Lecture notes for Math*2130

Marcus R. Garvie,
Department of Mathematics & Statistics
University of Guelph, ON, Canada

Winter 2012

Contents

Chapter 0

Numerical Analysis: more than just an academic exercise!

- The construction and subsequent analysis of numerical algorithms is a hugely important task in all areas of physical, social, and biological science.
- During the past half-century, the increase in computing power has led to more realistic models that require accurate and efficient numerical solutions.
- Incorrect numerical analysis has led to real-world numerical catastrophes (see the screen-shot on the next page).

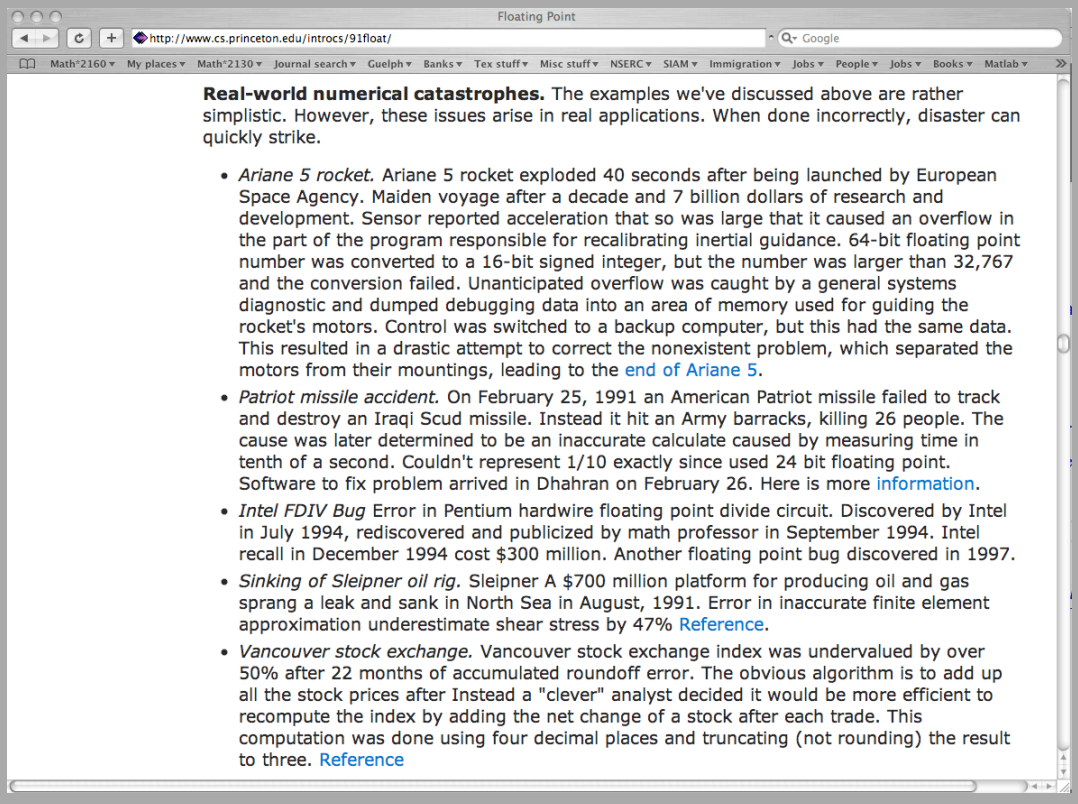

Figure 1: A screen capture

0.1 The Issues

(1) **why do we need numerical methods?**: e.g., approx of $\sqrt{2}$
(e.g., via the Bisection Method)

(2) **nonlinear versus linear**

(3) **accuracy**: e.g. $\frac{22}{7} \approx \pi$

(4) **precision**: e.g., 16 decimal digit arithmetic (16 significant figures)

(5) **rounding/truncation error**: (due to finite precision; affects accuracy)

(6) **practical numerical methods**: e.g., using $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots$ is NOT practical
(e.g., need 10,000 terms to get 4 d.p. correct)

(7) **ill-conditioned problems**: "small errors in the data/measurements cause large errors in the answer" (leads to poor accuracy)

(8) **error bounds**: e.g., |error| $\leq \mathbf{1.637}$

(9) **uniform error bounds**: e.g. |error| $\leq \mathbf{10.637}$ for all $x \in [\mathbf{0}, \mathbf{1}]$

(10) **tools**: inequalities / Taylor series/ Calculus / graphical techniques

Chapter 1

Basic tools

1.1 Review of Inequalities

In numerical analysis we are often bounding errors, which requires the use of inequalities. Thus we briefly recall the rules for using inequalities.

Rules of inequalities:

(i) **Multiplication by a positive constant**:
If $x > y$, then $ax > ay$ if a is a positive number.

(ii) **Multiplication by a negative constant**:
If $x > y$, then $ax < ay$ if a is a negative number.

(iii) **Addition of inequalities**:
If $x > y$, and $u > v$, then $x + u > y + v$.

(iv) **Subtraction of inequalities**:
If $x > y$, and $u > v$, then $(x - u) \not> (y - v)$.

(v) **Multiplication of inequalities**:
If x, y, u, v are positive, then $x > y$, $u > v$ implies $xu > yv$. The result is not necessarily valid when some of the numbers are negative.

(vi) **Division of inequalities**:
If $x > y$, and $u > v$, these do *not* imply $x/u > y/v$.

Monotonic functions

Inequalities involving functions often requires the consideration of when their graphs are increasing or decreasing.

Definition 1 (monotonic increasing)
A function $f(x)$ is said to be *monotonic increasing* in an interval (a, b), if, for all x_1, x_2 in (a, b)

$$x_1 < x_2 \implies f(x_1) \leq f(x_2).$$

Conversely, $f(x)$ is said to be *monotonic decreasing* in (a, b) if, for all $x_1, x_2 \in (a, b)$

$$x_1 < x_2 \implies f(x_1) \geq f(x_2).$$

If however, $f(x_1) < f(x_2)$, or $f(x_1) > f(x_2)$ for $x_1 < x_2$ the function is said to be *strictly monotonic increasing* or *strictly monotonic decreasing*, respectively.

Example 1
Consider the monotonicity of the function $f(x) = -x^2 + 4x - 3$.

Observe that
$f(x) = -(x-1)(x-3)$. So

- f is strictly monotonic increasing on $(-\infty, 2)$.
- f is strictly monotonic decreasing on $(2, \infty)$.

Moral: sometimes we have to split a problem up into separate monotonically increasing/decreasing pieces (requires Calculus/curve sketching techniques).

1.2 Taylor's Theorem

Motivation (polynomial approx. of degree 1)

Consider the graphical interpretation of approximating the derivative of the graph of a function $f(x)$ at x_0.

Figure 1.1: Slope of secant line.

We can see that:

$$f'(x_0) \approx \frac{f(x) - f(x_0)}{x - x_0} \; = \; \text{slope of the secant line AB},$$

or

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0),$$

or

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \text{error}.$$

The following theorem generalizes this formula and gives a specific form for the error function:

Theorem 1 (Taylor's Theorem with remainder)
Suppose the function $f(x)$ *has* $n+1$ *continuous derivatives on* $[a,b]$*, and* $x_1, x_2 \in [a,b]$*, then there exists a number* ξ_x *between* x_0 *and* x *s.t.*

$$\begin{aligned} f(x) &= f(x_0) + (x-x_0)f'(x_0) + \frac{(x-x_0)^2}{2!}f''(x_0) + \ldots \\ &\quad + \frac{(x-x_0)^n}{n!}f^{(n)}(x_0) + R_n(x), \\ &= P_n(x) + R_n(x), \\ \text{where} \quad P_n(x) &= \sum_{k=0}^{n} \frac{(x-x_0)^k}{k!}f^{(k)}(x_0), \\ \text{and} \quad R_n(x) &= \frac{(x-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(\xi_x). \end{aligned}$$

Note that the remainder $R_n(x)$ *is in the* **Lagrange** *form.*

Notes:

- An alternate form for the remainder is the **Cauchy** form:

$$R_n(x) = \frac{1}{n!}\int_{x_0}^{x} (x-t)^n f^{(n+1)}(t)\,dt.$$

- The remainder term $R_n(x)$ is an error term depending on x.
- ξ_x is an unknown number between x_0 and x.
- We say that P_n is the nth order Taylor polynomial (or series) about (or round) x_0.
- As n increases the 'fit' between the graphs of $f(x)$ and $P_n(x)$ *near* x_0 improves.
- When $x_0 = 0$ the Taylor Series is called a *Maclaurin Series*. Many examples are given at `http://mathworld.wolfram.com/MaclaurinSeries.html`.
- Taylor series converge for x within a certain interval (called the *interval of convergence*). Some functions converge for all x (e.g., e^x, $\sin x$, $\cos x$), while others have more restricted intervals of convergence (e.g., $\ln(1+x)$, valid for $x \in (-1,1)$).

Example 2

Construct the Taylor series for e^x about $x_0 = 0$ with a remainder term:

$$\begin{aligned} &f(x) = e^x \implies f(0) = 1, \\ &f'(x) = f''(x) = \cdots = f^{(n)}(x) = e^x, \\ &\text{so} \\ &f'(0) = f''(0) = \cdots = f^{(n)}(0) = 1. \end{aligned}$$

Thus

$$\begin{aligned} e^x &= P_n(x) + R_n(x), \\ &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \frac{x^{n+1}}{(n+1)!}e^{\xi_x}, \end{aligned}$$

where ξ_x is an unknown number between x and 0.

Example 3 (Uniformly bounding the error)

For $f(x) = e^x$ with $x_0 = 0$ find n s.t. the error between $f(x)$ and $P_n(x)$ is less than 10^{-6}, for all $x \in [-1, 1]$.

$$\text{Recall} \qquad e^x = P_n(x) + R_n(x),$$

so we want n s.t.

$$|e^x - P_n(x)| = |R_n(x)| = \left|\frac{x^{n+1}}{(n+1)!}e^{\xi_x}\right| < 10^{-6},$$

for all $x \in [-1, 1]$ and for ξ_x between 0 and x.

Now we will find an upper bound for $R(x)$ for all $x \in [-1, 1]$:

$$\left|\frac{x^{n+1}}{(n+1)!}e^{\xi_x}\right| = \frac{|x|^{n+1} \cdot e^{\xi_x}}{(n+1)!} \leq \frac{1 \cdot e^1}{(n+1)!} = \frac{e}{(n+1)!}$$

noting that:

- $e^{\xi_x} < e^1$ as e^z is increasing $(z_1 < z_2 \Rightarrow e^{z_1} < e^{z_2})$
- $x \in [-1, 1]$ or equivalently, $|x| \leq 1 \Rightarrow |x|^m \leq 1$, $m \geq 1$

Figure 1.2: The graph of $|x|$.

So we need to find n s.t.

$$\frac{e}{(n+1)!} < 10^{-6}$$

Consider the calculations:

n	1	2	...	8	9
$\frac{e}{(n+1)!}$	$\mathbf{1.36}$	$\mathbf{0.453}$	...	$\mathbf{7 \times 10^{-6}}$	$\mathbf{7.49 \times 10^{-7}}$

Thus $n = 9$, i.e. we need a polynomial of degree 9 for the absolute error between e^x and $P_n(x)$ to be less than 10^{-6} for *all* $x \in [-1, 1]$.

Alternative Form of the Taylor Series

In the previous formula (Theorem 1) let $h = x - x_0$ (so $x = x_0 + h$) leading to

$$\boxed{\begin{aligned} f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \ldots \\ + \frac{h^n}{n!}f^{(n)}(x_0) + \frac{h^{(n+1)}}{(n+1)!}f^{(n+1)}(\xi), \end{aligned}}$$

where ξ between x_0 and $x_0 + h$.

(Application Example (Taylor Series))

Intermediate, Mean and Extreme Value Theorems

We review some more fundamental results in Calculus

Theorem 2 (Mean Value Theorem (M.V.T.) for the differential Calculus)
Let $f(x)$ be continuous on $[a, b]$ and differentiable on (a, b), then $\exists\ \xi \in (a, b)$ s.t.

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

Figure 1.3: Illustration of Theorem 2.

Special Case: (Rolle's Theorem)

If $f(a) = f(b)$, then $\exists\ \xi \in (a, b)$ s.t. $f'(\xi) = 0$

Figure 1.4: Illustration of Rolle's Theorem.

Usefullness of the M.V.T.

Replace a with x_1, and b with x_2 and we get:

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

or

$$f(x_2) - f(x_1) = f'(\xi) \cdot (x_2 - x_1)$$

So we can bound $f(x_2) - f(x_1)$ via

$$\begin{aligned} |f(x_2) - f(x_1)| &= |f'(\xi)| \cdot |x_2 - x_1| \\ &\leq M \cdot |x_2 - x_1|, \end{aligned}$$

where M is an upper bound on $f'(\xi)$ (e.g., if $f'(x) = \mathbf{cos}(x)$ or $\mathbf{sin}(x)$, $M = \mathbf{1}$).

(see homework/exam questions)

Theorem 3 (Intermediate Value Theorem)
Let $f \in C([a, b])$ (l.e., f continuous on $[a, b]$), then given any η between $f(a)$ and $f(b)$, $\exists\ c \in [a, b]$ s.t. $f(c) = \eta$.

Figure 1.5: Illustration of the Theorem 3.

Theorem 4 (Extreme Value Theorem)
Let $f \in C([a,b])$, then f attains its maximum and minimum ('extreme') values on $[a,b]$. Moreover, extreme values are either endpoints, or critical points.

Figure 1.6: Illustration of the Theorem 4.

How can a function *not* attain it's max & min values?

Figure 1.7: A discontinuous function that doesn't have a min value.

Theorem 5 (Mean Value Theorem for the Integral Calculus)

- *First (simple) form:*

  Let $f \in C([a,b])$,
  then $\exists\ \xi \in (a,b)$ *s.t.* $\displaystyle\int_a^b f(x)\,dx = f(\xi)\cdot(b-a).$

- *Second form:*

  Let $f \in C([a,b])$ *and* $g(x)$ *does not change sign on* $[a,b]$,
  then $\exists\ \xi \in (a,b)$ *s.t.* $\displaystyle\int_a^b f(x)\cdot g(x)dx = f(\xi)\cdot\int_a^b g(x)dx.$

(with $g(x)=1$, this second form reduces to the first form).

Figure 1.8: Illustration of Theorem 5 (1st form).

$$\text{i.e. } \int_a^b f(x)\,dx = \text{ shaded area} = f(\xi)(b-a).$$

Theorem 6 (l'Hospital's rule)
Suppose we have two functions $f(x)$ and $g(x)$ which are zero when $x = a$. Then

$$\lim_{x\to a}\frac{f(x)}{g(x)} = \lim_{x\to a}\frac{f'(x)}{g'(x)},$$

provided the limit on the right exists.

- The rule is needed as $f(a)/g(a)$ is an undefined quantity $0/0$
- The rule also works when $\lim_{x\to a} f(x) = \infty$ and $\lim_{x\to a} g(x) = \infty$ (yielding the undefined quantity ∞/∞)
- We can replace $x \to a$ with $x \to \pm\infty$
- e.g. $\lim_{x\to 1}(\log_e x)/(x^2-1) = \lim_{x\to 1}(1/x)/(2x) = \lim_{x\to 1}\frac{1}{2x^2} = 1/2$

Proof of Theorem 6 (for the case when $f(a) = g(a) = 0$):
Consider the ratio of $f(x)$ and $g(x)$ and let both functions be expanded about the point $x = a$ using Taylor's theorem (without remainder). Then

$$\frac{f(x)}{g(x)} = \frac{f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \ldots}{g(a) + (x-a)g'(a) + \frac{(x-a)^2}{2!}g''(a) + \ldots} \quad (x \neq a).$$

Now by assumption $f(a) = g(a) = 0$. Hence we get after cancelling $(x-a)$ from top and bottom:

$$\frac{f(x)}{g(x)} = \frac{f'(a) + \frac{(x-a)}{2!}f''(a) + \ldots \text{higher powers of } (x-a)}{g'(a) + \frac{(x-a)}{2!}g''(a) + \ldots \text{higher powers of } (x-a)},$$

and consequently

$$\lim_{x\to a}\frac{f(x)}{g(x)} = \frac{f'(a)}{g'(a)} = \lim_{x\to a}\frac{f'(x)}{g'(x)},$$

provided $g'(a)$ is non-zero. □

1.3 Error and Asymptotic Error

Error

Let A_h be a quantity depending on a parameter h, and A_h approximates a quantity A. Then,

$$\begin{aligned}&\text{error} = A - A_h\\ &\text{absolute error} = |A - A_h|\\ &\text{relative error} = \frac{|A - A_h|}{|A|}\end{aligned}$$

- The relative error should be used when A is very small or very large.
- Multiply the relative error by 100 to get $\%$ error.

Approximate equality

$A \approx B$ means A and B are approximately equal.

Example 4
By making a change of variable, re-write the rule

$$x_n = 1 + \frac{1}{n}, \qquad n = 1, 2, 3, \ldots \quad (n \in \mathbb{N}).$$

so that the terms in the sequence are indexed by a small parameter h:
Clearly $\lim_{n\to\infty} x_n = 1$, so for large n, $x_n \approx 1$.

Alternatively, setting $h = \frac{1}{n}$ yields the rule,

$$x_h = 1 + h,$$

with $\lim_{h\to 0} x_h = 1$, so for small h, $x_h \approx 1$.

Asymptotic Order

Definition 2 ("Big-O")
(i) Let $f(h)$, $g(h)$ be functions depending on a **small** parameter h, s.t. $f(h), g(h) \to 0$ as $h \to 0$. We write

$$f(h) = \mathcal{O}(g(h)) \quad \text{if } \lim_{h\to 0}\left|\frac{f(h)}{g(h)}\right| = C < \infty, \quad \text{for some number } C \geq 0. \tag{1.1}$$

(ii) Let $f(n)$, $g(n)$ be functions depending on a **large** parameter, e.g. $n \in \mathbb{N}$ s.t. $f(n)$, $g(n) \to \infty$ as $n \to \infty$. We write

$$f(n) = \mathcal{O}(g(n)) \quad \text{if } \lim_{n\to \infty}\left|\frac{f(n)}{g(n)}\right| = C < \infty, \quad \text{for some number } C \geq 0. \tag{1.2}$$

Notes:

- (1.1) expresses the fact that the rates at which f and g go to zero are proportional (if $C = 0$ then f tends to zero faster than g tends to zero).
- (1.2) expresses the fact that the rates at which f and g go to ∞ are proportional (if $C = 0$ then g tends to infinity faster than f tends to infinity).
- We say "f is on the order of g", or "f is of the order g".
- For the case $C = 0$ the "Little-o" definition is used in the literature (not required in this course).

Example 5
Show for h a small parameter that $4h + 2 = \mathcal{O}(1)$:

$$\lim_{h\to 0}\left|\frac{4h+2}{1}\right| = 2 < \infty.$$

Example 6
Show for h a small parameter that $h^2 + 2h^4 = \mathcal{O}(h^2)$:

$$\lim_{h\to 0}\left|\frac{h^2+2h^4}{h^2}\right| = \lim_{h\to 0}|1+2h^2| = 1 < \infty$$

Example 7
Show for $n \in \mathbb{N}$ that $3n^3 - n^2 - n = \mathcal{O}(n^3)$:

$$\lim_{n\to\infty}\left|\frac{3n^3-n^2-n}{n^3}\right| = \lim_{n\to\infty}\left|3 - 1/n - 1/n^2\right| = 3 < \infty.$$

Example 8
Show for x large that $\ln(x) = \mathcal{O}(x)$.

Need l'Hospital's rule (Theorem 6) as we get the 'indeterminate form' ∞/∞:

$$\lim_{x\to\infty}\left|\frac{\ln(x)}{x}\right| = \lim_{x\to\infty}\left|\frac{(1/x)}{1}\right| = 0$$

(clearly x goes to ∞ faster than $\ln(x)$ does).

Application to Taylor Series

Use of the 'Big-O' notation allows us to neglect higher order terms in Taylor Series with respect to a small parameter h, for example:

$$e^h = 1 + h + \frac{h^2}{2!} + \overbrace{\frac{h^3}{3!}e^{\xi}}^{R_3}, \qquad \xi \text{ between } 0 \text{ and } h.$$

Notice $\lim_{h\to 0}\left|\frac{R_3}{h^3}\right| = \lim_{h\to 0}\left|\frac{e^{\xi}}{3!}\right| = \frac{e^{\xi}}{3!} < \infty$, i.e. $R_3 = \mathcal{O}(h^3)$.

Alternatively, $e^h = 1 + h + \frac{h^2}{2!} + \left(\frac{h^3}{3!} + \frac{h^4}{4!} + \ldots\right)$,

and $\lim_{h\to 0}\left|\frac{\frac{h^3}{3!} + \frac{h^4}{4!} + \ldots}{h^3}\right| = \lim_{h\to 0}\left|\frac{1}{3!} + \frac{h}{4!} + \text{terms with higher powers of } h\right|$

$$= \frac{1}{3!} < \infty, \text{ so } e^h = 1 + h + h^2/2! + \mathcal{O}(h^3).$$

By the same reasoning:

$$\begin{aligned} e^h &= 1 + \mathcal{O}(h) \\ &= 1 + h + \mathcal{O}(h^2), \quad \text{etc.} \end{aligned}$$

Similarly, for small x:

$$\begin{aligned} \sin(x) &= \mathcal{O}(x) \\ &= x + \mathcal{O}(x^3) \\ &= x - \frac{1}{3!}x^3 + \mathcal{O}(x^5), \quad \text{etc.} \end{aligned}$$

$$\begin{aligned}\cos(x) &= 1 + \mathcal{O}(x^2)\\ &= 1 - \frac{1}{2!}x^2 + \mathcal{O}(x^4), \quad \text{etc.}\end{aligned}$$

Observation:
"the leading power of the 1st term we neglect in the Taylor Series expansion, say p, leads to a remainder that is $\mathcal{O}(h^p)$" (h assumed small)

Excercise:
Use Taylor Series to explain why $\lim_{x\to 0}\left|\frac{\sin(x)}{x}\right| = 1$ and hence that $\sin(x) = \mathcal{O}(x)$. Check your answer with l'Hospital's rule (Theorem 6).

1.4 Computer Arithmetic

In the **floating point number system** we have a fixed number of significant figures ('Finite Precision'), due to using computers with finite memory. A non-zero number [1] x is stored in a computer with the following 'normalized floating-point representation':

$$\begin{aligned}fl(x) = \pm\, .d_1d_2...d_t \times \beta^e, \quad d_1 \neq 0\\ 0 \leq d_i \leq \beta - 1, \quad L \leq e \leq U,\end{aligned}$$

where d_i and e are integers, and

$e =$ exponent,
$\beta =$ base,
$.d_1d_2...d_t =$ mantissa,
$t =$ precision ("t-digit decimal arithmetic")

The set of all floating point numbers is denoted $\mathbb{F}(\beta, t, L, U)$ (a discrete *finite* set).

[1] The number 0 requires a special representation.

Example 9
Represent the number $\mathbf{0.01320}$ in standard floating point form using 4-decimal digit arithmetic in base 10:

With $\beta = \mathbf{10}$, $t = \mathbf{4}$, then

$$\mathbf{0.01320} = +\mathbf{.1320} \times \mathbf{10}^{-\mathbf{1}}.$$

Example 10
Represent the number $-\mathbf{398}$ in standard floating point form using 3-decimal digit arithmetic in base 10:

With $\beta = \mathbf{10}$, $t = \mathbf{3}$, then

$$-\mathbf{398} = -\mathbf{.398} \times \mathbf{10}^{\mathbf{3}}.$$

Notes:

- Computers usually use the binary ($\beta = \mathbf{2}$), the octal ($\beta = \mathbf{8}$), and the hexadecimal ($\beta = \mathbf{16}$) number systems, rather than the decimal system ($\beta = \mathbf{10}$).
- L and U depend on the particular computer used, e.g., on an Intel personal computer with a Pentium chip $\beta = \mathbf{2}$, $t = \mathbf{52}$, $L = -\mathbf{16382}$, $U = \mathbf{16383}$.
- Calculations leading to answers outside the range of the floating point system produce either *overflow* (too big), or *underflow* (too small) errors.

$$\text{Smallest number } = -\mathbf{.100\ldots0} \times \mathbf{10}^{L}$$
$$\text{Largest number } = +\mathbf{.999\ldots9} \times \mathbf{10}^{U}$$

- Depending on the computer, a certain number of bits ($\mathbf{0}$'s and $\mathbf{1}$'s) are allocated to a number. For example, $t = \mathbf{23}$ bits for the mantissa, and $\mathbf{8}$ bits for the exponent, and a single bit for the sign (*single precision*). *Double precision* has $\mathbf{64}$ bits ($t = \mathbf{52}$ bits for the mantissa, $\mathbf{11}$ bits for the exponent, single bit for the sign).
- The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is the most widely-used standard for floating-point computation (tells us how to deal with $\mathbf{0/0}$, $\infty - \infty$ etc.).

Chopping and Rounding

Even before a calculation is done on a computer, the storage of numbers usually involves error, because $fl(x) \approx x$ (and so the absolute error is $|fl(x) - x|$).

Chopping in base 10:
If $x = \pm(.d_1 d_2 ... d_t d_{t+1} \dots) \times 10^e$, $d_1 \neq 0$, then the chopped representation is

$$fl(x) = \pm .d_1 d_2 \dots d_t \times 10^e$$ (we truncate the mantissa after t digits.)

Error in Chopping to n d.p. (number not in standard form):
If we chop a number to n decimal places, then

$$\boxed{|\text{error}| \leq 10^{-n}.}$$

Example 11 (Chopping)
Bound the error in chopping the number 6.658 to $n = 2$ decimal places (using 3-decimal digit arithmetic in base 10):

$$6.658 = 6.65 \ (2 \text{ d.p.}), \quad \text{so}$$
$$|\text{error}| = |6.658 - 6.65| = 0.008 < 0.01 = 10^{-2}.$$

Example 12 (Illustrates worst case, and hence explains the formula)
Bound the error in chopping the number $6.65\dot{9}$ to $n = 2$ decimal places (using 3-decimal digit arithmetic in base 10):

$$6.65\dot{9} = 6.65 \ (2 \text{ d.p.}), \quad \text{so}$$
$$|\text{error}| = 0.00\dot{9} = 0.01 = 10^{-2}.$$

Rounding ('up') in base 10:
With $x = \pm(.d_1 d_2 ... d_t d_{t+1} ...) \times 10^e$, $d_1 \neq 0$, the rounded representation is

$$fl(x) = \begin{cases} \pm .d_1 d_2 \dots d_t \times 10^e, & 0 \leq d_{t+1} < 5, \\ \pm[(.d_1 d_2 ... d_t) + \underbrace{(.00\dots01)}_{10^{-t}}] \times 10^e, & 5 \leq d_{t+1} < 10. \end{cases}$$

Error in Rounding to n d.p. (number not in standard form):
If we round a number to n decimal places, then

$$\boxed{|\text{error}| \leq 5 \times 10^{-(n+1)} = \frac{10}{2} \times 10^{-(n+1)} = \frac{1}{2} 10^{-n}.}$$

Example 13 (Rounding)
Bound the error in rounding the number 6.658 to $n = 2$ decimal places (using 3-decimal digit arithmetic in base 10):

$$6.658 = 6.66 \ (2 \text{ d.p.})$$
$$|\text{error}| = 0.002 < 0.005 = 5 \times 10^{-3}$$

Example 14 (Illustrates worst case, and hence explains the formula)
Bound the error in rounding the number 6.655 to $n = 2$ decimal places (using 3-decimal digit arithmetic in base 10):

$$6.655 = 6.66 \ (2 \text{ d.p.})$$
$$|\text{error}| = 5 \times 10^{-3}.$$

Large loss of Significance

The main situations that lead to a *large loss of significance* in a calculation using finite precision:

Subtraction/division of two nearly equal numbers.

Strategies for avoiding loss of significance:

If we can replace the expression to be calculated with an equivalent expression that avoids the situations above, then we can avoid large loss of significance, e.g.

(a) rearrange the expression

(b) replace the expression using Taylor Series

Example 15

```
octave-> 2-sqrt(2)^2
ans = -4.44089209850063e-16
```

Of course the exact answer is zero! What is going on here?

Example 16

```
octave-> a = 0.1;
octave-> b = 1e-17;
octave-> c = 2e-17;
octave-> (a+c)-(a+b)
ans = 0
```

And again we know that the exact answer is $\mathbf{1 \times 10^{-17}}$. Explain.

```
octave-3.0.1:28> (a-a)+(c-b)
ans =  1.00000000000000e-17
```

Example 17

```
octave-3.0.1:29> (a+c)/(a+b)
ans =  1
```

Example 18
Undertake the numerical analysis of the previous example:
With $a = 0.1$, $b = 1 \times 10^{-17}$, $c = 2 \times 10^{-17}$ observe:

$$\begin{aligned} 1 < \frac{a+c}{a+b} &= \frac{(a+b)+(c-b)}{a+b} \\ &= 1 + \frac{c-b}{a+b} \\ &< 1 + \frac{c-b}{a} \qquad \left(c > b, \quad \frac{c-b}{a+b} \approx \frac{c-b}{a}\right) \\ &= 1 + \frac{1 \times 10^{-17}}{1 \times 10^{-1}} \\ &= 1 + (1 \times 10^{-16}). \end{aligned}$$

Conclusion: OCTAVE evaluates $\dfrac{a+c}{a+b}$ to be exactly 1, but our analysis shows that the true answer is somewhere between 1 and $1 + (1 \times 10^{-16})$. Thus the error is less than or equal to 1×10^{-16}.

Example 19 (Graphical example)
Consider graphing $y = \frac{(1-\cos(x))}{x^2}$ where x is close to 0:

Analysis:

For small x , $\cos(x) \approx 1$, so $\frac{1-\cos(x)}{x^2} \approx \frac{0}{0}$, thus need L'Hospital's Rule:

$$\begin{aligned} \lim_{x \to 0} \frac{(1-\cos(x))}{x^2} &= \lim_{x \to 0} \frac{\sin(x)}{2x} \\ &= \lim_{x \to 0} \frac{\cos(x)}{2} = 1/2. \end{aligned}$$

Thus for x close to 0 we expect the graph to be approximately flat ($y = 1/2$). What happens with finite precision?

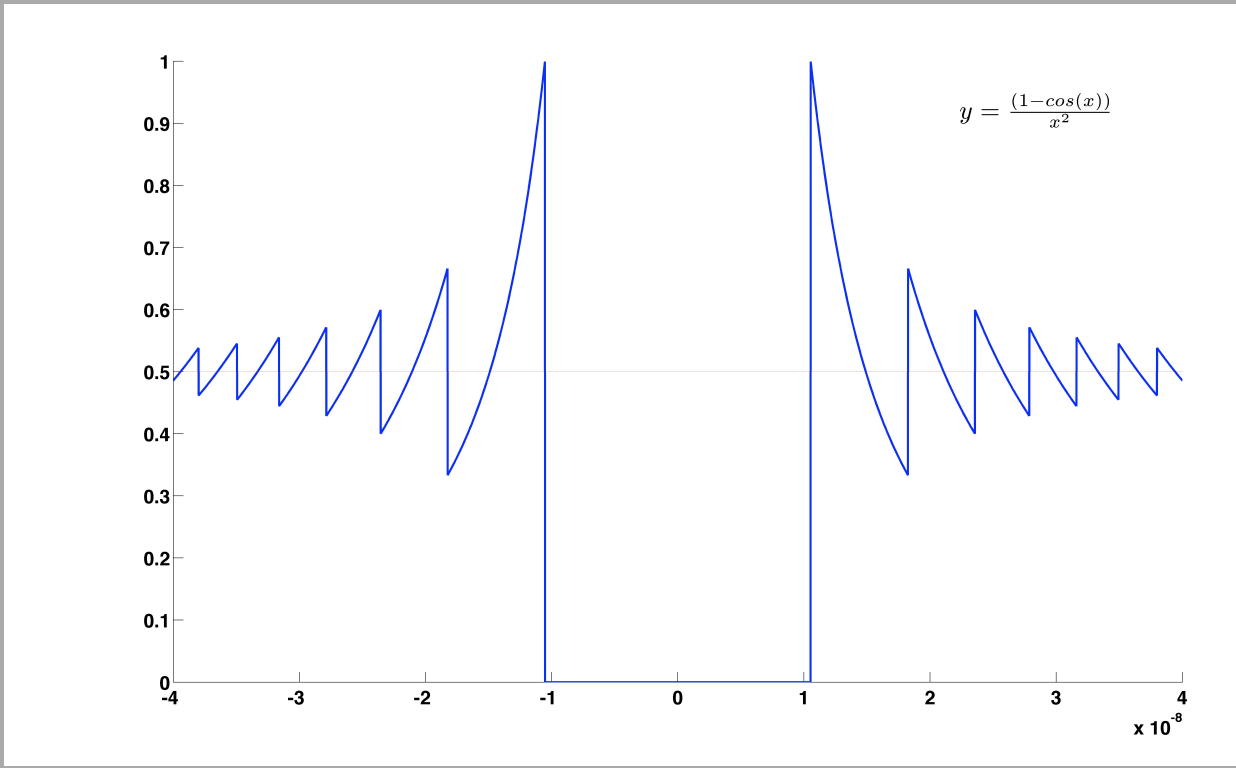

Figure 1.9: Illustration of large loss of significance:

Example 20

Consider 3-digit decimal arithmetic ($t = 3$) with rounding for calculation of

$$f(x) = \frac{1 - \cos(x)}{\sin(x)}, \text{ where } x \text{ is close to } 0.$$

Observe that $1 \approx \cos(x)$ and $\sin(x) \approx 0$ if x small, so $1 - \cos(x) \approx \sin(x)$ if $x \approx 0$.

Let $x = 0.03$, so after noting that $\cos(0.03) \approx 0.99955$ (angle in rads!), $fl(0.03) = 0.03$, $\sin(0.03) \approx 0.02999$, we have:

$$\begin{aligned} & fl\left(\frac{fl(1 - fl(\cos(0.03)))}{fl(\sin(0.03))}\right) \\ &= fl\left(\frac{fl(1 - 1.0)}{0.03}\right) \\ &= fl\left(\frac{0}{0.03}\right) \\ &= 0. \end{aligned}$$

The 'exact' answer is $\mathbf{0.015001\ldots}$. How can we avoid the loss of significance? We will rearrange $f(x)$:

$$\frac{1-\cos(x)}{\sin(x)} = \frac{1-\cos^2(x)}{\sin(x)(1+\cos(x))} = \frac{\sin^2(x)}{\sin(x)(1+\cos(x))} = \frac{\sin(x)}{1+\cos(x)}$$

So, again with $x = 0.03$:

$$\begin{aligned} &fl\left(\frac{fl(\sin(0.03))}{fl(1+fl(\cos(0.03)))}\right) \\ &= fl\left(\frac{0.03}{fl(1+1.0)}\right) \\ &= fl\left(\frac{0.03}{2}\right) \\ &= 0.015, \end{aligned}$$

which is a better answer than before.

Procedure for calculating the machine representation of an expression:

(i) round/chop numbers in the expression not belonging to $\mathbb{F}(\beta, t, L, U)$

(ii) do a single 'operation' exactly ($\pm$, $*$, $/$, $\sin(\cdot)$, etc)

(iii) round/chop exact answer

(iv) repeat (ii) etc.

Example 21
Write down the floating point representation of $\log\left(\frac{a+b}{c}\right)$:
Answer: $fl\left(\log\left(fl\left(\frac{fl(fl(a)+fl(b))}{fl(c)}\right)\right)\right)$.

Example 22
Re-arrange the following expression where x is close to 0:

$$g(x) = \frac{e^x - 1}{x}.$$

There will be a loss of significance for x close to 0, so we will use Taylor Series to avoid that loss:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad \text{, so}$$
$$\frac{e^x - 1}{x} = 1 + \frac{x}{2!} + \frac{x^2}{3!} + \dots \quad ------ \quad (*)$$

Exercise:
Consider $x = 0.01$, $t = 3$ with rounding. Show that the $g(x)$ is evaluated as 1, while the Taylor Series expansion is evaluated as 1.01 (exact answer is $1.0050167\cdots = 1.01$ (3 s.f.)).

Example 23 (An application - calculation of derivatives)

Recall that the derivative of a function f at a point x is defined as the limit of

$$\frac{f(x+h) - f(x)}{h},$$

as h tends to zero. In theory, the smaller the h, the better this difference quotient approximates the derivative. Initially this is true, but in practice as h gets below a certain size, finite precision on a computer leads to large loss of precision.

Consider calculating the derivative of $f(x) = \sin(x)$ at $x = 1$. Recall that

$$\left.\frac{d}{dx}\sin(x)\right|_{x=1} = \cos(1).$$

We tabulate the difference quotient for decreasing h.To make output easier to understand, incorrect digit have been replaced with periods:

h	$(\sin(1+h)-\sin(1))/h$
10^{-1}	$\ldots$
10^{-2}	$0.5\ldots$
10^{-3}	$0.5\ldots$
10^{-4}	$0.540\ldots$
10^{-5}	$0.540\ldots$
10^{-6}	$0.54030\ldots$
10^{-7}	$0.540302\ldots$
10^{-8}	$0.54030230\ldots$
10^{-9}	$0.5403023\ldots$
10^{-10}	$0.540302\ldots$
10^{-11}	$0.54030\ldots$
10^{-12}	0.5403
10^{-13}	$0.5\ldots$
10^{-14}	$0.54\ldots$
10^{-15}	$0.5\ldots$
10^{-16}	$0\ldots$

The 'exact' answer calculated to 15 s.f. in OCTAVE is $\cos(1) = 0.540302305868140$. When $h = 1\times 10^{-16}$ or smaller, the output is exactly zero because $\sin(1.0+h)$ equals $\sin(1.0)$ to machine precision. (In fact, $1+h$ equals 1 to machine precision. More on that below.) Once again we could use Taylor Series to obtain a better answer.

Definition 3 (Machine epsilon)
The *machine epsilon* of a computer is a number δ such that

(i) It is a positive floating point number, and

(ii) It is the smallest number δ ($\delta > 0$) for which

$$fl(1+\delta) > 1.$$

Notes

- In other words, *it is the smallest number we can add to* 1 *to get the next number in the floating point number system of a computer.*
- It is frequently called 'unit roundoff'.
- The definition in our book is not quite right.
- With the above definition, for any $\widehat{\delta} < \delta$ we have $fl(1+\hat{\delta}) = 1$, i.e. 1 and $1+\hat{\delta}$ represent the same number on the computer.

Example 24
What is machine epsilon when $\beta = 10$ and $t = 2$, with rounding?

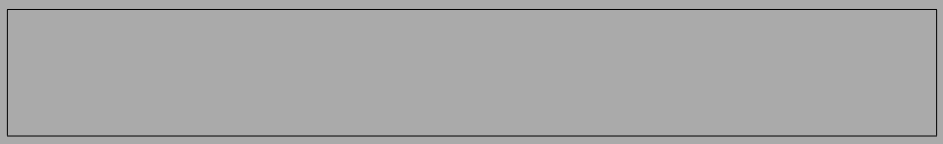

Figure 1.10: Floating point numbers on the real no. line.

Observe:

$$\begin{aligned} fl(1.0 + 0.04) &= fl(1.04) \\ &= 1.0 \\ fl(1.0 + 0.05) &= fl(1.05) \\ &= 1.1 \\ fl(1.0 + 0.06) &= fl(1.06) \\ &= 1.1 \end{aligned}$$

Clearly, $\delta = 0.05$.

Example 25
What is machine epsilon when $\beta = 10$ and $t = 2$, with chopping?

$$\begin{aligned} fl(1.0 + 0.09) = fl(1.09) = 1.0, \\ fl(1.0 + 0.0999) = fl(1.0999) = 1.0, \\ fl(1.0 + 0.1) = fl(1.1) = 1.1. \end{aligned}$$

Clearly in this case, $\delta = 0.1$ (the smallest number we add to 1.0 to get 1.1).
This generalizes to give the following results (see Q. 12 in your text - hint: $t = 24$):

Theorem 7 (Formulae for machine epsilon)
Consider the floating point number system $\mathbb{F}(\beta, t, L, U)$ with either rounding or chopping. The machine epsilon is given by:

$$\begin{aligned} &\textit{Rounding:} \quad \delta = \frac{1}{2}\beta^{1-t} \\ &\textit{Chopping:} \quad \delta = \beta^{1-t} \end{aligned}$$

Definition 4 (flops)
The number of floating point operations $(+, (-), \times, \div)$ performed by a computer to do a particular calculation is denoted by *flops*.

- The operation '$-$' is in brackets as we need only consider the plus operation, e.g., the calculation of $\mathbf{3-2=3+(-2)}$ requires a single flops (the negative sign is part of the floating point number $-\mathbf{2} = -\mathbf{.2} \times \mathbf{10^1}$).
- The definition is different from the computer science definition - e.g., the number of floating point operations performed by a computer per second.
- MATLAB used to have a command 'flops' that counted the number of floating point operations, but this command was removed to improve speed.

Example 26
How many flops are needed to evaluate $f(x) = \mathbf{3}x^2 + \mathbf{2}x - \mathbf{1}$?

$$\begin{aligned} f(x) &= \mathbf{3}x^2 + \mathbf{2}x - \mathbf{1} \\ &= \mathbf{3} * x * x + \mathbf{2} * x + (-\mathbf{1}) \end{aligned}$$

which requires:

$$\begin{aligned} * &: \mathbf{3} \text{ flops} \\ + &: \underline{\mathbf{2} \text{ flops}} \\ \text{total} &= \mathbf{5} \text{ flops} \end{aligned}$$

Why count flops?

- Counting the number of flops that an algorithm requires allows us to assess the efficiency of the algorithm relative to other algorithms. Suppose a particular numerical procedure takes $\mathcal{O}(N^q)$ flops, where N is some measure of the size of the problem (e.g., number of unknowns in a linear system of equations). Numerical analysts are interested in developing accurate numerical methods that are also efficient, i.e., methods where q is as small as possible.
- Counting the number of flops needed to perform a particular calculation also allows us to assess 'runtime' on a computer (in general, number of flops is proportional to runtime). And in industry, 'time is money'.
- Although the calculation of x^2 requires 1 flops, the calculation of 'intrinsic' functions usually requires many more flops. Can we make an educated guess as to how a calculator/computer calculates $\sin(x)$, or $x^{-\frac{1}{3}}$?

Chapter 2

A Survey of Simple Methods and Tools

We undertake a brief survey of simple numerical methods, which are also needed for more sophisticated methods.

2.1 Horner's Rule

Introduction

Example 27 (Nested Multiplication)
Re-write the polynomial $P_4(x) = -1+5x-3x^2+3x^3+2x^4$ so that it can be evaluated from the inside out:

$$\begin{aligned} P_4(x) &= -1 + x\left(5 - 3x + 3x^2 + 2x^3\right) \\ &= -1 + x\left(5 + x\left((-3) + 3x + 2x^2\right)\right) \\ &= -1 + x\left(5 + x\left((-3) + x\left(3 + 2x\right)\right)\right) \\ &= -1 + x * \left(5 + x * \left((-3) + x * \left(3 + 2 * x\right)\right)\right). \end{aligned}$$

Number of flops in evaluating the nested form of the polynomial:

$$\begin{aligned} &* : \quad 4 \text{ flops} \\ &+ : \quad 4 \text{ flops} \\ &\text{total} = 8 \text{ flops} \end{aligned}$$

The number of flops in evaluating the polynomial in standard form:

$$\begin{aligned} P_4(x) &= -1 + 5x - 3x^2 + 3x^3 + 2x^4 \\ &= (-1) + 5 * x + (-3) * x * x + 3 * x * x * x + 2 * x * x * x * x. \end{aligned}$$

Thus

$$\begin{aligned} &* : \ 1 + 2 + 3 + 4 = 10 \text{ flops} \\ &+ : \ 4 \text{ flops} \\ &\text{total} = 14 \text{ flops} \end{aligned}$$

Conclusion:
Evaluating the polynomial in nested form is much more efficient than evaluating in standard form.

General Case

Analyse the general case:
First note that the evaluation of each $a_n x^n$ term requires n multiplications. The general nth order polynomial in standard form

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n,$$

$$\text{requires: } * \ : \ 1 + 2 + 3 + ... + n = \sum_{k=1}^{n} k = \frac{n(n+1)}{2} \text{ flops}$$

$$+ \ : \ n \text{ flops}$$

$$\text{total flops} = n + \frac{n(n+1)}{2} = \frac{2n}{2} + \frac{n^2 + n}{2}$$

$$= \frac{n(n+3)}{2} = \mathcal{O}(n^2) \qquad (n \text{ large}).$$

While the general nested form, written as

$$P_n(x) = a_0 + x\left(a_1 + x\left(a_2 + ... + x\left(a_{n-1} + a_n x\right)\right)\right)$$
$$= a_0 + x * \left(a_1 + x * \left(a_2 + ... + x * \left(a_{n-1} + a_n * x\right)\right)\right),$$

$$\text{requires: } * \ : \ n \text{ flops}$$

$$+ \ : \ n \text{ flops}$$

$$\text{total flops} = 2n = \mathcal{O}(n) \qquad (n \text{ large}).$$

2.2 Difference Approximations to the Derivative

We develop finite difference formulae for approximating derivatives (numerical differentiation) of a function f, i.e., we wish to approximate $f'(x)$. These formulae can be used to approximate the solutions of differential equations.

Three simple finite difference formulae

Recall that by definition, $f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$, provided the limit exists. Thus, provided h small, we expect

$$\frac{f(x+h)-f(x)}{h} \approx f'(x) = \frac{df}{dx}.$$

The following difference quotients

$$\frac{f(x+h)-f(x)}{h} \quad \textit{Forward Difference Approximation } (\textbf{FDA}),$$

$$\frac{f(x)-f(x-h)}{h} \quad \textit{Backward Difference Approximation } (\textbf{BDA}),$$

$$\frac{f(x+h)-f(x-h)}{2h} \quad \textit{Centered Difference Approximation } (\textbf{CDA}),$$

are three well-known finite difference approximations of $f'(x)$. Note: the **CDA** is the average of the **FDA** and **BDA** forms.

Figure 2.1: Illustration of 3 finite difference approximations..

FDA:
BDA:
CDA:

Explicit formulae for the difference approximations using Taylor Series

How do we assess the accuracy of these approximations? Answer: use Taylor Series.

1. Forward Difference Form (**FDA**):

$$f(x+h) = f(x) + h \cdot f'(x) + \frac{h^2}{2!} f''(c_1)\,, \text{ for some } c_1 \text{ between } x \text{ and } x+h.$$

So,

$$\frac{f(x+h)-f(x)}{h} = f'(x) + \underbrace{\frac{h}{2} f''(c_1)}_{\mathcal{O}(h)},$$

or,

$$\boxed{f'(x) = \tfrac{f(x+h)-f(x)}{h} - \tfrac{h}{2} f''(c_1)}.$$

(the error in approximating $f'(x)$ by FDA is $\mathcal{O}(h)$)

2. Backward Difference Form (**BDA**):
Similarly, after writing $f(x-h) = f(x+(-h))$ we can show (Exercise) that

$$\boxed{f'(x) = \tfrac{f(x)-f(x-h)}{h} + \tfrac{h}{2} f''(c_2)},$$

where $\frac{h}{2} f''(c_2)$ is $\mathcal{O}(h)$ for some c_2 between x and $(x-h)$. Note: In both cases we must assume that f is twice continuously differentiable (i.e., f, f', f'' are continuous).

3. Centered Difference Form (**CDA**): $(h > 0)$

$$f(x+h) = f(x) + h \cdot f'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(c_1)\,,$$
$$f(x-h) = f(x) + (-h) \cdot f'(x) + \frac{(-h)^2}{2!} f''(x) + \frac{(-h)^3}{3!} f'''(c_2).$$
where $(x-h) < c_2 < x < c_1 < (x+h)$.

Thus (exercise),

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3!}\left(f'''(c_1) + f'''(c_2)\right).$$

So,

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \underbrace{\frac{h^2}{12}\left(f'''(c_1)+f'''(c_2)\right)}_{\mathcal{O}(h^2)},$$

or,

$$\boxed{f'(x) = \frac{f(x+h)-f(x-h)}{2h} - \frac{h^2}{12}\left(f'''(c_1)+f'''(c_2)\right)}$$

Notes:

- We assumed that f is thrice continuously differentiable (i.e., f, f', f'', f''' are continuous).
- For $h < 1 \implies h^2 < h$, i.e. for very small h our $\mathcal{O}(h^2)$ error term will tend to be smaller than the $\mathcal{O}(h)$ error term, regardless of the sizes of f'' and f'''. Thus, we expect the Centered Difference Approximation to be more accurate than the Forward, or Backward Difference Approximations.

Effect of Rounding Error

Approximating derivatives is an inherently unstable process (as we saw before), as the subtraction of two nearly equal numbers in the numerator of the formula when $h \approx 0$, leads to significant roundoff error[1] (see figure below): This diagram illustrates that for

Figure 2.2: Plot of the error in approximating f' against h..

sufficiently small h the numerical approximation of the derivative is dominated by rounding error, and that there will be an optimal h that minimizes the error.

[1] Error due to rounding.

Example 28

Consider the Centered Difference Approximation to e^x at $x = 1$.
The 'exact' error is

$$E(h) := \left| \frac{e^{1+h} - e^{1-h}}{2h} - e \right|$$

Tabulate $E(h)$ against decreasing h (using OCTAVE showing 3 s.f.):

$h = 10^{-n}$	$E(h)$	$h = 10^{-n}$	$E(h)$
$n = 1$	4.53×10^{-3}	$n = 9$	6.60×10^{-9}
3	4.53×10^{-7}	11	1.04×10^{-5}
5	5.86×10^{-11}	13	4.56×10^{-4}
7	5.86×10^{-11}	15	1.68×10^{-1}

Clearly for h smaller than about 10^{-7} the rounding errors begin to dominate, causing the errors to increase as h is decreased.

Determining the order of the errors

Suppose $A(h) \approx A$, where h is a small parameter,
e.g., $A = A(h) + E(h)$, where $E(h) = \mathcal{O}(h^p)$.
Then we say that the *method (or formula) is of order* p. We wish to work out the order numerically.
With the assumption that $E(h) = Ch^p$ (i.e., the leading term in a Taylor Series expansion of the error), observe

$$\boxed{\frac{E(h)}{E(h/2)} = \frac{Ch^p}{C(h/2)^p} = \frac{Ch^p}{Ch^p \cdot (1/2)^p} = 2^p.} \quad ------ \quad (*)$$

So, if the ratio of consecutive errors as we half h is approximately 2, we conclude that order $= 1$. If the ratio is approximately 4, we conclude that order $= 2$, etc. Important note: we neglected higher order terms in the expression for $E(h)$, thus this result is *approximately* true for h sufficiently small.

Alternatively, we can write $(*)$ as

$$E(h/2) = \frac{E(h)}{2^p}, \ p = 1, 2, 3 \ldots$$

E.g., if the formula is of order 1, then cutting h in half should cut the error approximately in half. And if the order is 2, then halving h quarters the error.

Example 29
With $f(x) = \ln(x)$, consider the formula for approximating $f'(1)$:

$$A(h) = \frac{8f(1+h) - 8f(1-h) - f(1+2h) + f(1-2h)}{12h} \approx f'(1).$$

What is the order of the formula?

First recall that $f'(x) = \frac{1}{x}$, so $f'(1) = 1 = A$.

The absolute error in the approximation of $f'(1)$ is

$$E(h) := |A(h) - f'(1)| = |A(h) - 1|$$

which leads to the tabulated results:

h	$E(h)$	$E(h)/E(h/2)$
2^{-2}	4.0024×10^{-3}	19.35
2^{-3}	2.0680×10^{-4}	16.70
2^{-4}	1.2380×10^{-5}	16.17
2^{-5}	7.6561×10^{-7}	16.04
2^{-6}	4.7725×10^{-8}	—

For h sufficiently small $E(h)/E(h/2) \approx 16 = 2^4 \implies$ order $= 4$.
(see Q8, Ex. 2.2)

Formulae for higher order derivatives

These are obtained in a similar manner to the first derivative case.

Example 30

Derive a formula for approximating $f''(x)$ using Taylor Series expansions.

Expand $f(x \pm h)$ upto the 4th power of h, and then add:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(c_1), \quad (c_1 \text{ between } x \text{ and } (x+h))$$

$$+$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(c_2), \quad (c_2 \text{ between } x \text{ and } (x-h))$$

− −

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{h^4}{24}\left(f^{(4)}(c_1) + f^{(4)}(c_2)\right),$$

Re-arranging yields:

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} - \underbrace{\frac{h^2}{24}\left(f^{(4)}(c_1) + f^{(4)}(c_2)\right)}_{\mathcal{O}(h^2)}.$$

2.3 Euler's Method

We use some of the techniques learned so far to approximate solutions of ordinary differential equations depending on time, called *Initial Value Problems* (IVPs).

IVPs

An IVP is a first-order ordinary differential equation, with an initial condition, solved on a specific (time) interval:

$$\begin{cases} y' = \frac{dy}{dt} = f(t, y) \qquad (y \equiv y(t)) \\ y(a) = y_0 \\ t \text{ in } [a, b] \end{cases}$$

Example 31 (The logistic d.e.)

$$\begin{cases} y' = ky(1-y),\ k > 0 \\ y(0) = y_0 \\ t \text{ in } [0,T] \end{cases}$$

Models the rate of change of a population with time, i.e. y', as proportional to $y(1-y)$. The method of 'separation of variables' yields the analytic solution

$$y(t) = 1 - \frac{1}{1 + \frac{y_0}{1-y_0}e^{kt}}, \quad y_0 \neq 1.$$

There are however many nonlinear IVPs that have no explicit solution formula, hence the need for a numerical method.

Finite Difference Approach to Euler's Method

We have $\frac{dy}{dt} = f(t,y),\ y(0) = y_0$ (taking $t_0 = 0$ for simplicity), $t \in [0,T]$. Define a *grid* of $N+1$ points that partitions $[0,T]$:

$$0 = t_0 < t_1 < t_2 < ... < t_n < ... < t_N = T,$$
where $t_n = nh$ (h is a small *time-step*), $n = 0, 1, 2, ..., N$.

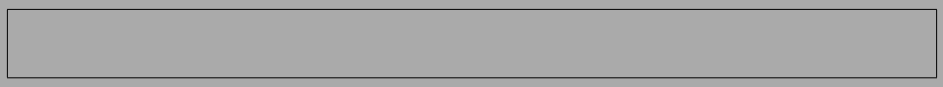

Figure 2.3: Illustration of grid points.

Aim: We construct a method that approximates y at each grid point, i.e. with

$$Y_n \approx y(t_n),\ n = 1, 2, ..., N$$

Notice that the IVP at $t_n = nh$ can be written as

$$\left.\frac{dy}{dt}\right|_{t=t_n} = f(t_n, y(t_n)),\ y(0) = y_0.$$

We approximate this IVP via a Forward Difference Approximation to dy/dt at time $t_n = nh$ to give:

$$\begin{cases} \frac{Y_{n+1}-Y_n}{h} = f(t_n, Y_n) \\ y(0) = Y_0 = y_0. \end{cases}$$

Here we have taken the initial approximation equal to y_0. Re-arranging yields:

$$\boxed{\begin{array}{l} Y_{n+1} = Y_n + hf(t_n, Y_n), \quad Y_0 = y_0 \\ \text{for } n = 0, 1, ..., N-1. \end{array}}$$

This difference equation is called *Euler's Method* for approximating ('solving') the IVP.

Notes:

- If we have $Y_n \approx y(t_n)$ (Y_n the approximation to y at time t_n), then

$$Y_{n+1} \approx y(t_{n+1}) = y((n+1)h) = y(nh + h) = y(t_n + h).$$

- Unlike in the text, I've deliberately used 'Y_n' instead of y_n.

Example 32 (Typical exam question)
Use Euler's Method to approximate the solution of

$$\begin{cases} y' = ty + t^3 = f(t, y) \\ y(0) = y_0 \\ t \in [0, 1] \end{cases}$$

with a step-size of $h = 0.2$ and $y_0 = 1$.
Euler's Method is:

$$\begin{aligned} Y_{n+1} &= Y_n + hf(t_n, Y_n), \\ Y_{n+1} &= Y_n + h(t_n Y_n + (t_n)^3), \\ Y_0 &= 1, \quad t_n = nh, \quad n = 0, 1, 2, \ldots \end{aligned}$$

Iterating the difference equation from Y_0:

$$\begin{aligned} \underline{n=0}: \quad Y_1 &= Y_0 + h(t_0 Y_0 + t_0^3) \\ &= 1 + (0.2)((0)(1) + (0)^3) = 1 \\ \underline{n=1}: \quad Y_2 &= Y_1 + h(t_1 Y_1 + (t_1)^3) \\ &= 1 + (0.2)((0.2)(1) + (0.2)^3) = 1.0416 \end{aligned}$$

$$\underline{n=2}: \quad \begin{aligned} Y_3 &= Y_2 + h\left(t_2\, Y_2 + (t_2)^3\right) \\ &= 1.0416 + (0.2)\left((0.4)(1.0416) + (0.4)^3\right) \\ &= 1.1377 \end{aligned}$$

Exercise: Check that $Y_4 = 1.3175$, $Y_5 = 1.6306$.

Arbitrary Uniform Grids

If $t \in [a, b]$, define a grid of $N+1$ points that partitions $[a, b]$:

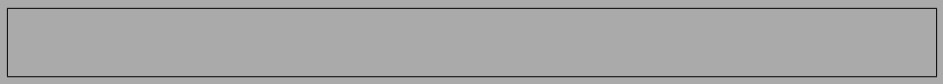

Figure 2.4: Illustration of grid points.

Generally, $\boxed{t_n = a + nh, \quad n = 0, 1, 2, \ldots, N}$

Note:
For a (uniform) partition, from the last grid point we have

$$\boxed{h = \frac{b-a}{N}.}$$

Two additional approaches to Euler's Method

Taylor Series Approach to Euler's Method:

Recall:
$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + ...$$

Expanding y about t_n:

$$\begin{aligned} y(t_{n+1}) &= y(t_n + h) \\ &= y(t_n) + hy'(t_n) + \frac{h^2}{2!}y''(t_n) + ... \end{aligned}$$

Truncating this expansion after the power of h term:

$$y(t_{n+1}) \approx y(t_n) + hy'(t_n),$$

where $y'(t_n) = \left.\frac{dy}{dt}\right|_{t=t_n} = f(t_n, y(t_n)),$

which is approximated by $\boldsymbol{Y}_{n+1} = \boldsymbol{Y}_n + hf(t_n, \boldsymbol{Y}_n)$ as before.

Geometrical Approach to Euler's Method:

The graph of $y(t)$ corresponding to $y' = \frac{dy}{dt} = f(t,y), \quad y(t_0) = y_0$, is illustrated below, together with the tangent line at $(t_0, y(t_0))$:

Figure 2.5: Geometrical illustration of Euler's Method.

Can we approximate Y_1 using the tangent line at (t_0, y_0)?

Observe:

$$\frac{Y_1 - Y_0}{h} = \text{slope at } (t_0, Y_0) = f(t_0, Y_0) \qquad (Y_0 = y_0).$$

$$\text{so } Y_1 = Y_0 + hf(t_0, Y_0). \qquad \text{Similarly,}$$

$$\frac{Y_2 - Y_1}{h} = \text{approx. of slope at } (t_1, y_1)$$

$$= f(t_1, Y_1) \qquad (\text{not } f(t_1, y_1) \text{ as we don't know } y_1!).$$

In general we have

$$\frac{Y_{n+1} - Y_n}{h} = f(t_n, Y_n), \quad \text{yielding the rule}$$

$$Y_{n+1} = Y_n + hf(t_n, Y_n), \quad n = 0, 1, 2, ..., N - 1.$$

Accuracy

The procedure repeated over many time-steps is illustrated below:

Figure 2.6: Several steps of Euler's.

Clearly, the error $|y(t_n) - Y_n|$ increases the more steps we take.

A bit more analysis

Reducing h reduces the max error on a fixed interval, but requires more steps (i.e., requires more computational work). Assuming the initial approximation is exact, the error after 1 time-step is

$$\mathcal{O}(h^2) \quad \text{('local error').}$$

This is easily justified from the Taylor Series expansion approach. The error after N time-steps is

$$\mathcal{O}(h) \quad \text{('global error').}$$

We can see this as follows. Recall that $N = (b-a)/h$. If the error in 1 step is $\propto h^2$, then the error in N steps is $\propto Nh^2 = (b-a)(1/h)h^2 \propto h$, i.e. $\mathcal{O}(h)$.

Conclusion

As the (global) error is $\mathcal{O}(h)$ we know that halving he step size will only cut the error in half. Euler's Method generally requires a very small step size to get good accuracy (with possible adverse roundoff and computing time drawbacks). Thus Euler's Method is rarely used in practice.

2.4 Linear Interpolation

Introduction

There are many situation where the data collected is incomplete, and we wish to estimate the missing data, e.g.

Figure 2.7: Population against year.

Definition 5 (Polynomial interpolation)
Polynomial Interpolation is the process of using a polynomial of a given degree to fit data. Given a set of points x_k, called *nodes*, we say a function P *interpolates* the underlying function f at the nodes if

$$P(x_k) = f(x_k), \quad \forall k.$$

How well do polynomials approximate arbitrary functions?

Theorem 8 (Weierstrass Approximation Theorem)
Suppose $f \in C([a,b])$. *For each* $\varepsilon > 0$ *there exists a polynomial* $P(x)$ *s.t.*

$$|f(x) - P(x)| < \varepsilon \quad \text{for all } x \in [a,b].$$

Comments:
The uniform bound above has a nice geometrical interpretation (see the diagram below). Unlike Taylor Series, which seeks a good approximation in the neighborhood of a specific point x_0, with polynomial interpolation we seek to bound the maximum error over an entire interval.

For any x, the distance between f and p is less $< \varepsilon$. This is just a re-statement of the theorem.

Figure 2.8: Illustration of Theorem 8

Linear Interpolation

We wish to find a first degree polynomial that interpolates a function with

$$P(x_0) = f(x_0), \quad P(x_1) = f(x_1).$$

The straight line P approximates the graph of f. We wish to find a formula for $P(x)$.

Figure 2.9: Linear interpolation.

Observe:

$$\begin{aligned} P(x) &= f(x_0) + \text{ a fraction of the distance from } f(x_0) \text{ to } f(x_1) \\ &= f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right) \cdot [f(x_1) - f(x_0)]. \end{aligned}$$

Or, after re-arranging (Exercise):

$$\boxed{P(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right) f(x_1).}$$

Observe, when:

$$\begin{aligned} &\underline{x = x_0,} \quad && P(x_0) = 1 \cdot f(x_0) + 0 \cdot f(x_1) = f(x_0). \\ &\underline{x = x_1,} \quad && P(x_1) = 0 \cdot f(x_0) + 1 \cdot f(x_1) = f(x_1). \end{aligned}$$

i.e., $P(x)$ interpolates $f(x)$ at x_0 and x_1.

Theorem 9 (Linear Interpolation Error)
Let $f \in C^2([x_0, x_1])$ (i.e., f, f' and f'' are continuous on $[x_0, x_1]$) and let $P_1(x)$ be the linear polynomial that interpolates f at x_0 and x_1. Then

$$|f(x) - P_1(x)| \leq \frac{M}{8}(x_1 - x_0)^2, \quad M = \max|f''(x)| \ \ \forall\, x \in [x_0, x_1].$$

Example 33
Construct a linear interpolating polynomial to the function $f(x) = \sqrt{x}$ using $x_0 = 1/4$, $x_1 = 1$. What is the upper bound on the error over the interval $[1/4, 1]$ according to the error estimate?

$$\begin{aligned} P_1(x) &= \left(\frac{x - x_1}{x_0 - x_1}\right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right) f(x_1) \\ &= \left(\frac{x - 1}{1/4 - 1}\right) \sqrt{1/4} + \left(\frac{x - 1/4}{1 - 1/4}\right) \sqrt{1} \\ &= -\frac{4}{3}(x - 1)\frac{1}{2} + \frac{4}{3}(x - 1/4) \\ &= \frac{2}{3}x + \frac{1}{3}. \end{aligned}$$

To get the upper bound on the error, we need the 2nd derivative of $f(x) = \sqrt{x}$:

$$f(x) = x^{1/2} \implies f'(x) = \frac{1}{2}x^{-1/2} \implies f''(x) = -\frac{1}{4}x^{-3/2},$$

so $M = \max\left|-\frac{1}{4}x^{-3/2}\right| = \frac{1}{4}|x|^{-3/2}$ for all $x \in [1/4, 1]$.

Now $|x|^{3/2}$ is increasing on $[1/4, 1]$, so $|x|^{-3/2}$ is decreasing on $[1/4, 1]$. Thus

$$M = \frac{1}{4}\left|\frac{1}{4}\right|^{-3/2} = \frac{1}{4} \cdot 4^{3/2} = \frac{1}{4} \cdot 8 = 2.$$

So,

$$\begin{aligned} |\sqrt{x} - p_1(x)| &\leq \frac{M}{8}(x_1 - x_0)^2 \\ &= \frac{2}{8}\left(1 - \frac{1}{4}\right)^2 \\ &= \frac{1}{4} \cdot \left(\frac{3}{4}\right)^2 \\ &= \frac{9}{64} \text{ or } 0.140625. \end{aligned}$$

If we graph the *exact* error $|\sqrt{x} - P_1(x)|$ on $[1/4, 1]$:

Figure 2.10: Exact error in approximating $\sqrt{x}$ with $P_1(x)$.

So the *theoretical* result is not a good error estimate for this problem as we can see that $P_1(x)$ approximates $\sqrt{x}$ well on $[1/4, 1]$.

Piecewise linear interpolation

Use linear interpolation on many subintervals $[x_k, x_{k+1}], \quad k = 0, 1, ..., n-1$

Figure 2.11: Graph of the piecewise linear interpolant of f.

The error is given by:

$$\boxed{|\text{Error}| \leq \max_{0 \leq k \leq n-1} \left|\text{Error on } I_k\right|.}$$

(max error over all subintervals)

Notes:

- To get the error on I_k use Theorem 9.
- Most computers use piecewise linear interpolation to plot graphs.
- The piecewise linear interpolant is useful in numerical integration (see next section).
- The piecewise linear interpolant is of theoretical importance in more advanced numerical analysis.

2.5 The Trapezoid Rule

Introduction

Aim: Find the area under the graph of f between $x = a$ and $x = b$:

Figure 2.12: Area under f on $[a, b]$

Area $= I(f) =$

If f is not integrable we must approximate $I(f)$, e.g., via the Riemann Sum:

$$I(f) \approx \sum_{k=1}^{n} f(x_k)h, \qquad \left(h = \frac{b-a}{n}\right).$$

Figure 2.13: Approximating f with a Riemann Sum

For theoretical purposes the Riemann Sum is useful, however for numerical approximations it is very crude.

The Trapezoid Rule (an improved approximation)

Instead of using a piecewise constant approximation to f, use a piecewise linear approximation. Then integrate the piecewise linear interpolant (see last section).

Single subinterval case

Integrate the linear interpolant $p_1(x)$:

Figure 2.14: Single interval case.

Recall **:**

$$P_1(x) = \left(\frac{x-b}{a-b}\right) f(a) + \left(\frac{x-a}{b-a}\right) f(b),$$

so with $T_1(x) = \int_a^b P_1(x)dx,$

$$T_1(x) = \frac{1}{2}(b-a)(f(a)+f(b)),$$

(Exercise)**.**

Note: this is just the area of a trapezium. How accurate is this approximation?

Theorem 10 (Error Estimate for Trapezoid Rule, single subinterval case)
Let $f \in C^2([a,b])$ and P_1 interpolate f at a and b. Then there exists $\eta \in [a,b]$ s.t.

$$\underbrace{\int_a^b f(x)dx}_{I(f)} = \underbrace{\frac{1}{2}(b-a)(f(a)+f(b))}_{T_1(f)} - \underbrace{\frac{(b-a)^3}{12}f''(\eta)}_{\text{error term}}.$$

Note: with $h = b-a$, error $= \mathcal{O}(h^3)$.

Many subinterval case (composite rule)

Assume we have a uniform partition of $[a,b]$ into n subintervals of length h ('uniform grid').

1 subinterval (as we saw before):

Figure 2.15: Single interval case.

$$T_1(f) = \frac{1}{2}h\bigg(f(x_0) + f(x_1)\bigg).$$

2 subintervals:

Figure 2.16: Double interval case.

$$\begin{aligned} T_2(f) &= \frac{1}{2}h\bigg(f(x_0) + f(x_1)\bigg) + \frac{1}{2}h\bigg(f(x_1) + f(x_2)\bigg) \\ &= \frac{1}{2}h\bigg(f(x_0) + 2f(x_1) + f(x_2)\bigg). \end{aligned}$$

n subintervals:

Figure 2.17: Multi-interval case.

$$T_n(f) = \frac{1}{2}h\bigg(f(x_0) + 2f(x_1) + 2f(x_2) + \ldots + 2f(x_{n-1}) + f(x_n)\bigg).$$

Error in the general ('composite') case

What about the error? In the single subinterval case error $\propto h^3$. Adding up errors over n subintervals yields

$$\text{error } \propto nh^3 = \left(\frac{b-a}{h}\right)h^3 = (b-a)h^2, \text{ which is } \mathcal{O}(h^2).$$

Theorem 11 (Error Estimate for Composite Trapezoid Rule, Uniform Grid) *Let $f \in C^2([a,b])$ and $T_n(f)$ be the n subinterval trapezoid rule approximation to $I(f)$ using a uniform grid. Then there exists an $\eta_h \in [a,b]$ s.t.*

$$I(f) = \int_a^b f(x)dx = T_n(f) - \frac{(b-a)}{12}h^2 f''(\eta_h).$$

Application

Estimate the error via

$$\boxed{|I(f) - T_n(f)| \leq \frac{(b-a)}{12}h^2 M, \text{ where } M = \max|f''(\eta_h)|, \quad \forall\, \eta_h \in [a,b].}$$

Example 34
Use the Trapezoid Rule with $h = \dfrac{1}{5}$ to approximate the integral

$$I := \int_0^1 \left(\frac{\sin(x)}{x}\right) dx = 0.94608307...$$

Arrange the function values in a table:

Note: we will define $\left.\dfrac{\sin(x)}{x}\right|_{x=0} = 1$ (why is this sensible ?)

i	x_i	$f(x_i) = \frac{sin(x_i)}{x_i}$
0	0.0	1.00000
1	0.2	0.99335
2	0.4	0.97355
3	0.6	0.94107
4	0.8	0.89670
5	1.0	0.84147

$$\begin{aligned} T_5(f) &= \frac{1}{2}h\Big(f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + f(x_5)\Big) \\ &= \left(\frac{1}{2}\right)\cdot\left(\frac{1}{5}\right)(1.0 + 2(0.99335) + ... + 2(0.89670) + 0.84147) \\ &= 0.94508 \quad \text{(to 5 d.p.)} \end{aligned}$$

Example 35 (Typical exam question)
Use the Trapezoid Rule to approximate

$$I := \int_0^1 e^{-x^2} dx \quad (\approx 0.746824)$$

with an error of at most $1/2 \times 10^{-4}$. How small does h have to be? What is the least number of subintervals required?

The error formula is:

$$-\frac{(b-a)}{12} \cdot h^2 \cdot f''(\eta), \quad \eta \, \epsilon \, [0,1].$$

Here $f(x) = e^{-x^2}$, $f'(x) = -2x \cdot e^{-x^2}$ and so

$$f''(x) = \frac{2(2x^2-1)}{e^{x^2}} \quad \text{(excercise)}.$$

To bound the error, consider

$$|I(f) - T_n(f)| \leq \frac{(b-a)}{12} h^2 M = \frac{h^2}{12} M, \qquad (*)$$
$$\text{where } M = \max_{\eta \in [0,1]} |f''(\eta)|.$$

How do we maximize $|f''(x)| = \left| \frac{2(2x^2-1)}{e^{x^2}} \right|$ over $[0,1]$?
Consider the function

$$g(x) = \frac{2(2x^2-1)}{e^{x^2}}.$$

To maximize $g(x)$ we want to maximize the numerator and minimize the denominator.

Numerator: Clearly the function $y = 2(2x^2-1)$ is monotonic increasing on $[0,1]$ and so this function has a maximum value at $x = 1$.

Denominator: Clearly the function $y = e^{x^2}$ is monotonic increasing on $[0, 1]$ and so this function is minimized at $x = 0$. Thus

$$|f''(x)| \leq \frac{2(2(1)^2 - 1)}{e^0} = \frac{2}{1} = 2 = M \quad \text{for all } x \in [0, 1],$$

and so from $(*)$ we seek h s.t.

$$|I(f) - T_n(f)| \leq \frac{h^2}{12} 2 \leq \frac{1}{2} \times 10^{-4},$$

or

$$h^2 \leq 0.0003 \text{ or } h \leq 0.0173205...$$

But we need h to divide $b - a = 1$ exactly. Recall $h = \frac{b-a}{n} = 1/n \leq 0.0173205...$, or $n \geq 57.73$. Thus the least number of subintervals is $\underline{n = 58}$ which implies $\underline{h = 0.01724...}$ $(h = 1/n)$.

2.6 Solution of Tridiagonal Linear Systems

Review of Matrices

Square Matrices

A square *matrix* (plural *matrices*) is a rectangular array of numbers (real or complex) of the form:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & & & \cdot \\ \cdot & & a_{ij} & \cdot \\ \cdot & & & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

We say A is n-by-n $(n \times n)$ with order n. The diagonal entries are $a_{11}, a_{22}, \dots, a_{nn}$.

If $\boldsymbol{A}$ is *upper triangular* then $\boldsymbol{A}$ has the form:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & 0 & a_{33} & \dots & a_{3n} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ 0 & 0 & \dots & 0 & a_{nn} \end{bmatrix}, \qquad (a_{ij} = 0 \quad \text{if } i > j).$$

$$\text{e.g. } \begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix}$$

.

A special case that we are concerned with here is when $\boldsymbol{A}$ is *tridiagonal*:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{21} & 0 & & \dots & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \dots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \dots & 0 \\ . & & & & & . \\ . & & & & & . \\ . & & & & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \dots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix}, \qquad (a_{ij} = 0 \text{ if } |i-j| > 1).$$

$$\text{e.g. } \begin{pmatrix} 2 & 3 & 0 & 0 \\ 4 & 1 & 5 & 0 \\ 0 & 2 & 6 & 1 \\ 0 & 0 & 3 & 4 \end{pmatrix}.$$

Vectors

A matrix with 1 row is called a *row vector*, and a matrix with 1 column is called a *column vector*.

e.g. $\quad \boldsymbol{y = \begin{pmatrix} y_1, & y_2, & \dots & , y_m \end{pmatrix}}$ (row vector $\boldsymbol{1 \times m}$)

$$\boldsymbol{x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix}} \quad \text{(column vector } \boldsymbol{n \times 1}\text{)}.$$

Matrix Multiplication (vector case)

In order to multiply two vectors together the number of columns of the first vector must equal the number of rows of the second vector:

$$\boldsymbol{(u_1, u_2, ..., u_n) \cdot \begin{pmatrix} v_1 \\ v_2 \\ . \\ . \\ . \\ v_n \end{pmatrix} := u_1 v_1 + u_2 v_2 + ... u_n v_n}$$

$$\boldsymbol{= \sum_{i=1}^{n} u_i \cdot v_i} \quad \text{(which is a scalar)}.$$

Example 36
Compute the product uv where

$$u = (8, -4, 5) \text{ and } v = \begin{pmatrix} 3 \\ 2 \\ -1 \end{pmatrix}.$$

We have

$$\begin{aligned} uv &= (8, -4, 5) \cdot \begin{pmatrix} 3 \\ 2 \\ -1 \end{pmatrix} \\ &= (8)(3) + (-4)(2) + (5)(-1) \\ &= 11. \end{aligned}$$

Matrix Multiplication (matrix times vector case)

This can be generalized to the 'matrix times vector' setting as follows. Let A be an $n \times n$ matrix and x be an $n \times 1$ column vector. the product Ax is the $n \times 1$ column vector b where each i^{th} entry b_i is obtained by multiplying x by the i^{th} row of A, shown below:

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{i1} & \dots & a_{in} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}, \qquad \text{i.e., } Ax = b. \quad ---- \quad (*)$$

So

$$\begin{aligned} b_i &= a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \dots + a_{in} \cdot x_n \\ &= \sum_{j=1}^{n} a_{ij}\, x_j \quad \text{for } i = 1, 2, \dots, n. \end{aligned}$$

Note: The product Ax is not defined unless the number of columns of A is equal to the number of rows of x.

Example 37
Compute $\boldsymbol{Ax}$ where

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{-1} \\ \mathbf{0} & \mathbf{2} & \mathbf{1} \\ \mathbf{3} & \mathbf{-2} & \mathbf{4} \end{pmatrix} \quad \text{and} \quad \boldsymbol{x} = \begin{pmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{pmatrix}.$$

Just do

$$\begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{-1} \\ \mathbf{0} & \mathbf{2} & \mathbf{1} \\ \mathbf{3} & \mathbf{-2} & \mathbf{4} \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{pmatrix} = \begin{pmatrix} \mathbf{(2)(1)+(3)(2)+(-1)(3)} \\ \mathbf{(0)(1)+(2)(2)+(1)(3)} \\ \mathbf{(3)(1)+(-2)(2)+(4)(3)} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{5} \\ \mathbf{7} \\ \mathbf{11} \end{pmatrix}.$$

Matrices and Systems of Linear Equations

Consider the square system of $\boldsymbol{n}$ linear equations in $\boldsymbol{n}$ unknowns:

$$\left.\begin{array}{c} \boldsymbol{a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1} \\ \boldsymbol{a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2} \\ \boldsymbol{a_{31}x_1 + a_{32}x_2 + \ldots + a_{3n}x_n = b_3} \\ \vdots \qquad\qquad \vdots \\ \boldsymbol{a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n} \end{array}\right\} \quad - - - - - \quad (**)$$

This linear system $(**)$ is equivalent to the matrix equation $\boldsymbol{Ax = b}$ $(*)$ that we saw earlier. In this context $\boldsymbol{A}$ is called the coefficient matrix of the system $\boldsymbol{Ax = b}$.

Example 38
Write the following system of linear equations in matrix form:

$$\left.\begin{array}{rl} \boldsymbol{3x + 4y - z} & \boldsymbol{= 2} \\ \boldsymbol{2x - 3y + z} & \boldsymbol{= -3} \\ \boldsymbol{x + y + 2z} & \boldsymbol{= 1} \end{array}\right\}$$

$$\begin{pmatrix} 3 & 4 & -1 \\ 2 & -3 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}.$$

Augmented System

The single *augmented matrix* associated with $\boldsymbol{Ax = b}$ is $[\boldsymbol{A\, b}]$, or $[\boldsymbol{A|b}]$

$$= \left[\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} & b_n \end{array}\right].$$

So for the example above

$$[A|b] = \left[\begin{array}{ccc|c} 3 & 4 & -1 & 2 \\ 2 & -3 & 1 & -3 \\ 1 & 1 & 2 & 1 \end{array}\right].$$

Elementary Row Operations

The following row operations applied to the augmented system $[\boldsymbol{A|b}]$ leave the solution of the associated linear system $\boldsymbol{Ax = b}$ unchanged:

(1.) Interchange two rows ($\boldsymbol{r_i \leftrightarrow r_j}$).

(2.) Multiply a row by a non-zero constant ($\boldsymbol{kr_i \longrightarrow r_i}$).

(3.) Add a multiple of a row $\boldsymbol{j}$ to another row $\boldsymbol{i}$ ($\boldsymbol{r_i + kr_j \longrightarrow r_i}$)

Aim

We apply the elementary row operations to the augmented matrix $[\boldsymbol{A|b}]$ so that $\boldsymbol{A}$ is in upper triangular form (i.e., the associated linear system is in triangular form). This process is called *Gaussian Elimination*. The process of *Back Substitution*, illustrated in the next example, yields the solution (if the solution exists).

Example 39
Solve the following tridiagonal system of linear equations:

$$\begin{bmatrix} 6 & 1 & 0 & 0 \\ 2 & 4 & 1 & 0 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 1 & 6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 13 \\ 22 \\ 27 \end{bmatrix} \qquad (Ax = b).$$

Consider the associated augmented system:

$$\left[\begin{array}{cccc|c} 6 & 1 & 0 & 0 & 8 \\ \textcircled{2} & 4 & 1 & 0 & 13 \\ 0 & \textcircled{1} & 4 & 2 & 22 \\ 0 & 0 & \textcircled{1} & 6 & 27 \end{array}\right].$$

We use the elementary row operations to eliminate the circled entries. The following procedure (Gaussian Elimination) illustrates an algorithm that can be easily programmed on a computer:

Eliminate $a_{21} = 2$:

$$\text{pivot} = a_{11} = 6$$
$$\text{multiplier} = m_1 = \frac{a_{21}}{a_{11}} = \frac{2}{6} = \frac{1}{3}$$
$$\text{do} \quad r_2 - m_1 \cdot r_1 = r_2 - \left(\frac{1}{3}\right) \cdot r_1 \longrightarrow r_2$$

yielding

$$\left[\begin{array}{cccc|c} 6 & 1 & 0 & 0 & 8 \\ 0 & \frac{11}{3} & 1 & 0 & \frac{31}{3} \\ 0 & 1 & 4 & 2 & 22 \\ 0 & 0 & 1 & 6 & 27 \end{array}\right] \qquad r_2 - \left(\frac{1}{3}\right) \cdot r_1 \longrightarrow r_2.$$

Eliminate $a_{32} = 1$:

$$\text{pivot} = a_{22} = \frac{11}{3}$$

$$\text{multiplier} = m_2 = \frac{a_{32}}{a_{22}} = \frac{3}{11}$$

$$\text{do} \quad r_3 - m_2 \cdot r_2 = r_3 - \left(\frac{3}{11}\right) \cdot r_2 \to r_3$$

yielding

$$\left[\begin{array}{cccc|c} 6 & 1 & 0 & 0 & 8 \\ 0 & \frac{11}{3} & 1 & 0 & \frac{31}{3} \\ 0 & 0 & \frac{41}{11} & 2 & \frac{211}{11} \\ 0 & 0 & 1 & 6 & 27 \end{array}\right] \quad r_3 - \left(\frac{3}{11}\right) \cdot r_2 \to r_3.$$

Eliminate $a_{43} = 1$:

$$\text{pivot} = a_{33} = \frac{41}{11}$$

$$\text{multiplier} = m_3 = \frac{a_{43}}{a_{33}} = \frac{11}{41}$$

$$\text{do} \quad r_4 - m_3 \cdot r_3 = r_4 - \left(\frac{11}{41}\right) \cdot r_3 \to r_4$$

yielding

$$\left[\begin{array}{cccc|c} 6 & 1 & 0 & 0 & 8 \\ 0 & \frac{11}{3} & 1 & 0 & 31/3 \\ 0 & 0 & \frac{41}{11} & 2 & 211/11 \\ 0 & 0 & 0 & \frac{224}{41} & 896/41 \end{array}\right] \quad r_4 - \left(\frac{11}{41}\right) \cdot r_3 \to r_4.$$

The coefficient matrix is now in upper triangular form, so we stop.

The associated matrix equation is

$$\begin{bmatrix} 6 & 1 & 0 & 0 \\ 0 & \frac{11}{3} & 1 & 0 \\ 0 & 0 & \frac{41}{11} & 2 \\ 0 & 0 & 0 & \frac{224}{41} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 31/3 \\ 211/11 \\ 896/41 \end{bmatrix},$$

i.e.

$$\left.\begin{aligned} 6x_1 + x_2 &= 8 \\ \tfrac{11}{3}x_2 + x_3 &= 31/3 \\ \tfrac{41}{11}x_3 + 2x_4 &= 211/11 \\ \tfrac{224}{41}x_4 &= 896/41 \end{aligned}\right\}$$

We solve this via 'back-substitution', i.e., starting from the last equation and working backwards:

$$x_4 = \left(\frac{896}{41}\right)\left(\frac{41}{224}\right) = 4.$$

Substitute into the 3rd equation:

$$\frac{41}{11}x_3 + (2)(4) = \frac{211}{11}$$
$$\implies x_3 = \left(\frac{123}{11}\right)\left(\frac{11}{41}\right) = 3.$$

Substitute into the 2nd equation:

$$\frac{11}{3}x_2 + 3 = \frac{31}{3}$$
$$\implies x_2 = \left(\frac{22}{3}\right)\left(\frac{3}{11}\right) = 2.$$

Substitute into the 1st equation:

$$6x_1 + 2 = 8 \implies x_1 = 1$$

Thus $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $x_4 = 4$.

Notes:

- This algorithm fails if a pivot is zero, causing us to divide by $\mathbf{0}$. It will also fail if the pivot is so small that this leads to overflow/underflow on a computer.
- How do we assess the accuracy of the answers we get? Check the residual $\boldsymbol{r := b - Ax}$. With ∞ precision we expect the residual to be the zero vector $\mathbf{0}$. With finite precision, and rounding, the residual will be *approximately* the zero vector (provided the problem is well-conditioned).
- The Gaussian Elimination procedure for solving a tridiagonal linear system of $\boldsymbol{n}$ equations in $\boldsymbol{n}$ unknowns requires: $\boldsymbol{8n - 7 = \mathcal{O}(n)}$ flops (Exercise), and is thus very fast. Solving a general linear system of equations requires $\boldsymbol{\mathcal{O}(n^3)}$.

Definition 6 (Diagonal Dominance)
Consider the following (real) tridiagonal matrix:

$$\begin{bmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}.$$

If

(i) $|a_1| > |c_1| > 0$,

(ii) $|a_i| > |b_i| + |c_i| > 0, \quad i = 2, \ldots, n-1$,

(iii) $|a_n| > |b_n| > 0$,

then the tridiagonal matrix is called *Diagonally Dominant*.

Theorem 12
If a tridiagonal matrix $\boldsymbol{A}$ is diagonally dominant, then the Gaussian elimination procedure applied to the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ is guaranteed to give the correct solution (within limitations of rounding error).

Example 40
The following matrix is diagonally dominant

$$\begin{bmatrix} -2 & 1 & 0 \\ 3 & 5 & -1 \\ 0 & 2 & 3 \end{bmatrix}.$$

Final Note:
Theorem 12 gives a sufficient condition, (but not a necessary condition), for Gaussian elimination to work (See Ex. 5 in the exercises).

2.7 Two point boundary value problems

We consider a specific class of second order, ordinary differential equations. Physical problems that are position dependent rather than time dependent are described in terms of differential equations with conditions imposed at more than one point, e.g., the bending of a beam supported at 2 ends.

Figure 2.18: Bending of a beam.

Definition 7 (Two point BVP)
A general second order boundary value problem seeks a solution of

$$\begin{cases} u'' = f(x, u, u'), \ a \leq x \leq b \\ u(a) = u_a \\ u(b) = u_b \end{cases}$$

where $u(a)$ and $u(b)$ are prescribed, and $u'' = \dfrac{d^2u}{dx^2}$, and $u' = \dfrac{du}{dx}$

Comparison of IVP and BVP

$$\begin{aligned} \text{IVP}: \quad & u'(a) = S_a \\ \text{BVP}: \quad & u(a) = u_a \\ & u(b) = u_b \end{aligned}$$

Figure 2.19: Graphical representation of prescribed conditions.

Example 41
Discuss the solutions of the following BVP:

$$\left\{\begin{array}{l} u'' = -u + 2\cos(x),\ 0 \leq x \leq \pi \\ u(0) = 0 \\ u(\pi) = 0 \end{array}\right\}$$

It's easy to check that $u(x) = x\sin(x)$ solves this BVP (Exercise). Thus this BVP has a unique solution.

Example 42
Discuss the solutions of the following BVP:

$$\left\{\begin{array}{l} u'' = -u,\ \ 0 \leq x \leq \pi \\ u(0) = 0 \\ u(\pi) = 1 \end{array}\right\}$$

This BVP has no solution. All solutions have the form $u(x) = a\cos(x) + b\sin(x)$. We cannot make this satisfy both boundary conditions simultaneously (Exercise). Thus, there is no solution.

Example 43
Discuss the solutions of the following BVP:

$$\left\{\begin{array}{l} u'' = -u,\ \ 0 \leq x \leq \pi \\ u(0) = 0 \\ u(\pi) = 0 \end{array}\right\}$$

It is easy to check that $u(x) = k\sin(x)$ satisfies the BVP, for all k (Exercise). Thus there are an infinite number of solutions.

Numerical Solution

These were relatively easy BVPs that can be analyzed by standard techniques. However, there are many examples that cannot be solved exactly, hence the need for numerical methods. We apply difference approximations to the derivatives in the BVPs. In particular,

recall from Section 2.2 that we derived:

$$\left.\begin{aligned} u'(x) &= \frac{u(x+h)-u(x-h)}{2h}+\mathcal{O}(h^2), \\ u''(x) &= \frac{u(x-h)-2u(x)+u(x+h)}{h^2}+\mathcal{O}(h^2) \end{aligned}\right\} \quad --- \quad \text{(FDA)}$$

Grid Partitions

Let h refer to the (uniform) partition of the 'space' interval $[a,b]$:

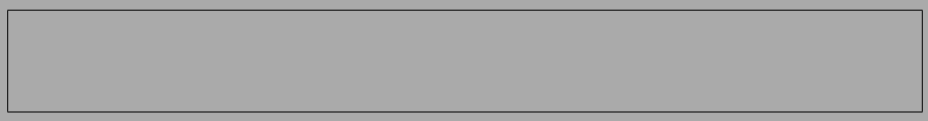

Figure 2.20: Uniform grid.

or $a = x_0 < x_1 < x_2 < \ldots < x_n = b$, so $\boxed{x_k = a + kh,\ k = 0, 1, \ldots, n}$. And from the last node: $\boxed{h = \frac{b-a}{n}}$.

Finite Difference Method

With $u_k \approx u(x_k)$, we replace the derivatives in the BVP with the 'discretized' forms in (FDA) above, i.e. replace:

$$\begin{aligned} &u(x_k) \text{ with } u_k, \\ &u'(x_k) \text{ with } \frac{u_{k+1}-u_{k-1}}{2h}, \\ &u''(x_k) \text{ with } \frac{u_{k-1}-2u_k+u_{k+1}}{h^2}. \end{aligned}$$

Also, replace:

$$\begin{aligned} u(x_0) = u_a &\text{ with } u_0 = u_a, \\ u(x_n) = u_b &\text{ with } u_n = u_b. \end{aligned}$$

Then let k run through the values $1, 2, \ldots, n-1$. In general we have $n+1$ nodes, but after applying the known values at the two endpoints this leaves us with $(n+1)-2 = \boxed{n-1}$ unknowns to solve for.

Example 44
Solve the following BVP using the Finite Difference Method with $h = 1/4$:

$$\left\{\begin{array}{l} -u'' + u = 4e^{-x}(1-x), \quad 0 \leq x \leq 1 \\ u(0) = 0 \\ u(1) = 0 \end{array}\right\}$$

Replace the above BVP with

$$\left\{\begin{array}{l} -\dfrac{u_{k-1} - 2u_k + u_{k+1}}{h^2} + u_k = 4e^{-x_k}(1-x_k) \\ u_0 = 0 \\ u_n = 0 \end{array}\right\} \quad ----- \quad (*)$$

Rewrite $(*)$ (after multiplying through by h^2):

$$\left\{\begin{array}{l} -u_{k-1} + (2+h^2)u_k - u_{k+1} = 4h^2 e^{-x_k}(1-x_k) \\ u_0 = 0 \\ u_n = 0 \end{array}\right\} \quad ----- \quad (**)$$

With $h = 1/4$, $n = \dfrac{b-a}{h} = \dfrac{1-0}{1/4} = 4$, giving 3 unknowns:

Figure 2.21: Grid.

So $(**)$ becomes

$$\left\{\begin{array}{l} -u_{k-1} + \left(\frac{33}{16}\right) u_k - u_{k+1} = \frac{1}{4} e^{-k/4}(1 - k/4), \\ u_0 = 0 \\ u_4 = 0 \end{array}\right\}$$

Now we run through the values of k, using the boundary conditions, to deduce the linear equations:

$$\begin{aligned}
\underline{k=1}: \ -u_0 + \left(\frac{33}{16}\right) u_1 - u_2 &= \frac{1}{4} e^{-1/4} \cdot (1 - 1/4) \\
\text{or} \qquad \left(\frac{33}{16}\right) u_1 - u_2 &= \frac{3}{16} \cdot e^{-1/4} \quad ------ \quad \textbf{(1)} \\
\underline{k=2}: \ -u_1 + \left(\frac{33}{16}\right) u_2 - u_3 &= \frac{1}{4} e^{-1/2} \cdot (1 - 1/2) \\
&= \frac{1}{8} \cdot e^{-1/2} \quad ------ \quad \textbf{(2)} \\
\underline{k=3}: \ -u_2 + \left(\frac{33}{16}\right) u_3 - u_4 &= \frac{1}{4} e^{-3/4} \cdot (1 - 3/4) \\
\text{or} \qquad -u_2 + \left(\frac{33}{16}\right) u_3 &= \frac{1}{16} \cdot e^{-3/4} \quad ------ \quad \textbf{(3)}
\end{aligned}$$

Now **(1)**, **(2)**, and **(3)** constitute a tridiagonal linear system, written in matrix form as:

$$\begin{pmatrix} 33/16 & -1 & 0 \\ -1 & 33/16 & -1 \\ 0 & -1 & 33/16 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} (3/16)e^{-1/4} \\ (1/8)e^{-1/2} \\ (1/16)e^{-3/4} \end{pmatrix}$$
$$= \begin{pmatrix} 0.146025 \\ 0.0758163 \\ 0.0295229 \end{pmatrix} \quad (6 \text{ s.f.})$$

Observe that the coefficient matrix is diagonally dominant. Solving this system via Gaussian Elimination yields (exercise): $u_1 = 0.1422, \ u_2 = 0.1473, \ u_3 = 0.08571$ (4 s.f.).

Checking the answer:
The exact solution to the BVP is given by (exercise)

$$u(x) = x(1-x)e^{-x}.$$

Recall that $u_k \approx u(x_k) = u(kh) = u(k/4)$, so (to 4 s.f.)

$$u_1 \approx u(1/4) = \frac{1}{4}\left(1 - \frac{1}{4}\right)e^{-1/4} = 0.1460,$$
$$u_2 \approx u(1/2) = 0.1516,$$
$$u_3 \approx u(3/4) = 0.08857.$$

Example 45
Solve the following BVP using the Finite Difference Method with $h = 1/6$:

$$\begin{cases} -u'' + 64u' + u = 1, \ ; \ 0 \leq x \leq 1 \\ u(0) = u(1) = 0 \end{cases}$$

Replace the above BVP with

$$\begin{cases} -\left(\frac{u_{k-1} - 2u_k + u_{k+1}}{h^2}\right) + 64\left(\frac{u_{k+1} - u_{k-1}}{2h}\right) + u_k = 1 \\ u_0 = u_n = 0 \end{cases}$$

Multiplying the differential equation by h^2 leads to

$$-u_{k-1} + 2u_k - u_{k+1} + 32h(u_{k+1} - u_{k-1}) + h^2 u_k = h^2$$

or

$$-(32h + 1)u_{k-1} + (2 + h^2)u_k + (32h - 1)u_{k+1} = h^2. \quad - - - \quad (*)$$

With $h = 1/6$, $n = \dfrac{b-a}{h} = \dfrac{1-0}{1/6} = 6$ so there will be 5 unknowns:

Figure 2.22: Grid.

Thus $(*)$ becomes:

$$\begin{cases} \left(-\frac{19}{3}\right)u_{k-1} + \left(\frac{73}{36}\right)u_k + \left(\frac{13}{3}\right)u_{k+1} = \frac{1}{36} \\ u_0 = u_6 = 0 \end{cases}$$

Now running through the values of k and using the b.c.'s yields:

$$\underline{k=1}: \quad \left(-\frac{19}{3}\right)u_0 + \left(\frac{73}{36}\right)u_1 + \left(\frac{13}{3}\right)u_2 = \frac{1}{36}$$

i.e.
$$\left(\frac{73}{36}\right)u_1 + \left(\frac{13}{3}\right)u_2 = \frac{1}{36} \quad ----------- \quad (1)$$

$$\underline{k=2}: \quad \left(-\frac{19}{3}\right)u_1 + \left(\frac{73}{36}\right)u_2 + \left(\frac{13}{3}\right)u_3 = \frac{1}{36} \quad --- \quad (2)$$

$$\underline{k=3}: \quad \left(-\frac{19}{3}\right)u_2 + \left(\frac{73}{36}\right)u_3 + \left(\frac{13}{3}\right)u_4 = \frac{1}{36} \quad --- \quad (3)$$

$$\underline{k=4}: \quad \left(-\frac{19}{3}\right)u_3 + \left(\frac{73}{36}\right)u_4 + \left(\frac{13}{3}\right)u_5 = \frac{1}{36} \quad --- \quad (4)$$

$$\underline{k=5}: \quad \left(-\frac{19}{3}\right)u_4 + \left(\frac{73}{36}\right)u_5 + \left(\frac{13}{3}\right)u_6 = \frac{1}{36}$$

i.e.
$$\left(-\frac{19}{3}\right)u_4 + \left(\frac{73}{36}\right)u_5 = \frac{1}{36} \quad ---------- \quad (5)$$

Equations **(1)**-**(5)** lead to the following system of tridiagonal matrix system:

$$\begin{pmatrix} 73/36 & 13/3 & 0 & 0 & 0 \\ -19/3 & 73/36 & 13/3 & 0 & 0 \\ 0 & -19/3 & 73/36 & 13/3 & 0 \\ 0 & 0 & -19/3 & 73/36 & 13/3 \\ 0 & 0 & 0 & -19/3 & 73/36 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} 1/36 \\ 1/36 \\ 1/36 \\ 1/36 \\ 1/36 \end{pmatrix}$$

Gaussian Elimination and back substitution yield (computer exercise):

$$\left.\begin{aligned} u_1 &= 0.006882 \\ u_2 &= 0.003190 \\ u_3 &= 0.01498 \\ u_4 &= 0.004065 \\ u_5 &= 0.02639 \end{aligned}\right\} \text{4 s.f.}$$

Accuracy

In most cases, halving h (i.e. doubling the number of grid points) will quarter the error. I.e., the method is $\mathcal{O}(h^2)$.

Chapter 3

Root - Finding

Introduction

Given a (usually) nonlinear equation $y = f(x)$, we are interested in finding a value α s.t.

$$f(\alpha) = 0.$$

Note that the solution of *any* nonlinear equation can be turned into a root finding problem by the simple procedure of taking all terms to one side of the equation. We will assume $\alpha \in \mathbb{R}$ (i.e., real). The number α is called a *zero*, or a *root*, illustrated below:

Figure 3.1: Graph of f near α.

3.1 The Bisection Method

We first look at a very simple numerical method for approximating the root of an equation. Recall the following special case of the Intermediate Value Theorem (IMVT).

Corollary 1 (To Theorem 3)
Let f be continuous on $[a, b]$. Then if $f(a)$ and $f(b)$ differ in sign, then by the IMVT there exists an $\alpha \in [a, b]$ s.t. $f(\alpha) = 0$ (i.e., α is a root).

Example 46
Apply the above Theorem to $\sin(x)$ on $[-\pi/2, \pi/2]$:

$$\left.\begin{array}{r} \sin(\pi/2) = 1 > 0 \\ \sin(-\pi/2) = -1 < 0 \end{array}\right\} \quad \Longrightarrow \quad \text{by the IMVT}$$

that $\sin(x)$ has a root in $[-\pi/2, \pi/2]$ (which of course it does; $\alpha = 0$).

Graphical interpretation

Figure 3.2: Illustration of Corollary.

$$f(a)f(b) < 0.$$

Example 47
Find the approximate location of the root to the equation $f(x) = x + e^x$.
We wish to solve

$$x + e^x = 0 \text{ or } e^x = -x$$

We cannot solve this analytically. Graphically, the root corresponds to the the intersection of the graphs of $y = e^x$ and $y = -x$:

Figure 3.3: Intersection of graphs.

We can clearly see that a solution exists. Furthermore,

$$\left.\begin{array}{ll} f(0) & = 0 + e^{0} \quad = 1 > 0 \\ f(-1) & = -1 + e^{-1} \approx -0.632 < 0 \end{array}\right\} \quad \Longrightarrow \quad \text{by the IMVT}$$

there exists a root of $f(x)$ in $[-1, 0]$. Thus a first estimate is $\alpha \approx -0.5$.

Bisection Method

The process of repeatedly 'bisecting' the interval using the IMVT to get closer to the root is called the *Bisection Method*, illustrated in the next example.

Example 48

Use the Bisection Method to get an improved estimate of the root in the previous example:
We bisect the interval $[-1, 0]$:
$f(-0.5) = -0.5 + e^{-0.5} \approx 0.106 > 0$, thus we have

$$\left.\begin{array}{l} f(-1) < 0 \\ f(-0.5) > 0 \end{array}\right\} \Longrightarrow \text{ by the IMVT}$$

that the root lies in $[-1, -0.5]$.

Figure 3.4: Bisection.

Bisect the interval again: $f(-0.75) = -0.75 + e^{-0.75} \approx -0.277 < 0$, and so

$$\left.\begin{array}{l} f(-0.75) < 0 \\ f(-0.5) > 0 \end{array}\right\} \Longrightarrow \text{ by the IMVT}$$

that the root lies in $[-0.75, -0.5]$., etc. ($\alpha \approx \frac{-0.75-0.5}{2} = -0.625$). We can continue bisecting the interval to get a better and better estimate for α.

Figure 3.5: Bisection.

Let x_k = the approximation to α after k bisections. Then we have $x_1 = -0.5$, $x_2 = -0.75$, $x_3 = -0.625$, etc.

Error Bound Formula

There is a simple result that allows us to assess the accuracy of the Bisection Method, which is independent of the function itself.

Theorem 13 (Error Bound Formula for the Bisection Method)
Let $\alpha \in [a,b]$ be a root of $f(x)$, and let x_k be the approximation to α after k bisections. Then

$$|\alpha - x_k| \leq \frac{b-a}{2^k}$$

Sketch Proof

Let k be the number of bisections, and $\alpha =$ root $\in [a,b]$.

$\underline{k=1}$:

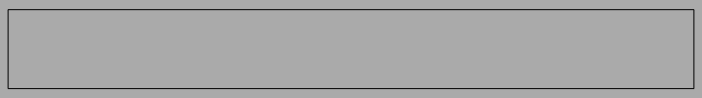

Figure 3.6: 1st Bisection.

$$x_1 = \frac{a+b}{2} \text{ so } |\alpha - x_1| \leq \frac{b-a}{2^1} \quad (= \text{ case if } \alpha = a \text{ or } b).$$

With no loss in generality, suppose $\alpha \in [x_1, b]$:

$\underline{k=2}$:

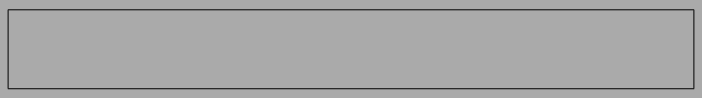

Figure 3.7: 2nd Bisection.

$$x_2 = \frac{x_1+b}{2} \text{ so } |\alpha - x_2| \leq \frac{b-a}{2^2} \quad (= \text{ case if } \alpha = x_1 \text{ or } b).$$

Continuing this process leads to the required result. □

Corollary 2 (To Theorem 13)
To achieve an accuracy of $|\alpha - x_k| \leq \varepsilon$ in the Bisection Method it suffices to take

$$k \geq \frac{\ln(b-a) - \ln(\varepsilon)}{\ln(2)}.$$

Proof
From Theorem 13 we seek k such that $|\alpha - x_k| \leq \frac{b-a}{2^k} \leq \varepsilon$. This yields:

$$\begin{aligned} & b - a \leq \varepsilon 2^k \\ \text{or} \quad & \ln(b-a) \leq \ln(\varepsilon \cdot 2^k) = \ln(\varepsilon) + k\ln(2) \\ \Longrightarrow \quad & k \geq \frac{\ln(b-a) - \ln(\varepsilon)}{\ln(2)}. \end{aligned}$$

□

Example 49
Using the Bisection Method, how many steps (bisections) are needed to obtain an error of no more than 10^{-6}, if we know that the root lies in $[-1, 0]$?

We have $a = -1,\ b = 0,\ \varepsilon = 10^{-6}$. We seek k s.t.

$$\begin{aligned} |\alpha - x_k| &\leq \frac{b-a}{2^k} \leq \varepsilon \\ \text{or} \quad \frac{0-(-1)}{2^k} &\leq 10^{-6} \\ 1 &\leq 10^{-6} \cdot 2^k \\ \ln(1) &\leq \ln(10^{-6}) + k \cdot \ln(2) \\ k &\geq -\frac{\ln(10^{-6})}{\ln(2)} = 19.9315\ldots \end{aligned}$$

Hence the answer is $k = 20$.
number of bisections

3.2 Newton's Method

Introduction

Newton's Method is a faster method than the Bisection Method for approximating the roots of an equation $f(x) = 0$, but requires us to know the derivative of the function f, namely, $f'(x) = \frac{df}{dx}$. We recall that the derivative of a function f at a point x_0 can be represented graphically by the slope of a tangent line to the graph at the point $(x_0, f(x_0))$.

Graphical Derivation

Consider the graph of a function $y = f(x)$ with a root α (i.e. $f(\alpha) = 0$). Consider an initial 'guess' x_0 to α. Then to get a better approximation to α, denoted x_1, use the tangent line to the graph at x_0 (illustrated below):

Figure 3.8: Graphical illustration of Newton's Method.

Slope of tangent line $= f'(x_0) = \dfrac{f(x_0)}{x_0 - x_1}$, at $x = x_0$, so

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

To improve this estimate, use the tangent line at x_1 to get x_2:

Slope of tangent line $= f'(x_1) = \dfrac{f(x_1)}{x_1 - x_2}$, at $x = x_1$, so

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

and so on, leading to the following iterative formula:

Newton's Iterative Method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots$$

given $x_0 =$ an initial guess.

Failure of Newton's Method

The method fails if $f'(x_n) = 0$, which corresponds to a horizontal tangent line at $x = x_n$:

Figure 3.9: Failure at step n.

The method also fails if $|f'(x_n)|$ is so small that it leads to overflow/underflow on a computer.

Example 50

(a) Write down Newton's Method for finding the single root of $f(x) = x^3 + x - 1$. Simplify the computation as much as possible and check your answer.

(b) Do 5 iterations of Newton's Method for the function given in (a), starting from a 'sensible' initial guess:

$f'(x) = 3x^2 + 1$, so

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \\ &= x_n - \frac{(x_n^3 + x_n - 1)}{3x_n^2 + 1} \\ &= \frac{2x_n^3 + 1}{3x_n^2 + 1.} \quad - - - - - - - - - \quad (*) \end{aligned}$$

Quick check of the algebra: if $x_{n+1} = x_n = \alpha$ (i.e., we are at the root), then we have

$$\alpha = \frac{2\alpha^3 + 1}{3\alpha^2 + 1}$$

or $3\alpha^3 + \alpha = 2\alpha^3 + 1$ or $\alpha^3 + \alpha - 1 = 0$. ✓

Observe:

$$\left.\begin{matrix} f(0) = -1 \\ f(1) = +1 \end{matrix}\right\} \implies \text{by the IMVT that } \alpha \in [0, 1].$$

Furthermore, as $f' > 0$ the function f is strictly monotonic increasing and thus there is only one root. Choose $x_0 = 0.5$.

Figure 3.10: Graph of f

From $(*)$ (incorrect significant figures to the right of $\big|$)

$$x_1 = \frac{2x_0^3+1}{3x_0^2+1} = \frac{2(0.5)^3+1}{3(0.5)^2+1} = 0.714285714286\ldots$$
$$x_2 = \frac{2x_1^3+1}{3x_1^2+1} = 0.68\big|3179723502\ldots$$
$$x_3 = \frac{2x_2^3+1}{3x_2^2+1} = 0.68232\big|8423305\ldots$$
$$x_4 = \frac{2x_3^3+1}{3x_3^2+1} = 0.682327803828\ldots$$
$$x_5 = \frac{2x_4^3+1}{3x_4^2+1} = 0.682327803828\ldots$$

The last two answers are the same. This suggests that x_5 is correct to the number of decimal digits shown.

Checking our answers in MATLAB

- Check the residual is close to zero:

```
>> x = 0.682327803828;
>> x^3+x-1
ans =
         -4.640732242933154e-14
```

- Get MATLAB to find the roots:

```
>> p = [1 0 1 -1];   % coefficients in x^3, x^2, x^1, and x^0
>> r = roots(p)
r =
 -0.341163901914010 + 1.161541399997253i
 -0.341163901914010 - 1.161541399997253i
  0.682327803828019
```

Convergence

Newton's Method is characteristic of a "quadratically convergent" method[2], in that once convergence starts to take hold, the number of correct places in x_n approximately doubles on each iteration (although it may also be only "linearly convergent"[1]). Newton's Method may also diverge, i.e., successive iterates get further and further away from the root.

Theorem 14 (Convergence of Newton's Method)
Assume $f(x)$, $f'(x)$, and $f''(x)$ are continuous for all x in some neighborhood of α, and assume $f(\alpha) = 0$, $f'(\alpha) \neq 0$. Then if x_0 is chosen sufficiently close to α, Newton's Method converges quadratically.

[2]We say that the 'order of convergence' is $\mathbf{2}$.
[1]I.e., the 'order of convergence is $\mathbf{1}$'

3.3 How to Stop Newton's Method

We want some guidance as to when to stop Newton's Method in practice.

Recall that for appropriate conditions on f,

$$f(x_0 + h) = f(x_0) + hf'(\xi), \quad \xi \text{ between } x_0 \text{ and } x_0 + h. \quad ---- \quad (1)$$

Set $x_0 \longrightarrow x_n, \quad h \longrightarrow e_n = \alpha - x_n$. Thus (1) becomes:

Figure 3.11: Real number line.

$$0 = f(\underbrace{x_n + e_n}_{\alpha}) = f(x_n) + e_n f'(\xi), \quad \xi \text{ between } x_n \text{ and } \alpha.$$

so

$$e_n = -\frac{f(x_n)}{f'(\xi)} \quad ------ \quad (2)$$

Recall Newton's Method:

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

Rearranging gives

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

Substitute this into (2):

$$\alpha - x_n = e_n = (x_{n+1} - x_n) \cdot \frac{f'(x_n)}{f'(\xi)}, \qquad (f'(\xi) \neq 0)$$

so

$$|\alpha - x_n| = |x_{n+1} - x_n| \cdot c_n,$$

where

$$c_n = \left|\frac{f'(x_n)}{f'(\xi)}\right|, \qquad (f'(\xi) \neq 0).$$

Stopping Criterion

Thus the error $|\alpha - x_n|$ is a multiple of $|x_{n+1} - x_n|$ (computable!). Thus stop Newton's Method when $|x_{n+1} - x_n|$ is sufficiently small. In practice one should also check that $f(x_n) \approx 0$. Thus stop when

$$\boxed{|f(x_{n+1})| + |x_{n+1} - x_n| < \text{ tol}}$$

where 'tol' is a small tolerance to be chosen.

Graphical justification

We show graphically that we need to check that *both* $|f(x_{n+1}|$ and $|x_{n+1} - x_n|$ are small during Newton's Iterative procedure.

Case (i): ($|f(x_{n+1})|$ small, but $|x_{n+1} - x_n|$ large)

Figure 3.12: Illustration of stopping criteria.

Case (ii): ($|f(x_{n+1})|$ large, but $|x_{n+1} - x_n|$ small)

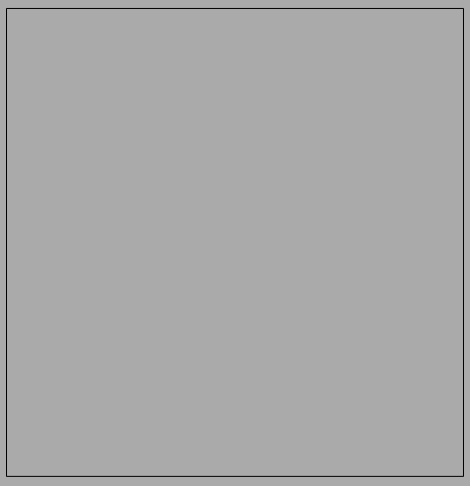

Figure 3.13: Illustration of stopping criteria.

Example 51
Approximate the cube root of 7 using Newton's Method. Stop the method with a tolerance in the above error check of $\mathbf{10^{-4}}$.

Set $f(x) = 7 - x^3$.
Then $f(\alpha) = 0 \Leftrightarrow \alpha = \sqrt[3]{7}$.

Newton's Method is:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{(7 - x_n^3)}{-3x_n^2} = \frac{2x_n^3 + 7}{3x_n^2}.$$

We note:

$$\left.\begin{array}{l} f(2) = 7 - 8 = -1 \\ f(1) = 7 - 1 = 6 \end{array}\right\} \Longrightarrow$$

by the IMVT that $\alpha \in [1, 2]$.

Thus choose $x_0 = 1.5$. This leads to the following tabulated values:

n	x_n	$\lvert x_n - x_{n-1}\rvert$	$\lvert f(x_n)\rvert$
0	1.5	—	3.625
1	2.037...	0.5370...	1.4527...
2	1.920...	0.1166...	0.0816...
3	1.9129...	0.0073...	0.0003...
4	1.91293118...	$< 10^{-5}$	$< 10^{-9}$

Thus after 4 iterations we stop as $|f(x_4)| + |x_4 - x_3| < 10^{-4}$.

3.4 The Secant method

Recall Newton's Method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots, \qquad (x_0 \text{ given}). \quad - - - - \quad (*)$$

What do we do if the exact calculation of f' is either impossible (i.e., there is no analytical solution), or impractical? *Answer*: use the secant line joining $(x_n, f(x_n))$ and $(x_{n-1}, f(x_{n-1}))$ to approximate $f'(x_n)$.

$$\begin{aligned} f'(x_n) &\approx \text{ slope of the secant line} \\ &= \frac{f(x_{n-1}) - f(x_n)}{x_{n-1} - x_n} \\ &= \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \quad - - - - - \quad (**) \end{aligned}$$

Figure 3.14: Illustration of the Secant Method.

Now substitute $(**)$ into $(*)$, giving us:

Secant Method

$$x_{n+1} = x_n - f(x_n)\left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right), \quad n = 1, 2, 3, \ldots$$

given initial approximations x_0 and x_1.

Comments

- The method requires 2 values, x_0 and x_1 to get it started.
- At each stage this method requires a single function evaluation $f(x_n)$ (with $f(x_{n-1})$ calculated in the last step), while Newton's Method needs two function evaluations ($f(x_n)$ and $f'(x_n)$) per step.
- We don't need to know f'.
- The rate of convergence of the Secant Method is similar to the rate of convergence of Newton's Method (a bit slower, but better than linearly convergent).

Example 52

Do 3 steps of the Secant Method to find the root of $f(x) = 7 - x^3$ (i.e., $\sqrt[3]{7}$), using $x_0 = 1$ and $x_1 = 2$.

The Secant Method is

$$\begin{aligned} x_{n+1} &= x_n - f(x_n) \cdot \left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right) \\ &= x_n - (7 - x_n^3) \cdot \left(\frac{x_n - x_{n-1}}{(7 - x_n^3) - (7 - x_{n-1}^3)}\right) \\ &= x_n - (7 - x_n^3) \cdot \left(\frac{x_n - x_{n-1}}{x_{n-1}^3 - x_n^3}\right) \\ &= x_n - \frac{(x_n^3 - 7)}{(x_{n-1}^2 + x_{n-1} \cdot x_n + x_n^2)} \quad ----- \quad (\star), \end{aligned}$$

where we used $\frac{b^3 - a^3}{b - a} = b^2 + ab + a^2$ with $b = x_{n-1}$, $a = x_n$.

With $x_0 = 1$ and $x_1 = 2$ we iterate via $(\star)$ as follows:

$$\begin{aligned}
\underline{n=1}: \quad x_2 &= x_1 - \frac{(x_1^3 - 7)}{(x_0^2 + x_0 \cdot x_1 + x_1^2)} \\
&= 2 - \frac{(2^3 - 7)}{(1^2 + (1) \cdot (2) + 2^2)} = 1.857\ldots \\
\underline{n=2}: \quad x_3 &= x_2 - \frac{(x_2^3 - 7)}{(x_1^2 + x_1 \cdot x_2 + x_2^2)} \\
&= 1.857\ldots - \frac{((1.857\ldots)^3 - 7)}{(2^2 + (2) \cdot (1.857\ldots) + (1.857\ldots)^2)} = 1.910\ldots \\
\underline{n=3}: \quad x_4 &= x_3 - \frac{(x_3^3 - 7)}{(x_2^2 + x_2 \cdot x_3 + x_3^2)} = 1.9130059\ldots
\end{aligned}$$

Note: $\sqrt[3]{7} \approx 1.91293\ldots$, so x_4 is correct to 4 s.f.

3.5 Fixed Point Iteration

A fixed point for a function g is a number α s.t. $g(\alpha) = \alpha$, illustrated below:

Figure 3.15: Illustration of a fixed point.

Clearly the point $(\alpha, g(\alpha))$ lies on the line $y = x$. What does this have to do with finding roots?

Consider the function

$$f(x) := x - g(x)$$

Then if α is a fixed point of g, i.e. $g(\alpha) = \alpha$, then

$$\begin{aligned} f(\alpha) &= \alpha - g(\alpha) \\ &= \alpha - \alpha = 0 \end{aligned}$$

i.e. α is a root of f.

Usefulness

Finding the roots of a nonlinear function f can often be turned into the problem of finding the fixed points of another function g (which in some cases is easier to do). With $f(x) := x - g(x)$ we have shown

$$\boxed{\alpha \text{ is a root of } f \iff \alpha \text{ is a fixed point of } g}$$

.

Example 53

Find the fixed points of $g(x) = x^2 - 6$.

We seek α s.t. $g(\alpha) = \alpha$, i.e.

$$\begin{aligned} &\alpha^2 - 6 = \alpha \\ &\implies \alpha^2 - \alpha - 6 = 0 \\ &\implies (\alpha - 3) \cdot (\alpha + 2) = 0 \\ &\implies \alpha = 3 \text{ or } \alpha = -2 \end{aligned}$$

Example 54

Find the fixed points of $g(x) = \frac{1}{2}e^{-x}$, i.e. find α s.t.

$$\alpha = \frac{1}{2}e^{-\alpha} \quad ???$$

We can't solve this analytically, so we need a numerical method.

Fixed Point Iteration

To approximate the fixed points of a function $g(x)$, iterate via

$$\boxed{\begin{array}{l} x_{n+1} = g(x_n), \quad n = 0, 1, 2, \ldots, \\ x_0 \text{ to be chosen} \end{array}}$$

This iterative process may or may not converge – details to follow. The function g is called an *iteration function*.

Example 55
Apply fixed point iteration to the previous example with $x_0 = 0$ (i.e., we seek the fixed point(s) of $g(x) = \frac{1}{2}e^{-x}$, or equivalently, the root(s) of $f(x) := x - \frac{1}{2}e^{-x}$):

$$x_{n+1} = \frac{1}{2}e^{-x_n}, \quad x_0 = 0.$$

$$\begin{aligned}
\underline{n = 0}: \quad & x_1 = \frac{1}{2}e^{-x_0} = \frac{1}{2}e^{0} = \frac{1}{2} \\
\underline{n = 1}: \quad & x_2 = \frac{1}{2}e^{-x_1} = \frac{1}{2}e^{-1/2} = 0.303265\ldots \\
\underline{n = 2}: \quad & x_3 = \frac{1}{2}e^{-x_2} = \frac{1}{2}e^{(-0.303\ldots)} = 0.369201\ldots \\
\underline{n = 3}: \quad & x_4 = \frac{1}{2}e^{-x_3} = 0.345643025214\ldots \quad \text{etc.}
\end{aligned}$$

How can we check the answer? If $x_n = \alpha$, then $g(\alpha) - \alpha = 0$, so we expect the residual $g(x_n) - x_n \approx 0$ for a x_n 'close' to α. Consider

$$\begin{aligned}
g(x_4) - x_4 &= \frac{1}{2}e^{(-0.345\ldots)} - 0.345\ldots \\
&= 0.00823\ldots \approx 0.
\end{aligned}$$

Alternatively, set

$$\begin{aligned} f(x) &= x - g(x) \\ &= x - \frac{1}{2}e^{-x}, \end{aligned}$$

and use Newton's Method to approximate the root(s) of f, and compare the answers with the fixed point answers.

Questions

- When does an iteration function g have a fixed point?
- If it has a fixed point, is it unique? (I.e., if $g(\alpha_1) = \alpha_1$ and $g(\alpha_2) = \alpha_2$ does $\alpha_1 = \alpha_2$?)
- And finally, under what conditions does the iterative process $x_{n+1} = g(x_n)$, x_0 given, converge?

Theorem 15
Sufficient conditions for existence/uniqueness of fixed points and convergence of Fixed Point Iteration

Assume the following two conditions on the function g hold:

(i) $g : [a, b] \subset [a, b]$ *(i.e.* $a \leq g(x) \leq b$ *for all* $x \in [a, b]$*).*
(ii) $|g'(x)| \leq L < 1$ *for all* $x \in (a, b)$*.*

Then

(a) There exists a unique fixed point of g in $[a, b]$.
(b) For any $x_0 \in [a, b]$ the sequence $x_{n+1} = g(x_n)$ converges.

Graphical Interpretation of the above Theorem

Figure 3.16: Illustration of the Fixed Point Theorem.

Observe that $g : [a, b] \subset [a, b]$, and $|g'(x)| < 1$.

Notes:

- These are *sufficient* conditions, but not *necessary* (i.e., in certain cases they may be violated, but the conclusions of the theorem still hold).
- Condition (i) in the theorem implies that g has at least 1 fixed point in $[a, b]$ ('Existence').
- Condition (ii) in the theorem implies that the fixed point of g is unique ('Uniqueness').

Proof of Existence & Uniqueness of α

Existence: let $f(x) := x - g(x)$.

$$\left.\begin{array}{l} g : [a, b] \subset [a, b] \implies \\ f(a) = a - g(a) < 0 \\ f(b) = b - g(b) > 0 \end{array}\right\} \implies \text{by the IMVT}$$

that $f(x) := x - g(x)$ has at least 1 root in $[a, b]$ $\implies$ g has at least one fixed point in $[a, b]$.

Uniqueness:

Suppose $|g'(x)| < 1$ and that both α_1 and α_2 are fixed points in $[a, b]$, with the assumption that $\alpha_1 \neq \alpha_2$. By the Mean Value Theorem (for the Differential Calculus) a number ξ exists between α_1 and α_2, and hence in (a, b) with

$$\frac{g(\alpha_2) - g(\alpha_1)}{\alpha_2 - \alpha_1} = g'(\xi). \quad - - - - \quad (*)$$

Thus

$$\begin{aligned} |\alpha_2 - \alpha_1| &= |g(\alpha_2) - g(\alpha_1)| \quad \text{(as } \alpha_1 \text{ and } \alpha_2 \text{ are fixed points)} \\ &= |g'(\xi)| \cdot |\alpha_2 - \alpha_1| \quad \text{using } (*) \\ &< |\alpha_2 - \alpha_1| \quad \text{using } |g'(x)| < 1 \text{ for all } x \in (a, b). \end{aligned}$$

Contradiction. This contradiction must come from the only assumption, $\alpha_1 \neq \alpha_2$. Hence $\alpha_1 = \alpha_2$ and the fixed point in $[a, b]$ is unique. (Convergence: not proved here). □

Example 56
(a) Use the Fixed Point Theorem to show that $g(x) = \cos(x)$ has a unique fixed point in $[0, 1]$. (b) Apply Fixed Point Iteration to approximate the fixed point.

Consider the graph of $g(x) = \cos(x)$:

Figure 3.17: Graph of $\cos(x)$.

(i) Observe

$$\left.\begin{aligned} g(0) &= \cos(0) = 1 \\ g(1) &= \cos(1) < 1 \end{aligned}\right\}$$

and as $g(x)$ is monotonically decreasing on $[0,1]$, $g(x)$ between 0 and 1, i.e. $g : [0,1] \subset [0,1]$. ✓

(ii)

$$\begin{aligned} g'(x) &= -\sin(x) \\ |g'(x)| &= |\sin(x)| \leq 1 \quad \text{(equal 1, for } x = \pi/2 \approx 1.57) \end{aligned}$$

so $|g'(x)| < 1$ for all $x \in (0,1)$. ✓

Thus by Theorem 15, $g(x) = \cos(x)$ has a fixed point α on $[0,1]$. Furthermore, the following iterative process converges:

$$x_{n+1} = \cos(x_n), \quad \text{for all } x_0 \in [0,1].$$

E.g.,

$$\begin{aligned} x_0 &= 0.5 \\ &\Longrightarrow x_1 = \cos(0.5) = 0.877\ldots \\ &\Longrightarrow x_2 = \cos(0.877\ldots) = 0.639\ldots \\ &\Longrightarrow x_3 = \cos(0.639\ldots) = 0.802\ldots \\ &\quad\vdots \qquad\qquad\quad \vdots \\ &\Longrightarrow x_{70} = 0.739085\ldots \end{aligned}$$

Notes:

- the solution to $f(x) = x - \cos(x) = 0$ is 0.739085...
- to do fixed point iteration on the calculator efficiently, use the **ANS** key.
  step (i): type 0.5, then **ENTER**
  step (ii): type $\cos(\text{ANS})$, then **ENTER**
  step (iii): press **ENTER**
  step (iv): repeat from (iii)

(everybody tries in class)

Example 57 (Exercise)
Apply Theorem 15 to $g(x) = \frac{1}{2}e^{-x}$ to show that a unique fixed point of g exists on $[0, 1]$

Corollary 3 (Error Bound Formula)
Assume the conditions of Theorem 15 hold, so the iterative procedure $x_{n+1} = g(x_n)$ converges for all $x_0 \in [a, b]$. Then

$$|x_n - \alpha| \leq \left(\frac{L^n}{1-L}\right)|x_1 - x_0|, \quad n \geq 1$$

(Note: L here refers to $|g'(x)| \leq L < 1$).

Example 58
Given $g(x) = x^3 - 0.5x + 0.25$ and suppose the 2 assumptions of the Fixed Point Theorem hold on $[0, 1/2]$. With $x_0 = 0$ find n s.t. |error| $< 10^{-4}$.

Figure 3.18: Graph of $|g'(x)|$.

Finding L:

$|g'(x)| = |3x^2 - 0.5|, \forall x \in (0, 1/2)$.
(roots of $(3x^2 - 0.5)$ are $\approx \pm 0.408$).
Notice: $|g'(0)| = 0.5$, and $|g'(0.5)| = 0.25$.

Thus on $(0, 1/2)$, $|g'(x)| \leq 0.5 = L < 1$.

Finding x_1:

$$\begin{aligned} x_{n+1} &= g(x_n) \\ &= x_n^3 - 0.5x_n + 0.25, \ x_0 = 0, \\ \implies x_1 &= 0.25. \end{aligned}$$

Finding n: From the formula we seek n s.t.

$$\begin{aligned} |x_n - \alpha| &\leq \left(\frac{L^n}{1-L}\right)|x_1 - x_0| < 10^{-4} \\ &\implies \frac{0.5^n}{(1-0.5)}|0.25 - 0| < 10^{-4} \\ &\implies (0.5)^n < 2 \times 10^{-4} \\ &\implies n\ln(0.5) < \ln(2 \times 10^{-4}) \\ &\implies n > 12.287\ldots \\ &\text{so } \underline{n = 13}. \end{aligned}$$

Chapter 4

Interpolation and Approximation

We generalize our earlier work on linear interpolation to polynomial interpolation of arbitrary degree. The use of more nodes (> 2) allows us to construct an interpolant that better matches the graphs of nonlinear functions.

4.1 Lagrange Interpolation

Given tabulated nodes (not necessarily equalled spaced) and corresponding function values

x_0	x_1	...	x_{n-1}	x_n
y_0	y_1	...	y_{n-1}	y_n

where $y_i = f(x_i),\ i = 0, 1, ..., n$, we seek a polynomial of degree n, denoted $P_n(x)$ s.t.

$$P_n(x_i) = y_i, \text{ for } i = 0, 1, ..., n. \quad ----- \quad (\star)$$

(Reminder: we say in this case that P *interpolates* the function f at the nodes x_i).

Lagrange Basis Functions

To construct $P_n(x)$ we first assign to each x_i, a *Lagrange Basis Function*, $L_i(x)$ s.t.

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1, \text{ if } i = j, \\ 0, \text{ if } i \neq j. \end{cases}$$

(Note: the function δ_{ij} is called the *Kronecker delta* function.)

E.g., given 3 nodes

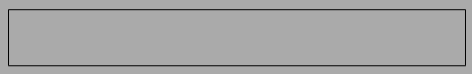

Figure 4.1: Three nodes.

We have:

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \quad \text{is zero when } x = x_1 \text{ or } x_2, \text{ but 1 if } x = x_0.$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \quad \text{is zero when } x = x_0 \text{ or } x_2, \text{ but 1 if } x = x_1.$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \quad \text{is zero when } x = x_0 \text{ or } x_1, \text{ but 1 if } x = x_2.$$

We can generalize this via

$$L_k(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{k-1})(x - x_{k+1})\dots(x - x_n)}{(x_k - x_0)(x_k - x_1)\dots(x_k - x_{k-1})(x_k - x_{k+1})\dots(x_k - x_n)}.$$

Note that the numerator is 'missing' $(x - x_k)$ and that the denominator is the same as the numerator with $x \longrightarrow x_k$. We write the above in a shorter notation, as

$$\boxed{L_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \left(\frac{x - x_j}{x_k - x_j} \right). \quad - - - - - \quad (\star\star)}$$

Note: this function is of degree n (*not* degree $n + 1$, due to the 'missing' factor).

Lagrange Polynomial

To construct a function that satisfies $(\star)$ set

$$\boxed{P_n(x) = \sum_{k=0}^{n} f(x_k) L_k(x).}$$

For example, if $n = 2$ we have:

$$P_2(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x).$$

So e.g., with $x = x_1$ we have:

$$\begin{aligned} P_2(x_1) &= f(x_0) \underbrace{L_0(x_1)}_{=0} + f(x_1) \underbrace{L_1(x_1)}_{=1} + f(x_2) \underbrace{L_2(x_1)}_{=0} \\ &= f(x_1), \end{aligned}$$

as required.

Example 59
Construct the Lagrange polynomial that interpolates the function $f(x) = e^x$ at the nodes 0, $1/2$, 1:

$$\text{We have } x_0 = 0, \ x_1 = \frac{1}{2}, \ x_2 = 1, \quad \text{hence } n = 2.$$

So we construct a quadratic approximation $P_2(x)$. First we construct the Lagrange Basis

Functions:

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-1/2)(x-1)}{(0-1/2)(0-1)} = 2(x-1/2)(x-1),$$
$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-0)(x-1)}{(1/2-0)(1/2-1)} = -4x(x-1),$$
$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-0)(x-1/2)}{(1-0)(1-1/2)} = 2x(x-1/2).$$

Thus

$$\begin{aligned} P_2(x) &= \underbrace{f(x_0)}_{e^0=1}\cdot L_0(x) + \underbrace{f(x_1)}_{e^{1/2}}\cdot L_1(x) + \underbrace{f(x_2)}_{e}\cdot L_2(x) \\ &= 2(x-1/2)(x-1) + e^{1/2}(-4x(x-1)) + e\cdot 2x(x-1/2) \\ &\text{(or after some manipulation)} \\ P_2(x) &= 1 + (-e+4\sqrt{e}-3)x + (2e-4\sqrt{e}+2)x^2. \end{aligned}$$

Check:

$$P_2(0) = 1 = f(0)\ \checkmark$$
$$P_2(1/2) = \sqrt{e} = f(1/2)\ \checkmark$$
$$P_2(1) = e = f(1)\ \checkmark$$

Advantage of Lagrange Interpolation

It provides us with a neat formula for constructing the interpolant. Consider the alternative – determine coefficients $a_0, a_1, ..., a_n$ in the general formula

$$P_n(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n,$$

using the data

x_0	x_1	...	x_{n-1}	x_n
y_0	y_1	...	y_{n-1}	y_n.

Now as $P_n(x_i) = y_i,\ i = 0, 1, ..., n$, this leads to the following system of simultaneous linear equations:

$$\left.\begin{array}{rl} y_0 &= P_n(x_0) = a_0 + a_1 x_0 + \cdots + a_n x_0^n \\ y_1 &= P_n(x_1) = a_0 + a_1 x_1 + \cdots + a_n x_1^n \\ \vdots & \qquad\qquad \vdots \\ y_n &= P_n(x_n) = a_0 + a_1 x_n + \cdots + a_n x_n^n \end{array}\right\},$$

or written in matrix form:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \text{or } V_n a = y.$$

For large n this involves much work. However, the system is of theoretical interest and can be used to prove that the Lagrange Interpolant $P_n(x)$ is uniquely determined by the nodes (see below). The matrix V_n is called the *Vandermonde matrix*.

Disadvantage of the Lagrange Interpolant

The degree of $P_n(x)$ is fixed by the number of data points. To overcome this we could use piecewise polynomial interpolation, i.e., apply polynomial interpolation to subintervals of $[a, b]$, e.g., piecewise linear interpolation. This approach is more flexible as the case of an evenly spaced grid and ordinary polynomial interpolation may fail, while the appropriate use of piecewise polynomial interpolation is guaranteed to converge. Details are outside the scope of this course.

Existence and Uniqueness of $P_n(x)$

We saw a second method above for calculating a polynomial interpolant. Is such a polynomial unique? Can it always be found?

Theorem 16 (Existence/Uniqueness of $P_n(x)$)
Assume the x_i are distinct (i.e., if $i \neq j$ then $x_i \neq x_j$). Then there exists a unique polynomial of degree $\leq n$ satisfying

$$P_n(x_i) = y_i, \text{ for } i = 0, 1, ..., n$$

Theorem 17 (Error estimate for $P_n(x)$)
Let f have $n+1$ continuous derivatives on $[x_0, x_n]$. Then given any $x \in [x_0, x_n]$ there exists a number ξ_x (depending on x) in (x_0, x_n) s.t.

$$f(x) - P_n(x) = \prod_{i=0}^{n}(x - x_i)\frac{f^{(n+1)}(\xi_x)}{(n+1)!}.$$

Comments

- The proof of the above theorems are standard in any traditional numerical analysis course, but are not covered here.
- If f is a polynomial of degree $\leq n$, then $f^{(n+1)} = 0$, i.e., the approximation is exact ($P_n(x) = f(x)$).
- We bound the absolute error in the approximation in the usual way, via

  $$|f(x) - P_n(x)| \leq |(x - x_0)(x - x_1)...(x - x_n)| \cdot \frac{M}{(n+1)!}$$

  where M is an upper bound for $|f^{(n+1)}(x)|$ on (x_0, x_n).

Connection with the linear interpolation formula

With $n = 1$ (nodes x_0, x_1), the formula becomes:

$$f(x) - P_1(x) = (x - x_0)(x - x_1)\frac{f''(\xi_x)}{2!}, \ \xi_x \in (x_0, x_1).$$

Thus

$$|f(x) - P_1(x)| \leq |(x - x_0)(x - x_1)|\frac{M}{2}, \quad ----- \quad (*)$$

$$\text{where } M = \max |f''(x)|, \ x \in (x_0, x_1).$$

How do we maximize $|(x - x_0)(x - x_1)|$ for all $x \in (x_0, x_1)$? Consider the graph of $g(x) = |(x - x_0)(x - x_1)|$:

The maximum value occurs

at $x = \dfrac{x_0 + x_1}{2}$.

Figure 4.2: Graph of $g(x)$

Thus

$$g\left(\frac{x_0 + x_1}{2}\right) = \left|\left(\frac{x_0 + x_1}{2} - x_0\right) \cdot \left(\frac{x_0 + x_1}{2} - x_1\right)\right|$$
$$= \frac{(x_1 - x_0)^2}{4}. \quad \text{(Exercise)}$$

Thus $(*)$ becomes

$$|f(x) - P_n(x)| \leq \frac{(x_1 - x_0)^2}{4}\frac{M}{2} = (x - x_0)^2\frac{M}{8},$$

which is the formula we had in the Linear Interpolation case.

The Quadratic Case (with equally spaced nodes)

This is harder. Now $n = 2$ (nodes x_0, x_1 and x_2), so the error formula becomes

$$f(x) - P_2(x) = (x - x_0)(x - x_1)(x - x_2)\frac{f'''(\xi_x)}{3!} \text{ where } \xi_x \in (x_0, x_2).$$

Thus

$$|f(x) - P_2(x)| \leq |(x - x_0)(x - x_1)(x - x_2)|\frac{M}{6}, \quad ------ \quad (**)$$

where $M = \max |f'''(x)|$, for all $x \in (x_0, x_2)$.

Set $g(x) = |(x - x_0)(x - x_1)(x - x_2)|$. We want to maximize $g(x)$, but can't use symmetry here. As the nodes are equally spaced, set

$$h := x_1 - x_0 = x_2 - x_1.$$

Observe that:

$$\begin{aligned}(x-x_0)(x-x_1)(x-x_2) &= \big\{(x - x_1)+(x_1 - x_0)\big\}(x - x_1)\big\{(x - x_1)+(x_1 - x_2)\big\} \\ &= \big\{(x - x_1) + h\big\}(x - x_1)\big\{(x - x_1) - h\big\} \\ &= (x - x_1)\big\{(x - x_1) + h\big\}\big\{(x - x_1) - h\big\} \\ &= (x - x_1)\big\{(x - x_1)^2 - h^2\big\} \quad \text{('Difference of 2 squares')} \\ &= t(t^2 - h^2),\end{aligned}$$

where $t =: x - x_1$.

Now

$$\begin{aligned} & x_0 \leq x \leq x_2 \\ \Longrightarrow \quad & x_0 - x_1 \leq x - x_1 \leq x_2 - x_1 \\ \text{or} \quad & -h \leq t \leq h.\end{aligned}$$

So we wish to maximize $g(t) = |t(t^2 - h^2)|$ for $-h \leq t \leq h$.

Consider $G(t) = t(t^2 - h^2) = t^3 - th^2$ with roots $t = 0,\ t = \pm h$. Finding the critical values:

$$G'(t) = 3t^2 - h^2 = 0$$
$$\implies t = \pm\frac{h}{\sqrt{3}}.$$

So,

$$G\left(\pm\frac{h}{\sqrt{3}}\right) = \pm\frac{h}{\sqrt{3}}\left(\frac{h^2}{3} - h^2\right)$$
$$= \mp\frac{2h^3}{3\sqrt{3}}.$$

So

$$\boxed{g(t) = |G(t)| \leq \frac{2h^3}{3\sqrt{3}}.}$$

$- - - - - - - - -$ $(\star\star)$

Thus $(**)$ becomes

$$|f(x) - P_2(x)| \leq \frac{2h^3}{3\sqrt{3}} \cdot \frac{M}{6} = \frac{h^3}{9\sqrt{3}} \cdot M$$

i.e., the formula for the quadratic case (with equally spaced nodes) is:

$$\boxed{|f(x) - P_2(x)| \leq \frac{h^3}{9\sqrt{3}}M, \text{ where } M = \max_{x\in(x_0,x_2)} |f'''(x)|.}$$

(Derive from error estimate and $(\star\star)$).

Example 60 (Typical Exam Question)
What is the error in quadratic interpolation to $f(x) = e^{-x}$ using equally spaced nodes on the interval $[-1, 1]$? (Hint: $|(x - x_0)(x - x_1)(x - x_2)| \leq \frac{2h^3}{3\sqrt{3}}$).

Recall the error estimate for quadratic interpolation:

$$f(x) - P_2(x) = (x - x_0)(x - x_1)(x - x_2)\cdot\frac{f'''(\xi_x)}{3!}, \text{ where } \xi_x \in (x_0, x_2). \quad --- \quad (\#)$$

Now $n = 2$, so we have $h = \frac{b-a}{n} = \frac{1-(-1)}{2} = 1$, thus $x_0 = -1,\ x_1 = 0,\ x_2 = 1$. From $(\#)$ and the 'hint' we have

$$\begin{aligned} |f(x) - P_2(x)| &= |(x - x_0)(x - x_1)(x - x_2)| \cdot \frac{|f'''(\xi_x)|}{3!} \\ &\leq \frac{2h^3}{3\sqrt{3}} \cdot \frac{M}{6} \\ &= \frac{h^3 \cdot M}{9\sqrt{3}}, \quad - - - - - \quad (\#\#) \end{aligned}$$

where $M = \max_{x \in (x_0, x_2)} |f'''(x)|$.

Now,

$$\begin{aligned} f(x) = e^{-x} &\implies f'(x) = -e^{-x} \\ &\implies f''(x) = e^{-x} \\ &\implies f'''(x) = -e^{-x} \end{aligned}$$

$$\begin{aligned} \text{So } |f'''(x)| &= |-e^{-x}| \\ &= e^{-x} \leq e^{-(-1)} = e = M, \end{aligned}$$

since e^{-x} is decreasing on $[-1, 1]$.

Figure 4.3: Graph of e^{-1}.

Thus from $(\#\ \#)$ $\quad |e^{-x} - P_2(x)| \leq \frac{1^3 \cdot e}{9\sqrt{3}} = 0.174$ (3 s.f.).

4.2 Hermite Interpolation

We generalize Lagrange Interpolation in order to get an even better 'fit' between the interpolant $P(x)$ and a given function $f(x)$. We match function values *and* derivatives at the nodes, i.e.

$$P(x_i) = f(x_i), \quad P'(x_i) = f'(x_i), \quad 1 \leq i \leq n \quad ---- \quad (*)$$

Note: Nodes are indexed from $1 \longrightarrow n$ instead of $0 \longrightarrow n$.

The Method

Start be recalling the Lagrange Basis Functions:

$$L_k(x) = \prod_{\substack{j=1 \\ j \neq k}}^{n} \left(\frac{x - x_j}{x_k - x_j} \right), \quad k = 1, ..., n \quad \underbrace{(\text{degree } n-1)}_{\textit{not } \text{degree n}}$$

We use these to define the so called *Hermite Basis Functions*:

$$\left.\begin{aligned} h_k(x) &= [1 - 2(x - x_k)L_k'(x_k)]L_k^2(x) \\ \tilde{h}_k(x) &= (x - x_k)L_k^2(x) \end{aligned}\right\} \quad k = 1, ..., n \quad --- \quad (**)$$

Note that these basis functions have degree $2n-1$ (why?). These functions are constructed so that

$$\left.\begin{aligned} h_k(x_i) &= \delta_{ki}, \quad h_k'(x_i) = 0 \\ \tilde{h}_k(x_i) &= 0, \quad \tilde{h}_k'(x_i) = \delta_{ki} \end{aligned}\right\}, \quad --- \quad (***)$$

where

$$\delta_{ki} = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases},$$

is the Kronecker Delta function. It is a tedious exercise to verify $(***)$!
Now define the *Hermite Interpolant* with degree $2n - 1$:

$$H(x) = \sum_{k=1}^{n} (f(x_k)h_k(x) + f'(x_k)\tilde{h}_k(x)). \quad --- \quad (****)$$

Observe that after taking $\boldsymbol{H(x) = P(x)}$ the conditions $(*)$ are satisfied (after using $(***)$):

$$\begin{aligned} \boldsymbol{H(x_i)} &= \sum_{k=1}^{n}(f(x_k)\cdot h_k(x_i) + f'(x_k)\cdot \tilde{h}_k(x_i)) \\ &= \sum_{k=1}^{n}(f(x_k)\cdot \delta_{ki} + f'(x_k)\cdot 0) \\ &= f(x_i)\cdot 1 = \boldsymbol{f(x_i).} \quad \checkmark \end{aligned}$$

$$\begin{aligned} \boldsymbol{H'(x)} &= \frac{d}{dx}\sum_{k=1}^{n}(f(x_k)\cdot h_k(x) + f'(x_k)\cdot \tilde{h}_k(x)) \\ &= \sum_{k=1}^{n}(f(x_k)\cdot \underbrace{\frac{d}{dx}(h_k(x))}_{h_k'(x)} + f'(x_k)\cdot \underbrace{\frac{d}{dx}(\tilde{h}_k(x))}_{\tilde{h}_k'(x)}). \end{aligned}$$

Thus

$$\begin{aligned} \boldsymbol{H'(x_i)} &= \sum_{k=1}^{n}(f(x_k)\cdot h_k'(x_i) + f'(x_k)\cdot \tilde{h}_k'(x_i)) \\ &= \sum_{k=1}^{n}(f(x_k)\cdot 0 + f'(x_k)\cdot \delta_{ki}) \\ &= f'(x_i)\cdot 1 = \boldsymbol{f'(x_i).} \quad \checkmark \end{aligned}$$

taylor series everything
all numerical methods except hermite - always given
error bound formule always given

Example 61

Construct the cubic Hermite Interpolant to $f(x) = e^x$ using the nodes $a = -1,\ b = 1$. The degree of $H(x)$ is $2n-1$, so $2n-1 = 3$ (cubic), so $n = 4/2 = 2$, i.e. we have two nodes $x_1 = -1,\ x_2 = 1$.

$$L_1(x) = \frac{x - x_2}{x_1 - x_2} = \frac{x-1}{-1-1} = -\frac{1}{2}(x-1) = \frac{1}{2}(1-x)$$
$$L_2(x) = \frac{x - x_1}{x_2 - x_1} = \frac{x-(-1)}{1-(-1)} = \frac{1}{2}(x+1)$$

Thus we also have:

$$L_1'(x) = -\frac{1}{2},$$
$$L_2'(x) = \frac{1}{2}.$$

parts of this could be structured as a mc question -degree of hermite interpolation

Thus, using $(**)$ we have

$$\begin{aligned} h_1(x) &= [1 - 2(x-x_1)\cdot L_1'(x_1)]\cdot L_1^2(x) \\ &= \left[1 - 2(x-(-1))\cdot\left(-\frac{1}{2}\right)\right]\cdot\frac{1}{4}(1-x)^2 \\ &= \frac{1}{4}(x+2)(1-x)^2. \quad ---------- \quad (1) \end{aligned}$$

$$\begin{aligned} h_2(x) &= [1 - 2(x-x_2)\cdot L_2'(x_2)]\cdot L_2^2(x) \\ &= \left[1 - 2(x-1)\cdot\left(\frac{1}{2}\right)\right]\cdot\frac{1}{4}(x+1)^2 \\ &= \frac{1}{4}(2-x)(x+1)^2. \quad ---------- \quad (2) \end{aligned}$$

$$\begin{aligned}\tilde{h}_1(x) &= (x - x_1) \cdot L_1^2(x) \\ &= (x - (-1)) \cdot \frac{1}{4}(1 - x)^2 \\ &= \frac{1}{4}(x + 1)(1 - x)^2. \quad - - - - - - - - - \quad (3)\end{aligned}$$

$$\begin{aligned}\tilde{h}_2(x) &= (x - x_2) \cdot L_2^2(x) \\ &= (x - 1) \cdot \frac{1}{4}(x + 1)^2 \\ &= \frac{1}{4}(x - 1)(x + 1)^2. \quad - - - - - - - - - \quad (4)\end{aligned}$$

Thus from $(****)$ and (1)-(4) we have

$$\begin{aligned}H(x) &= f(x_1) \cdot h_1(x) + f'(x_1) \cdot \tilde{h}_1(x) + f(x_2) \cdot h_2(x) + f'(x_2) \cdot \tilde{h}_2(x) \\ &= e^{-1}\frac{1}{4}(x + 2)(1 - x)^2 + e^{-1}\frac{1}{4}(x + 1)(1 - x)^2 \\ &\qquad + e^{1}\frac{1}{4}(2 - x)(x + 1)^2 + e^{1}\frac{1}{4}(x - 1)(x + 1)^2 \\ &= \text{ (after simplifying !)} \\ &= 0.184x^3 + 0.588x^2 + 0.991x + 0.955 \quad \text{(coefficients to 3 s.f.)}\end{aligned}$$

(See Figure 4.6 in your text for the graph of $H(x)$ and e^x on $[-1, 1]$).

Interpolation Error

The following result provides an error estimate for Hermite Interpolation.

Theorem 18
Let f have $2n$ continuous derivatives on $[x_1, x_n]$. Then given any $x \in (x_1, x_n)$ there exists a number ξ_x (depending on x) in (x_1, x_n) s.t.

$$f(x) - H(x) = \prod_{i=1}^{n}(x - x_i)^2 \cdot \frac{f^{(2n)}(\xi_x)}{(2n)!}$$

Note: $\prod_{i=1}^{n}(x - x_i)^2 = (x - x_1)^2(x - x_2)^2 \dots (x - x_n)^2$.

Comments

- compare the error estimate for Lagrange Interpolation
- if f is a polynomial of degree $\leq 2n - 1$, then $f^{(2n)} = 0$, i.e., the approximation is exact ($H(x) = f(x)$).
- we bound the absolute error in the approximation via

$$|f(x) - H(x)| \leq \left|\prod_{i=1}^{n}(x - x_i)^2\right| \cdot \frac{M}{(2n)!}$$

where M is an upper bound for $|f^{(2n)}|$ on (x_1, x_n).

Example 62
What is the error in the cubic Hermite Interpolation to $f(x) = \dfrac{1}{x}$ on $[1, 2]$?

Degree of $H(x) = 2n - 1 = 3$. Thus $n = 2$, so we take nodes $x_1 = 1,\ x_2 = 2$. Recall

$$f(x) - H(x) = (x - x_1)^2(x - x_2)^2 \cdot \frac{f^{(4)}(\xi_x)}{4!},$$
$$\text{so } |f(x) - H(x)| \leq |(x-1)^2(x-2)^2| \cdot \frac{M}{24}, \quad ----- \quad (1)$$

where M is an upper bound for $|f^{(4)}(x)|$ on $[1, 2]$.

$$\begin{aligned} f(x) = x^{-1} &\implies f'(x) = -x^{-2} \\ &\implies f''(x) = 2x^{-3} \\ &\implies f'''(x) = -6x^{-4} \\ &\implies f^{(4)}(x) = 24x^{-5}, \end{aligned}$$

which is a monotonic decreasing function on $[1, 2]$, so

$$|f^{(4)}(x)| \leq 24 \cdot |1|^{-5} = 24 = M. \quad ----- \quad (2)$$

Now we need to maximize $|(x-1)^2(x-2)^2|$ on $[1, 2]$.

The basic shape of $h(x) = (x-1)^2(x-2)^2$ is

Figure 4.4: Graph of $h(x)$.

so there are double roots at $x = 1$ and $x = 2$.

We need to find the local max on $[1, 2]$:

$$\begin{aligned} h'(x) &= (x-1)^2 \cdot 2(x-2) + (x-2)^2 \cdot 2(x-1) \\ &= 2(x-1)(x-2) \cdot [(x-1) + (x-2)] \\ &= 2(x-1)(x-2)(2x-3) \\ &\implies \text{local extrema occur at } x = 1, 2, 3/2. \end{aligned}$$

so $x = 3/2$ gives the local max.
Thus

$$\begin{aligned} |h(x)| &\leq |h(3/2)| \\ &= |(3/2-1)^2(3/2-2)^2| \\ &= \frac{1}{16}. \quad --- \quad (3) \end{aligned}$$

Thus using (2) and (3) in (1):

$$\left| \frac{1}{x} - H(x) \right| \leq \frac{1}{16} \cdot \frac{24}{24} = \frac{1}{16} = 0.0625.$$