1\right]$.

1. Start by sketching a graph of $y = g(x)$ and $y = x$. What does the equation $g(x) = x$ mean on our graph?

2. Choose an initial seed x_0. This is any number. Find the point $(x_0, g(x_0))$ on your graph.

Notes

Fixed Point Iteration

Consider the related problem of finding a solution to $g(x) = x$.
Example: For an example take $g(x) = \frac{1}{3}\left[x^2 + 1\right]$.

1. Start by sketching a graph of $y = g(x)$ and $y = x$. What does the equation $g(x) = x$ mean on our graph?

2. Choose an initial seed x_0. This is any number. Find the point $(x_0, g(x_0))$ on your graph.

3. Use the height of your point $g(x_0)$ to give you your next x-value: $x_1 = g(x_0)$. Find that point on your graph.

Notes

Fixed Point Iteration

Consider the related problem of finding a solution to $g(x) = x$.
Example: For an example take $g(x) = \frac{1}{3}\left[x^2 + 1\right]$.

1. Start by sketching a graph of $y = g(x)$ and $y = x$. What does the equation $g(x) = x$ mean on our graph?

2. Choose an initial seed x_0. This is any number. Find the point $(x_0, g(x_0))$ on your graph.

3. Use the height of your point $g(x_0)$ to give you your next x-value: $x_1 = g(x_0)$. Find that point on your graph.

4. Continue this process until you are confident that (a) The iteration converges to the solution or (b) The iteration will never converge to the solution.

Notes

Fixed Point Iteration

Consider the related problem of finding a solution to $g(x) = x$.
Example: For an example take $g(x) = \frac{1}{3}\left[x^2 + 1\right]$.

5. For what values of your initial seed will the answer converge? Can you find the larger root?

Notes

Fixed-Point Iteration

Theorem

Convergence of Fixed-Point Iteration Let $x = s$ be a solution of $x = g(x)$ and suppose that g has continuous derivative in some interval J containing s. Then if $|g'(x)| \leq K < 1$ in J then the iteration process $x_{n+1} = g(x_n)$ converges to s for any x_0 in J.

Example:
For the example above: $g'(x) = 2/3x$ which is bigger than 1 in absolute value for $x > 3/2$. In particular, with the second root at 2.618 there is no seed we can use to find it using this method.

Notes

Example: Fixed Point Iteration

Use the method iteration method to find a solution of $f(x) = \cos(x) + \frac{1}{3} = 0$.

You will first need to transform the equation to one of the form $g(x) = x$. There are many ways to do this.

Notes

Newtons-Method

The presence of the derivative in the Theorem for Fixed-Point iteration actually hints at a method we can use to approximate solutions of $f(x) = 0$ by iteration.

Consider the graph of $y = f(x)$. Provided that the zero $f(x)$ is **nice** note that the tangent lines through points near the zero will intersect the horizontal axis closer to the zero.

Notes

Newtons-Method

Consider the graph of $y = f(x)$. Provided that the zero $f(x)$ is **nice** note that the tangent lines through points near the zero will intersect the horizontal axis closer to the zero.

1. Given a x_0, find the x-intercept of the tangent line to $y = f(x)$ at $(x_0, f(x_0))$ as a function of x_0.

Notes

Newtons-Method

We find

$$f'(x_0) = \frac{\text{rise}}{\text{run}} = \frac{f(x_0) - 0}{x_0 - x_1}$$

So we can solve for x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Notes

Newtons-Method

Given a function f with a continuous non-zero first derivative f' in some interval J.

Step 1 Choose an initial seed x_0.

Step 2 define the iterates

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

Step 3 Run the algorithm until the relative error:

$$\epsilon = \frac{|x_{n+1} - x_n|}{|x_n|}$$

is small enough.

Otherwise if the algorithm reaches the maximum number of iterations, N, produce a warning/error that the algorithm has failed to converge successfully.

Notes

Example:

Problem

Find a root of $f(x) = 2\sin(x) - x$ with initial seed $x_0 = 3$ to a tolerance of 10^{-3} with at most $N = 10$ iterations.

Notes

Order of an Iteration Method

Consider an iteration method $x_{n+1} = g(x_n)$ with solution s, so that $s = g(s)$.

Let $\epsilon_n = x_n - s$ denote the error of x_n. Then Taylors Formula gives:

$$\begin{aligned} x_{n+1} &= g(x_n) \\ &= g(s) + g'(s)(x_n - s) + \frac{1}{2}g''(s)(x_n - s)^2 + \dots \\ &= g(s) + g'(s)\epsilon_n + \frac{1}{2}g''(s)\epsilon_n^2 + \dots \end{aligned}$$

subtracting s from both sides we have:

$$\epsilon_{n+1} = g'(s)\epsilon_n + \frac{1}{2}g''(s)\epsilon_n^2 + \dots$$

If $g'(s) \neq 0$ then we call the method **First Order**.

If both $g'(s) = 0$ and $g''(s) \neq 0$ then we call the method **Second Order** and so on.

Notes

Order of Newtons Method

The iterator for Newtons Method is:

$$g(x) = x - \frac{f(x)}{f'(x)}$$

The first derivative is

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}$$

Notes

Convergence of Newtons Method

Indeed setting $f(s) = 0$ we find that $g'(s) = 0$.

Theorem

Convergence of Newtons Method If $f(x)$ has 3 continuous derivatives and f' and f'' are not zero at a solution s of $f(x) = 0$, then for x_0 sufficiently close to s, Newtons Method is of **Second Order**

Notes

Secant Method

The problem with Newtons Method is that we need to know the derivative of $f(x)$, and it should be well behaved.

We could use a Secant Line instead of a Tangent Line.

Here we seed two values: x_0 and x_1, and then choose the third x_2 to be the intersection of the secant line with the horizontal axis.

We have

$$\text{slope} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

so the intersection is

$$x_2 = x_1 - f(x_1)\frac{(x_1 - x_0)}{f(x_1) - f(x_0)}$$

Notes

Secant Method

In general then the iterator is:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

This has similar issues as Newtons Method: the denominator could be zero, it is not guaranteed to converge. Except in special circumstances it will only be a first order method.

Notes

Second Programing Assignment

You may do this assignment with a partner. The advantage of the secant method is that you do not need to compute a derivative. Write a C++, Matlab or similar script for using the Secant method to find a zero of a function.
Rubric:

- Your code should compile
- It should be easy to change the function
- The user could choose the initial seeds, or the program can
- It should compute the root to a specified error tolerance.
- It should have a maximum number of iterations (no infinite loops) and produce a warning if it reaches this number without being within the specified error.
- It should produce an error message if it finds a horizontal secant line
- You will receive bonus points if your program searches for multiple roots automatically.

Notes

Notes