Concerned with the application of mathematics to the development of constructive methods for the approximate solution of mathematical problems. (Mathematical problems, in turn, come from mathematical models of “physical” processes.)

Concerned with the application of mathematics to the development of constructive methods for the approximate solution of mathematical problems. (Mathematical problems, in turn, come from mathematical models of “physical” processes.)

Borrows ideas from many areas of mathematics including such “pure” areas as functional analysis, algebra, algebraic topology, etc., but also requires the development of an entirely new set of techniques. Classical results must also be used carefully when they are applicable.

Example

A polynomial of degree n is known to have n roots, real and complex. It is also known that there is no formula (analogous to the quadratic formula) for finding the roots of a general polynomial of degree $n > 4$. So, how are these roots to be found? (Note that even the quadratic formula has its pitfalls, as we shall see, unless carefully applied).

Example

Find

$$\int_a^b f(x)\,dx,$$

where f is a continuous function on $[a, b]$.

Example

Find

$$\int_a^b f(x)\,dx,$$

where f is a continuous function on $[a, b]$. By the Fundamental Theorem of Calculus,

$$\int_a^b f(x)\,dx = F(b) - F(a),$$

where $F(x)$ is an antiderivative of $f(x)$ (and $F(x)$ is guaranteed to exist).

Example

Find

$$\int_a^b f(x)\,dx,$$

where f is a continuous function on $[a, b]$. By the Fundamental Theorem of Calculus,

$$\int_a^b f(x)\,dx = F(b) - F(a),$$

where $F(x)$ is an antiderivative of $f(x)$ (and $F(x)$ is guaranteed to exist).

However, e.g.

$$\int_a^b e^{-x^2}\,dx$$

(an important integral in probability) cannot be found this way, since the antiderivative of e^{-x^2} cannot be expressed in terms of standard elementary functions. Thus,

$$\int_a^b e^{-x^2}\,dx$$

(along with a large proportion of the definite integrals encountered in practice) must be found numerically.

Example

Solve the initial value problem:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

Example

Solve the initial value problem:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

Existence and uniqueness theorems are available, along with lots of "tricks" for solving such problems analytically. The tricks almost never apply in practice, and although it is important to know the problem has a unique solution, such theorems provide no practical clue as to how to go about finding the solution. Numerical solutions are an important tool for understanding differential equations.

Example

Solve $Ax = b$ where A is an $n \times n$ matrix, and $\det A \neq 0$. Cramer's rule, from elementary linear algebra, gives that the problem has a unique solution with

$$x_i = \frac{\det A_i}{\det A}, \; 1 \leq i \leq n$$

Example

Solve $Ax = b$ where A is an $n \times n$ matrix, and $\det A \neq 0$. Cramer's rule, from elementary linear algebra, gives that the problem has a unique solution with

$$x_i = \frac{\det A_i}{\det A}, \ 1 \leq i \leq n$$

Suppose $n = 26$. Then each determinant (using the definition) involves the sum of 26! products of 26 elements each, or $25 \times 26!$ multiplies alone, and there are 27 determinants to be calculated. If we could do 1 billion multiplies/second, it would take 9 trillion years to find the solution. Moreover, because of the huge number of calculations involved, the accumulation of roundoff errors that would result would make the final answer useless.

Example

Solve $Ax = b$ where A is an $n \times n$ matrix, and $\det A \neq 0$. Cramer's rule, from elementary linear algebra, gives that the problem has a unique solution with

$$x_i = \frac{\det A_i}{\det A},\ 1 \leq i \leq n$$

Suppose $n = 26$. Then each determinant (using the definition) involves the sum of 26! products of 26 elements each, or $25 \times 26!$ multiplies alone, and there are 27 determinants to be calculated. If we could do 1 billion multiplies/second, it would take 9 trillion years to find the solution. Moreover, because of the huge number of calculations involved, the accumulation of roundoff errors that would result would make the final answer useless.

On the other hand, elimination produces the solution in $\frac{1}{3}n^3$ multiplies. When $n = 26$, this is approximately 5859 multiplies, which can be done in a fraction of a second, with controllable roundoff error.

Example

Solve $Ax = b$ where A is an $n \times n$ matrix, and $\det A \neq 0$. Cramer's rule, from elementary linear algebra, gives that the problem has a unique solution with

$$x_i = \frac{\det A_i}{\det A}, \; 1 \leq i \leq n$$

Suppose $n = 26$. Then each determinant (using the definition) involves the sum of 26! products of 26 elements each, or $25 \times 26!$ multiplies alone, and there are 27 determinants to be calculated. If we could do 1 billion multiplies/second, it would take 9 trillion years to find the solution. Moreover, because of the huge number of calculations involved, the accumulation of roundoff errors that would result would make the final answer useless.

On the other hand, elimination produces the solution in $\frac{1}{3}n^3$ multiplies. When $n = 26$, this is approximately 5859 multiplies, which can be done in a fraction of a second, with controllable roundoff error.

Another approach would be to find A^{-1} and then $x = A^{-1}b$. This would seem particularly attractive if one needed to solve $Ax_i = b$, for $1 \leq i \leq 1000$, for example. But it turns out, for reasons of efficiency and accuracy, that A^{-1} should never be computed for the purpose of solving $Ax = b$. There are situations, however, when one may want to find A^{-1} for other reasons.

MACHINE NUMBERS

Another class of problems arises when using a computer because one is computing in finite-precision arithmetic, rather then using the real number system. The normalized floating-decimal representation of a number a is given by

$$a = \pm q \times 10^e,$$

where $.1 \leq q < 1$ (the *mantissa*), and e (the *exponent*) is an integer. For example, $a = 0.0003288$ is represented by $a = .3288 \times 10^{-3}$.

MACHINE NUMBERS

Another class of problems arises when using a computer because one is computing in finite-precision arithmetic, rather then using the real number system. The normalized floating-decimal representation of a number a is given by

$$a = \pm q \times 10^e,$$

where $.1 \leq q < 1$ (the *mantissa*), and e (the *exponent*) is an integer. For example, $a = 0.0003288$ is represented by $a = .3288 \times 10^{-3}$.

A computer can only store a finite number of digits, so it has to chop off the tail somewhere. In a base 10 computer, a is approximated by the number $\bar{a} = \pm\bar{q} \times 10^e$, where $\bar{q}$ is q rounded off or chopped to t decimal places. Typically, computers work with the binary system (base 2). (We will talk more about these in the next lecture.)

MACHINE NUMBERS

Another class of problems arises when using a computer because one is computing in finite-precision arithmetic, rather then using the real number system. The normalized floating-decimal representation of a number a is given by

$$a = \pm q \times 10^e,$$

where $.1 \leq q < 1$ (the *mantissa*), and e (the *exponent*) is an integer. For example, $a = 0.0003288$ is represented by $a = .3288 \times 10^{-3}$.

A computer can only store a finite number of digits, so it has to chop off the tail somewhere. In a base 10 computer, a is approximated by the number $\bar{a} = \pm\bar{q} \times 10^e$, where $\bar{q}$ is q rounded off or chopped to t decimal places. Typically, computers work with the binary system (base 2). (We will talk more about these in the next lecture.)

The number t of digits in the mantissa $\bar{q}$ is called the *significance*. The exponent e is restricted in size, i.e., $M_1 \leq e \leq M_2$. The fact that we are only dealing with a **finite** set of numbers causes a new set of problems.

Although integers can be represented exactly (provided they are not too large), other numbers often are not. This can result in errors in the representation of a problem on the machine (input errors), which in turn can lead to drastic errors in the solution.

Although integers can be represented exactly (provided they are not too large), other numbers often are not. This can result in errors in the representation of a problem on the machine (input errors), which in turn can lead to drastic errors in the solution.

The sum of two machine numbers, for example, is found by adding and then rounding or chopping the result to t places, leading to a *roundoff error*. The accumulation of many of these small errors can pose a serious problem.

Although integers can be represented exactly (provided they are not too large), other numbers often are not. This can result in errors in the representation of a problem on the machine (input errors), which in turn can lead to drastic errors in the solution.

The sum of two machine numbers, for example, is found by adding and then rounding or chopping the result to t places, leading to a *roundoff error*. The accumulation of many of these small errors can pose a serious problem.

Some of the properties of real numbers that we take for granted are no longer valid. For example, the commutative and associative laws for addition hold true for real numbers, and imply that

$$a + b + c = b + a + c.$$

While the commutative law is still valid in floating point arithmetic, the associative law is not, and hence the order of addition of a collection of numbers becomes important.

As an example, suppose we add the numbers $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, whose true sum is 55, using a "computer" with base 10 (decimal), with only one significant digit. Then, adding left to right,

As an example, suppose we add the numbers $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, whose true sum is 55, using a "computer" with base 10 (decimal), with only one significant digit. Then, adding left to right,

	round	chop
$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$	70	20
$10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1$	60	10
$6 + 5 + 7 + 4 + 8 + 3 + 9 + 2 + 10 + 1$	50	20

As an example, suppose we add the numbers $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$, whose true sum is 55, using a "computer" with base 10 (decimal), with only one significant digit. Then, adding left to right,

	round	chop
$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$	70	20
$10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1$	60	10
$6 + 5 + 7 + 4 + 8 + 3 + 9 + 2 + 10 + 1$	50	20

Adding small numbers to large numbers is an excellent way to create errors. For example, if $a = 1$, and $b = .00004$, both a and b have 1 significant digits. But $a + b = 1$ if $t < 5$. When you have a code that adds many small numbers to larger numbers, this can be a source of significant error. This situation arises often in the solution of differential equations, as we shall see.

CANCELLATION

CANCELLATION

Subtracting two numbers of nearly equal size can lead to serious loss of significance. If

$$a = .123500 \quad \text{and } b = .123499$$

for example, then using base 10 and $t = 3$, we get

$$a - b = .001$$

rather than .000001.

CANCELLATION

Subtracting two numbers of nearly equal size can lead to serious loss of significance. If

$$a = .123500 \quad \text{and } b = .123499$$

for example, then using base 10 and $t = 3$, we get

$$a - b = .001$$

rather than .000001. If we take $t = 6$, then

$$a - b = .1 \times 10^{-5}$$

and only the first digit is now significant.

Suppose we need to calculate $f(x) = \sqrt{x^2+1} - 1$ for $x = .20$. The exact value is .0198. Using $t = 2$, we get $f(.20) = 0$, which is 100% in error.

Suppose we need to calculate $f(x) = \sqrt{x^2+1} - 1$ for $x = .20$. The exact value is .0198. Using $t = 2$, we get $f(.20) = 0$, which is 100% in error.

On the other hand, $f(x) = \dfrac{x^2}{\sqrt{x^2+1}+1}$, and using this form with $t = 2$, we get $f(.20) = .02$ - about a 1% error.

CONDITIONING AND INPUT ERRORS

CONDITIONING AND INPUT ERRORS

Consider the matrix $A = \begin{pmatrix} 1 & .99 \\ .99 & .98 \end{pmatrix}$, and solve $Ax = b$, and $A\bar{x} = \bar{b}$, where

$$b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \quad \text{and} \quad b - \bar{b} = \begin{pmatrix} .000097 \\ .000106 \end{pmatrix}.$$

CONDITIONING AND INPUT ERRORS

Consider the matrix $A = \begin{pmatrix} 1 & .99 \\ .99 & .98 \end{pmatrix}$, and solve $Ax = b$, and $A\bar{x} = \bar{b}$, where

$$b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \quad \text{and} \quad b - \bar{b} = \begin{pmatrix} .000097 \\ .000106 \end{pmatrix}.$$

The true solutions are

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \bar{x} = \begin{pmatrix} 3 \\ -1.0203 \end{pmatrix},$$

with $x - \bar{x} = \begin{pmatrix} -2 \\ 2.0203 \end{pmatrix}$!!

CONDITIONING AND INPUT ERRORS

Consider the matrix $A = \begin{pmatrix} 1 & .99 \\ .99 & .98 \end{pmatrix}$, and solve $Ax = b$, and $A\bar{x} = \bar{b}$, where

$$b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \quad \text{and} \quad b - \bar{b} = \begin{pmatrix} .000097 \\ .000106 \end{pmatrix}.$$

The true solutions are

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \bar{x} = \begin{pmatrix} 3 \\ -1.0203 \end{pmatrix},$$

with $x - \bar{x} = \begin{pmatrix} -2 \\ 2.0203 \end{pmatrix}$!!

This problem is called ill-conditioned, meaning that small relative changes in input can cause large relative changes in the solution. In this case, the change in the solution is approximately 20000 times the change in the right hand side vector b.

CONDITIONING AND INPUT ERRORS

Consider the matrix $A = \begin{pmatrix} 1 & .99 \\ .99 & .98 \end{pmatrix}$, and solve $Ax = b$, and $A\bar{x} = \bar{b}$, where

$$b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \quad \text{and} \quad b - \bar{b} = \begin{pmatrix} .000097 \\ .000106 \end{pmatrix}.$$

The true solutions are

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \bar{x} = \begin{pmatrix} 3 \\ -1.0203 \end{pmatrix},$$

with $x - \bar{x} = \begin{pmatrix} -2 \\ 2.0203 \end{pmatrix}$!!

This problem is called ill-conditioned, meaning that small relative changes in input can cause large relative changes in the solution. In this case, the change in the solution is approximately 20000 times the change in the right hand side vector b.

As interpretations:

CONDITIONING AND INPUT ERRORS

Consider the matrix $A = \begin{pmatrix} 1 & .99 \\ .99 & .98 \end{pmatrix}$, and solve $Ax = b$, and $A\bar{x} = \bar{b}$, where

$$b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \quad \text{and} \quad b - \bar{b} = \begin{pmatrix} .000097 \\ .000106 \end{pmatrix}.$$

The true solutions are

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \bar{x} = \begin{pmatrix} 3 \\ -1.0203 \end{pmatrix},$$

with $x - \bar{x} = \begin{pmatrix} -2 \\ 2.0203 \end{pmatrix}$!!

This problem is called ill-conditioned, meaning that small relative changes in input can cause large relative changes in the solution. In this case, the change in the solution is approximately 20000 times the change in the right hand side vector b.

As interpretations:

If $\bar{b}$ is measured experimentally with an accuracy of $\pm.0001$ or less, trouble.

It would make <u>no sense whatever</u> to attempt to solve on a 3-figure machine.

CONDITIONING AND INPUT ERRORS

Consider the matrix $A = \begin{pmatrix} 1 & .99 \\ .99 & .98 \end{pmatrix}$, and solve $Ax = b$, and $A\bar{x} = \bar{b}$, where

$$b = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} 1.989903 \\ 1.970106 \end{pmatrix}, \quad \text{and} \quad b - \bar{b} = \begin{pmatrix} .000097 \\ .000106 \end{pmatrix}.$$

The true solutions are

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \bar{x} = \begin{pmatrix} 3 \\ -1.0203 \end{pmatrix},$$

with $x - \bar{x} = \begin{pmatrix} -2 \\ 2.0203 \end{pmatrix}$!!

This problem is called ill-conditioned, meaning that small relative changes in input can cause large relative changes in the solution. In this case, the change in the solution is approximately 20000 times the change in the right hand side vector b.

As interpretations:

If $\bar{b}$ is measured experimentally with an accuracy of $\pm.0001$ or less, trouble.

It would make <u>no sense whatever</u> to attempt to solve on a 3-figure machine.

The problem in the above example is that it is *ill-conditioned*. $\det(A) = -.0001 \neq 0$, so A is invertible. But just barely!

APPROXIMATE SOLUTIONS

APPROXIMATE SOLUTIONS

In real arithmetic, x is the solution of $Ax = b$ iff $r = Ax - b = 0$. The vector r is called the *residual*.

APPROXIMATE SOLUTIONS

In real arithmetic, x is the solution of $Ax = b$ iff $r = Ax - b = 0$. The vector r is called the *residual*.

Consider the system:

$$.780x + .563y = .217$$
$$.913x + .659y = .254$$

Someone proposes two approximate solutions:

$$(x_1, y_1) = (.999, -1.001), \quad \text{and} \quad (x_2, y_2) = (.341, -.087)$$

Which is better?

APPROXIMATE SOLUTIONS

In real arithmetic, x is the solution of $Ax = b$ iff $r = Ax - b = 0$. The vector r is called the *residual*.

Consider the system:

$$\begin{aligned} .780x + .563y &= .217 \\ .913x + .659y &= .254 \end{aligned}$$

Someone proposes two approximate solutions:

$$(x_1, y_1) = (.999, -1.001), \quad \text{and} \quad (x_2, y_2) = (.341, -.087)$$

Which is better?

Since

$$A\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} - b = \begin{pmatrix} .001343 \\ .001572 \end{pmatrix}, \quad \text{and} \quad A\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - b = \begin{pmatrix} .000001 \\ 0 \end{pmatrix},$$

if we judged approximations solely on the size of the residual, we would consider (x_2, y_2) to be the better solution. However, the true solution is $(1, -1)$!!

APPROXIMATE SOLUTIONS

In real arithmetic, x is the solution of $Ax = b$ iff $r = Ax - b = 0$. The vector r is called the *residual*.

Consider the system:

$$\begin{aligned} .780x + .563y &= .217 \\ .913x + .659y &= .254 \end{aligned}$$

Someone proposes two approximate solutions:

$$(x_1, y_1) = (.999, -1.001), \quad \text{and} \quad (x_2, y_2) = (.341, -.087)$$

Which is better?

Since

$$A\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} - b = \begin{pmatrix} .001343 \\ .001572 \end{pmatrix}, \quad \text{and} \quad A\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - b = \begin{pmatrix} .000001 \\ 0 \end{pmatrix},$$

if we judged approximations solely on the size of the residual, we would consider (x_2, y_2) to be the better solution. However, the true solution is $(1, -1)$!!

Moral: Beware of ill-conditioned matrices!