CSCD37H – Analysis of Numerical Algorithms for Computational Mathematics

Wayne Enright

enright@cs.utoronto.ca

Department of Computer Science

University of Toronto

General Information and Math Review

1. What is Numerical Analysis ?
   Why do we need to approximate?
2. Notation and Mathematical Review :
   - Floating point arithmetic
   - Linear Algebra: Notation and Review of key results.
   - Calculus: Notation and Review of key results.

What is Numerical Analysis?

- Consider the investigation of a well defined mathematical model arising in any application area. Assume the model is 'well defined' in the sense that there exists a 'solution' and it is unique. Examples include modelling the spread of an infectious disease, modelling cancer treatments, or modelling the pricing of 'options'.
- We are interested in the 'Conditioning' (or sensitivity) of the underlying mathematical problem to 'small' changes in the problem definition.
- For virtually all mathematical models of practical interest one cannot determine a useful 'closed form' expression for the exact solution and one must approximate the exact solution.

Scientific Computing

1. Formulate a mathematical model of the problem.
2. Approximate the solution of the model.
3. Visualize the approximate solution.
4. Verify that the approximate solution is consistent with the model.
5. Verify that the model is well-posed.

In this course we will focus on developing, analysing and evaluating software/methods for addressing 2.

Focus of Numerical Analysis

The emphasis is on the development and analysis of algorithms to approximate the exact solution to mathematical models.

- Algorithms must be constructive and finite .
- We must analyse the errors in the approximation.
- We must also quantify the stability and efficiency of the algorithms.

Numerical Analysis (cont)

We will be concerned with the intelligent use of existing algorithms embedded in widely used numerical software. We will not spend much time on developing algorithms or on writing code.

- How to interpret the numerical (approximate) results.
- What method (algorithm) should be used.
- What methods are available in the usual ‘Problem Solving Environments’ that scientists, engineers and students work in. For example in MATLAB, MAPLE or Mathematica.
- In order to appreciate the limitations of the methods we must analyse and understand the underlying algorithms on which the methods are based.

A Famous Numerical Algorithm

For details see [A $25 Billion Dollar Eigenvector Algorithm, SIAM Review, September 2006, pp. 569-581.]
The development of an algorithm for the Google Search Engine.

- Locate and access all public web pages.
- Identify those pages that satisfy a search criteria. Let this set of pages be $p_1, p_2, \ldots p_n$.
- Rank these 'hit pages' in order of their importance.
  1. The important pages (the most relevant) must be listed first.
  2. This ordering is accomplished using a Page Rank Algorithm.
  3. A 'score', x_i, is assigned to each page, $p_i, i = 1, 2 \ldots n$ with $x_i \geq 0$.
- Pages are returned (listed) in order of decreasing scores.

Google Search Algorithm

Use a directed graph, G, to represent the set of all pages with vertices, $v_1, v_2 \ldots v_n$ and an edge (v_i, v_k) iff p_i has a link to p_k. Assume that a page is important (an authority) if several pages link to it. For example consider the case where there are four hit pages represented by, v_1, v_2, v_3, v_4, where the first page has links to p_2, p_3, p_4; the second page has links to to p_3, p_4; the third page has a link to p_1; and the fourth page has links to p_1, p_3. G can be represented by its adjacency (or incidence) matrix, A:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

Search Algorithm (cont)

With this representation of G, and with, $e = (1, 1 \ldots 1)^T$ we have:

1. $A\ e\ = (m_1, m_2 \ldots m_n)^T$, where m_i is the number of pages that p_i links to and the i^{th} component of $A^T\ e$ is the number of pages that have links to p_i.
2. If x_i is the score of p_i and $x = (x_1, x_2 \ldots x_n)^T$ then,

$$y \ \equiv \ A^T\ x \ = \ (y_1, y_2 \ldots y_n)^T,$$

   where y_i is the sum of all the scores of pages that link to p_i.
3. Therefore, a natural definition for x_i is y_i. That is, the vector x is identified by $x \ = \ A^T x$ or $A^T x \ = \ x$. This implies that x is an eigenvector of A^T corresponding to the eigenvalue $\lambda \ = \ 1$.

The Google Algorithm (Cont)

This definition for x_i gives too much influence to those pages with lots of links (m_i large) and we can improve the measure of importance by modifying our definition of G and A, by assigning a weight of $1/m_i$ to the edge from v_i to v_k (if it exists). That is, for the above example, A is modified and becomes:

$$A = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{bmatrix}.$$

The Google Algorithm (Cont)

With this modified definition (for A as well as the corresponding x and y) we will always have:

1. The rows of A sum to 1. That is, $A\,e \;=\; e$, and therefore 1 is an eigenvalue of A with an associated eigenvector $v = e$.
2. The corresponding importance of $p_i,\; y_i$, is then (as above), the i^{th} component of $y \;=\; A^T\,x$.

We have shown that an appropriate score vector, x, is the solution of:

$$x \;=\; A^T x \text{ or } A^T x \;=\; x.$$

The Google Algorithm (Cont)

Note that $\lambda = 1$ is an eigenvalue of A^T, since it is an eigenvalue of A. As a result of this observation, a suitable Page Rank Algorithm can be designed based on finding an eigenvector of A^T corresponding to the 'known' eigenvalue $\lambda = 1$.
Questions:

1. Is such an x unique and does it matter?
2. Will the resulting scores all be non-negative (ie. $x_i \geq 0$)?
3. Is there a fast algorithm for computing x?

For the above example, with $n = 4$, an eigenvector is $v = (12, 4, 9, 6)^T$, which when normalised becomes, $x = v/\|v\|_2 \ = \ (.72, .24, .54, .36)^T$.

Review of Relevant Mathematics

Floating Point Arithmetic

- Recall that a floating point number system, Z, can be characterized by four parameters, (β, s, m, M), and each element of Z is defined by:

$$z = .d_1 d_2 \cdots d_s \times \beta^e,$$

  where $d_1 \neq 0$, $0 \leq d_i \leq (\beta - 1)$, and $m \leq e \leq M$.

- The floating point representation mapping, $fl(x)$, is a mapping from the Reals to Z that satisfies:

$$fl(x) = x(1 + \epsilon), \text{ with } |\epsilon| \leq \mu.$$

  where μ is the 'unit roundoff' and is defined to be $1/2\ \beta^{1-s}$.

FP Arithmetic (cont)

- For any standard elementary arithmetic operation (+, -, $\times$ and /), we have the corresponding F.P. approximation (denoted by $\oplus, \ominus, \otimes$ and $\oslash$) which satisfies, for any $a, b \in Z$,

$$a \odot b = fl(a \cdot b) = (a \cdot b)(1 + \epsilon),$$

  where $|\epsilon| \leq \mu$ and $\cdot$ is any elementary operation.

- For any real-valued function, $F(a_1, a_2, \cdots a_n)$, the most we can expect is that the floating point implementation $\bar{F}$, will return (when invoked) the value $\bar{y}$ satisfying:

$$\begin{aligned} \bar{y} &= \bar{F}(fl(a_1), fl(a_2), \cdots fl(a_n)), \\ &= \bar{F}(a_1(1+\epsilon_1), a_2(1+\epsilon_2), \cdots a_n(1+\epsilon_n)), \\ &= fl(F(a_1(1+\epsilon_1), a_2(1+\epsilon_2), \cdots a_n(1+\epsilon_n)). \end{aligned}$$

FP Function Evaluation

In this case,

$$\begin{aligned}\bar{y} - y &= [fl(F(a_1(1+\epsilon_1), a_2(1+\epsilon_2), \cdots a_n(1+\epsilon_n)) \\ &\quad -F(a_1(1+\epsilon_1), a_2(1+\epsilon_2), \cdots a_n(1+\epsilon_n))] \\ &\quad +[F(a_1(1+\epsilon_1), a_2(1+\epsilon_2), \cdots a_n(1+\epsilon_n)) \\ &\quad -F(a_1, a_2 \cdots a_n)]. \\ &\equiv A + B,\end{aligned}$$

where $\frac{|A|}{|y|} < \mu$ and $|B|$ can be bounded using the MVT for multivariate functions.

FP Error Bound

If $y = F(a_1, a_2, \cdots a_n)$ is the desired result (defined by exact arithmetic over the Reals), the computed value, $\bar{y}$, will at best satisfy:

$$\begin{aligned} \frac{|\bar{y} - y|}{|y|} &\leq \mu + \frac{\|(\frac{\partial F}{\partial \underline{x}})^T (a_1\epsilon_1, a_2\epsilon_2 \cdots a_n\epsilon_n)^T\|}{\|F\|}, \\ &\leq \mu + \frac{\|\frac{\partial F}{\partial \underline{x}}\| \ \|a\|\mu}{\|F\|}, \end{aligned}$$

where

$$(\frac{\partial F}{\partial \underline{x}})^T = [\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \cdots \frac{\partial F}{\partial x_n}],$$

evaluated at $\underline{x} = a = (a_1, a_2, \cdots a_n)$. That is, the relative errors can be large (independent of the approximation used) whenever

$$\frac{\|\frac{\partial F}{\partial \underline{x}}\|\|a\|}{\|F\|} \quad \text{is large.}$$

Linear Algebra – A Review

We will first review results from Linear Algebra. In doing so we introduce our notation and recall the standard definitions and results that you should be familiar with from previous courses.

The $n \times m$ matrix, A is represented by,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \text{ where } a_{ij} \in \Re.$$

$\Re^{n \times m}$ denotes the set of all such matrices.

Basic Definitions

- The elements $\{a_{ii} : i = 1, 2 \cdots \min(n, m)\}$ form the diagonal of A.
- $\{a_{i\ i+1} : i = 1, 2 \cdots \min(n, m - 1)\}$ is the superdiagonal of A.
- $\{a_{i\ i-1} : i = 2, 3 \cdots \min(n, m + 1)\}$ is the subdiagonal of A.
- A is Lower Triangular if $a_{ij} = 0$ for $i < j$. A is Upper Triangular if $a_{ij} = 0$ for $i > j$. Furthermore we will say that A is 'strictly' Lower (Upper) Triangular if it is Lower (Upper) Triangular and the diagonal of A is $= 0$.

Matrix Multiplication

If A and B are both $n \times n$ (square) matrices then the product is,

$$C = A\ B, \text{ where } C \equiv [c_{ij}]$$

and c_{ij} is the inner product of row i of A with column j of B. That is,

$$c_{ij} \equiv \sum_{r=1}^{n} a_{ir} b_{rj}.$$

For nonsquare matrices ($m \neq n$) the definition of matrix multiplication holds provided the inner product is well defined.

Matrix Multiplication (cont)

- Matrix Multiplication is Associative:

$$A(BC) = (AB)C.$$

- Matrix Multiplication is not Commutative:

$$AB \text{ may not } = BA.$$

- The cancellation law does not hold. That is

$$AB = AC \text{ and } A \neq \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \text{does not imply } B = C.$$

Example

Consider,

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

In this case $B \neq C$ but $AB = AC$.

The Identity Matrix

The unit element for matrix multiplication is the identity matrix, denoted by I_n or $diag(1, 1 \cdots 1)$. It is the $n \times n$ square matrix,

$$I_n \equiv \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

For any $A \in \Re^{n \times m}$ we have

$$I_n A = A I_m = A.$$

Note that we will often write I for I_n if the dimension is obvious from the context of the expression.

The Transpose of a Matrix

- For $A = (a_{ij}) \in \Re^{n \times m}$, $A^T \in \Re^{m \times n}$ and is defined by,

$$A^T \equiv (\alpha_{ij}), i = 1, 2 \cdots m, \; j = 1, 2 \cdots n,$$

  where

$$\alpha_{ij} = a_{ji}.$$

  Note that A^T is called the transpose of A and can be considered the 'reflection' of A about the diagonal.

- For vectors $\underline{x} \in \Re^{n \times 1}$ we have,

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, x^T = [x_1, x_2 \cdots x_n], x^T \in \Re^{1 \times n}.$$

Properties of the Transpose

- The matrix A is symmetric iff $A = A^T$.
- Properties of matrix products: For matrices A and B with the dimensions such that the products and sums are well defined we have,
  - $(A^T)^T = A$
  - $(A + B)^T = A^T + B^T$
  - For $\lambda \in \Re$, $(\lambda A)^T = \lambda A^T$
  - $(A\ B)^T = B^T A^T$
  - $A^T A$ and $A\ A^T$ are symmetric

Linear Equations – A Review

- From mathematics we know that the problem,

$$Ax = b,$$

where $A \in \Re^{n\times n}$ and $b,\ x \in \Re^{n\times 1}$ has a solution iff b is linearly dependent on the columns of A. That is, if

$$A = \left[\begin{pmatrix} \\ \underline{a_1} \\ \\ \end{pmatrix} \begin{pmatrix} \\ \underline{a_2} \\ \\ \end{pmatrix} \cdots \begin{pmatrix} \\ \underline{a_n} \\ \\ \end{pmatrix} \right],$$

then $b = \sum_{r=1}^{n} c_r \underline{a_r}$, for some $c_1, c_2 \cdots c_n$.

- The solution is unique $\Leftrightarrow \det A \neq 0 \Leftrightarrow A$ is nonsingular $\Leftrightarrow \exists B \in \Re^{n\times n} \ni BA = AB = I_n$. Such a B is the <u>inverse</u> of A and is denoted A^{-1}.

Mathematical Preliminaties

- The matrix $Q \in \Re^{n \times n}$ is <u>orthogonal</u> if $Q^{-1} = Q^T$. (Note if Q is both symmetric and orthogonal then $Q^2 = Q\,Q^{-1} = I_n$.)
- Properties of inverses:
  - $(A^{-1})^{-1} = A$.
  - $(\lambda A)^{-1} = \frac{1}{\lambda} A^{-1}$ for $\lambda \in \Re$.
  - $(AB)^{-1} = B^{-1} A^{-1}$ for nonsingular A, B.
- Formally we can 'solve'

$$Ax = b$$

by multiplying by A^{-1} to obtain,

$$\begin{aligned} A^{-1}(Ax) &= A^{-1}b \\ (A^{-1}A)x &= A^{-1}b \\ x &= A^{-1}b. \end{aligned}$$

This is useful from a theoretical (but not computational) viewpoint.

Solving $Ax = b$

An alternative technique is to 'solve' the equation by first factoring (decomposing) A,

$$A = S\,T,$$

where S and T have special structure such that 'solving' the linear systems $Sx = b,$ and $Tx = b$ are both 'easy'. With this decomposition and $z = Tx$ we have,

$$\begin{aligned} Ax &= b \\ S\,Tx &= b \\ Sz &= b. \end{aligned}$$

Therefore to determine x we first solve the 'easy' problem $Sz = b$, and then solve a second 'easy' problem $Tx = z$. The special cases we will consider, are when S or T are triangular (forward or Back substitution is used) or orthogonal.

Calculus – A Review

Notation:

- $[a, b]$ is the closed interval, ($x \in R$, such that $a \leq x \leq b$).
- (a, b) is the open interval, ($x \in R$, such that $a < x < b$).
- $f^n(x) = \frac{d^n}{dx^n} f(x)$.
- $f \in C^n[a, b] \;\Rightarrow\; f$ is n times differentiable on $[a, b]$ and $f^n(x)$ is continuous on (a, b).
- $g_x(x, y) \equiv \frac{\partial}{\partial x} g(x, y)$, $g_y(x, y) \equiv \frac{\partial}{\partial y} g(x, y)$, $g_{xy}(x, y) \equiv \frac{\partial^2}{\partial x \partial y} g(x, y)$ etc.
- $g(h) = O(h^n)$ as
  $h \rightarrow 0 \Leftrightarrow \exists h_0 > 0 \; and \; K > 0 \ni |g(h)| < K h^n \; \forall \; 0 < h < h_0$.

Theorems From Calculus

- **Intermediate Value Theorem**
  Let $f(x)$ be continuous on $[a,b]$. If $f(x_1) < \alpha < f(x_2)$ for some α and $x_1,\ x_2 \in [a,b]$, then $\alpha = f(\eta)$ for some $\eta \in [a,b]$.

- **Max-Min Theorem**
  Let $f(x)$ be continuous on $[a,b]$. Then $f(x)$ assumes its maximum and minimum values on $[a,b]$. (That is, $\exists \underline{x}\ and\ \bar{x}\ \in [a,b] \ni \forall x \in [a,b]$, we have $f(\underline{x}) \leq f(x) \leq f(\bar{x})$.)

- **Mean Value Theorem for Integrals**
  Let $g(x)$ be a non-negative (or non-positive) integrable function on $[a,b]$. If $f(x)$ is continuous on $[a,b]$ then

$$\int_a^b f(x)g(x)dx = f(\eta)\int_a^b g(x)dx,$$

  for some $\eta \in [a,b]$.

Theorems (cont)

- **Mean Value Theorem for Sums**
  Let $f(x) \in C^1[a,b]$, let $x_1, x_2, \cdots, x_n$ be points in $[a,b]$ and let $w_1, w_2, \cdots, w_n$ be real numbers of one sign, then

$$\sum_{i=1}^{n} w_i f(x_i) = f(\eta) \sum_{i=1}^{n} w_i,$$

  for some $\eta \in [a,b]$.

- **Rolle's Theorem**
  Let $f(x) \in C^1[a,b]$. If $f(a) = f(b) = 0$ then $f'(\eta) = 0$ for some $\eta \in (a,b)$.

Theorems (cont)

- **Mean Value Theorem for Derivatives**

  If $f(x) \in C^1[a,b]$ then

$$\frac{f(b) - f(a)}{b - a} = f'(\eta),$$

  for some $\eta \in (a,b)$.

- **Fundamental Theorem of Calculus**

  If $f(x) \in C^1[a,b]$ then $\forall x \in [a,b]$ and any $c \in [a,b]$ we have

$$f(x) = f(c) + \int_c^x f'(s)ds.$$

Theorems (cont)

- **Taylor's Theorem (with remainder)**
  If $f(x) \in C^{n+1}[a,b]$ and $c \in [a,b]$, then for $x \in [a,b]$,

$$\begin{aligned} f(x) &= f(c) + f'(c)(x-c) + \cdots + f^n(c)\frac{(x-c)^n}{n!} \\ &\quad + R_{n+1}(x), \end{aligned}$$

  where $R_{n+1}(x) = \frac{1}{n!}\int_c^x (x-u)^n f^{n+1}(u)du$.

Note that Taylor's Theorem is particularly relevant to this course. We can observe that, since $(x-u)^n$ is of constant sign for $u \in [c,x]$,

$$R_{n+1}(x) = \frac{1}{n!}\int_c^x (x-u)^n f^{n+1}(u)du = f^{n+1}(\eta)\frac{(x-c)^{n+1}}{(n+1)!},$$

for some $\eta \in [c,x]$.

Taylors Theorem (cont)

We can also observe the first few terms of the Taylor Series provides an accurate approximation to $f(c+h)$ for small h since we have for $h = x - c$,

$$
\begin{aligned}
f(c+h) &= f(c) + hf'(c) + \cdots \frac{h^n}{n!} f^n(c) \\
&+ \frac{h^{n+1}}{(n+1)!} f^{n+1}(\eta).
\end{aligned}
$$

where the error term, $E(h)$ is $O(h^{n+1})$.