Numerical Differentiation

We assume that we can compute a function \( f \), but that we have no information about how to compute \( f' \). We want ways of estimating \( f'(x) \), given what we know about \( f \).

**Reminder: definition of differentiation:**

\[
\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}
\]

For second derivatives, we have the definition:

\[
\frac{d^2f}{dx^2} = \lim_{\Delta x \to 0} \frac{f(x + 2\Delta x) - 2f(x + \Delta x) + f(x)}{\Delta x^2}
\]

Error Estimation in Differentiation II

We don’t know the value of either \( f'' \) or \( \xi \), but we can say that the error is order \( h \):

\[
R_T \text{ for } D_x(h) \text{ is } O(h)
\]

so the error is proportional to the step size — as one might naively expect.

For \( D_- (h) \) we get a similar result for the truncation error — also \( O(h) \).

First Derivative

We can use this formula, by taking \( \Delta x \) equal to some small value \( h \), to get the following approximation,

- known as the Forward Difference \( (D_x(h)) \):

\[
f'(x) = D_x(h) = \frac{f(x + h) - f(x)}{h}
\]

- Alternatively we could use the interval on the other side of \( x \), to get the Backward Difference \( (D_- (h)) \):

\[
f'(x) = D_- (h) = \frac{f(x) - f(x - h)}{h}
\]

- A more symmetric form, the Central Difference \( (D_0 (h)) \), uses intervals on either side of \( x \):

\[
f'(x) \approx D_0 (h) = \frac{f(x + h) - f(x - h)}{2h}
\]

All of these give (different) approximations to \( f'(x) \).

Second Derivative

The simplest way is to get a symmetrical equation about \( x \) by using both the forward and backward differences to estimate \( f''(x + \Delta x) \) and \( f''(x) \) respectively:

\[
f''(x) \approx \frac{D_+(h) - D_-(h)}{h} = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}
\]

Error Estimation in Differentiation I

We shall see that the error involved in using these differences is a form of truncation error \( (R_T) \):

\[
R_T = D_x(h) - f'(x)
\]

Using Taylor’s Theorem:

\[
f(x + h) = f(x) + f'(x)h + f''(x)h^2/2! + f'''(x)h^3/3! + \cdots
\]

\[
R_T = \frac{1}{2} f''(x)h + \frac{f''(x)h^2}{2!} + \frac{f'''(x)h^3}{3!} + \cdots - f'(x)
\]

\[
= \frac{1}{2} f''(x)h + \frac{1}{2} f'''(x)h^2/2! + f'''(x)h^3/3! + \cdots)
\]

Using the Mean Value Theorem, for some \( \xi \) within \( h \) of \( x \):

\[
R_T = \frac{f''(\xi)h}{2}
\]

Exercise: differentiation I

Limit of the Difference Quotient. Consider the function \( f(x) = e^x \).

- compute \( f'(1) \) using the sequence of approximation for the derivative:

\[
D_{x^k} = \frac{f(x + (x^k) - f(x)}{x^k}
\]

with \( x_k = 10^{-k}, \ k \geq 1 \)

- for which value \( k \) do you have the best precision (knowing \( e^1 = 2.71828182845905 \)), Why?
Exercise: differentiation II

We evaluate

Let try again the example:

Example.

Numerical values: xls/Lect13.xls

Central Difference

- we have looked at approximating \( f'(x) \) with the backward \( D_{-1}(h) \) and forward difference \( D_{+1}(h) \).
- Now we just check out the approximation with the central difference:

\[
 f'(x) \approx D_0(h) = \frac{f(x + h) - f(x - h)}{2h}
\]

- Richardson extrapolation

Error analysis of Central Difference I

We consider the error in the central difference estimate \( D_0(h) \) of \( f'(x) \):

\[
 D_0(h) = \frac{f(x + h) - f(x - h)}{2h}
\]

We apply Taylor’s Theorem,

\[
 f(x + h) = f(x) + f'(x)h + \frac{f''(x)h^2}{2!} + \frac{f'''(x)h^3}{3!} + \frac{f^{(4)}(x)h^4}{4!} + \cdots \tag{A}
\]

\[
 f(x - h) = f(x) - f'(x)h + \frac{f''(x)h^2}{2!} - \frac{f'''(x)h^3}{3!} + \frac{f^{(4)}(x)h^4}{4!} + \cdots \tag{B}
\]

\[
 \frac{(A) - (B)}{2h} = f'(x) + \frac{f''(x)h^2}{3!} + \frac{f^{(3)}(x)h^3}{5!} + \cdots
\]

Error analysis of Central Difference II

We see that the difference can be written as

\[
 D_0(h) = f'(x) + f''(x)xh^2 + \frac{f^{(4)}(x)}{24} + \cdots
\]

or alternatively, as

\[
 D_0(h) = f'(x) + b_1 h^2 + b_2 h^4 + \cdots
\]

where we know how to compute \( b_1, b_2, \) etc.

We see that the error \( R_T = D_0(h) - f'(x) \) is \( O(h^2) \).

Remark. Remember: for \( D_{-1} \) and \( D_{+1} \), the error is \( O(h) \).

Error analysis of Central Difference III

Example.

Let try again the example:

\[
 f(x) = e^x \quad f'(x) = e^x
\]

We evaluate \( f'(1) = e^1 \approx 2.71828 \ldots \) with

\[
 D_0(h) = \frac{f(1 + h) - f(1 - h)}{2h}
\]

for \( h = 10^{-k}, \ k \geq 1 \).

Numerical values: xls/Lect13.xls

Rounding Error in Difference Equations I

- When presenting the iterative techniques for root-finding, we ignored rounding errors, and paid no attention to the potential error problems with performing subtraction. This did not matter for such techniques because:
  - the techniques are self-correcting, and tend to cancel out the accumulation of rounding errors
  - the iterative equation \( x_{n+1} = \xi_n - \kappa_n x_n \) where \( \kappa_n \) is some form of correction factor has a subtraction which is safe because we are subtracting a small quantity \( \kappa_n \) from a large one (e.g. for Newton-Raphson, \( \kappa_n = \frac{f(x)}{f'(x)} \)).
Rounding Error in Difference Equations II

- However, when using a difference equation like
  \[ D_0(h) = \frac{f(x+h) - f(x-h)}{2h} \]
  we seek a situation where \( h \) is small compared to everything else, in order to get a good approximation to the derivative. This means that \( x + h \) and \( x - h \) are very similar in magnitude, and this means that for most \( f \) (well-behaved) that \( f(x + h) \) will be very close to \( f(x - h) \). So we have the worst possible case for subtraction: the difference between two large quantities whose values are very similar.

- We cannot re-arrange the equation to get rid of the subtraction, as this difference is inherent in what it means to compute an approximation to a derivative (differentiation uses the concept of difference in a deeply intrinsic way).

Rounding Error in Difference Equations III

- We see now that the total error in using \( D_0(h) \) to estimate \( f'(x) \) has two components:
  - the truncation error \( R_T \) which we have already calculated,
  - and a function calculation error \( R_{\text{calc}} \) which we now examine.

- When calculating \( D_0(h) \), we are not using totally accurate computations of \( f \), but instead we actually compute an approximation \( \tilde{f} \) to get
  \[ D_0(h) = \frac{\tilde{f}(x + h) - \tilde{f}(x - h)}{2h} \]

- We shall assume that the error in computing \( f \) near to \( x \) is bounded in magnitude by \( c \):
  \[ |f(x) - \tilde{f}(x)| \leq c \]

Rounding Error in Difference Equations IV

- The calculation error is then given as
  \[ R_{\text{calc}} = D_0(h) - D_0(h) \]
  \[ = \frac{\tilde{f}(x + h) - \tilde{f}(x - h) - f(x + h) - f(x - h)}{2h} \]
  \[ = \frac{\tilde{f}(x + h) - f(x + h) + f(x - h) - \tilde{f}(x - h)}{2h} \]

- However, we have already calculated
  \[ |R_{\text{calc}}| \leq \frac{c^2}{2h} \]
  \[ \leq \frac{c}{h} \]

- So we see that \( R_{\text{calc}} \) is proportional to \( 1/h \), so as \( h \) shrinks, this error grows, unlike \( R_T \) which shrinks quadratically as \( h \) does.

Rounding Error in Difference Equations V

- We see that the total error \( R \) is bounded by \( |R_T| + |R_{\text{calc}}| \), which expands out to
  \[ |R| \leq \frac{f'''(\xi)}{6} h^2 + c \]

So we see that to minimise the overall error we need to find the value of \( h = b_{\text{opt}} \) which minimises the following expression:
\[ \frac{f'''(\xi)}{6} h^2 + c \]

Unfortunately, we do not know \( f''' \) or \( \xi \)!

Many techniques exist to get a good estimate of \( b_{\text{opt}} \), most of which estimate \( f''' \) numerically somehow. These are complex and not discussed here.

Richardson Extrapolation I

- The trick is to compute \( D_0(h) \) for 2 different values of \( h \), and combine the results in some appropriate manner, as guided by our knowledge of the error behaviour.

In this case we have already established that
\[ D_0(h) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + b_1 h^2 + O(h^4) \]

We now consider using twice the value of \( h \):
\[ D_0(2h) = \frac{f(x+2h) - f(x-2h)}{4h} = f'(x) + b_2 4h^2 + O(h^4) \]

We can subtract these to get:
\[ D_0(2h) - D_0(h) = 3b_2 h^2 + O(h^4) \]

We divide across by 3 to get:
\[ \frac{D_0(2h) - D_0(h)}{3} = b_2 h^2 + O(h^4) \]

Richardson Extrapolation II

- The righthand side of this equation is simply \( D_0(h) - f'(x) \), so we can substitute to get
  \[ \frac{D_0(2h) - D_0(h)}{3} = D_0(h) - f'(x) + O(h^4) \]

This re-arranges (carefully) to obtain
\[ f'(x) = D_0(h) + \frac{D_0(2h) - D_0(h)}{3} + O(h^4) \]

\[ = \frac{4D_0(h) - D_0(2h)}{3} + O(h^4) \]
Richardson Extrapolation III

- It is an estimate for \( f'(x) \) whose truncation error is \( O(h^4) \), and so is an improvement over \( D_0 \) used alone.
- This technique of using calculations with different \( h \) values to get a better estimate is known as Richardson Extrapolation.

Richardson’s Extrapolation. Suppose that we have the two approximations \( D_0(h) \) and \( D_0(2h) \) for \( f'(x) \), then an improved approximation has the form:

\[
f'(x) = \frac{4D_0(h) - D_0(2h)}{3} + O(h^4)
\]

Summary

- Approximation for numerical differentiation:

<table>
<thead>
<tr>
<th>Approximation for ( f'(x) )</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward/backward difference ( D_0, D )</td>
<td>( O(h) )</td>
</tr>
<tr>
<td>Central difference ( D_0 )</td>
<td>( O(h^2) )</td>
</tr>
<tr>
<td>Richardson Extrapolation</td>
<td>( O(h^4) )</td>
</tr>
</tbody>
</table>

- Considering the total error (approximation error + calculation error):

\[
|R| \leq \left| \frac{f''''(\xi)}{6!} \right| h^4 + \left| \frac{\epsilon}{2} \right|
\]

remember that \( h \) should not be chosen too small.

Solving Differential Equations Numerically

Definition.
The Initial value Problem deals with finding the solution \( y(x) \) of

\[
y' = f(x, y) \quad \text{with the initial condition} \quad y(x_0) = y_0
\]

- It is a 1st order differential equations (D.E.s).
- Alternative ways of writing \( y' = f(x, y) \) are:

\[
\begin{align*}
y'(x) &= f(x, y) \\
\frac{dy}{dx} &= f(x, y)
\end{align*}
\]

Working Example

- We shall take the following D.E. as an example:

\[
f(x, y) = y
\]

or \( y' = y \) (or \( y'(x) = y(x) \)).
- This has an infinite number of solutions:

\[
y(x) = C \cdot e^x \quad \forall C \in \mathbb{R}
\]
- We can single out one solution by supplying an initial condition \( y(x_0) = y_0 \).
- So, in our example, if we say that \( y(0) = 1 \), then we find that \( C = 1 \) and our solution is

\[
y = e^x
\]

The Lipschitz Condition I

We can give a condition that determines when the initial condition is sufficient to ensure a unique solution, known as the Lipschitz Condition.

Lipschitz Condition:
For \( a \leq x \leq b \), for all \( -\infty < y, y' < \infty \), if there is an \( L \) such that

\[
|f(x, y) - f(x, y')| \leq L |y - y'|
\]

Then the solution to \( y' = f(x, y) \) is unique, given an initial condition.
- \( L \) is often referred to as the Lipschitz Constant.
- A useful estimate for \( L \) is to take \( \frac{\| f \|}{\| y \|} < L \), for \( x \) in \( (a, b) \).
The Lipschitz Condition II

Example.

Given our example of \( y' = f(x,y) \), then we can see do we get a suitable \( L \).

\[
\frac{\partial f}{\partial y} = \frac{\partial (y)}{\partial (x)} = 1
\]

So we shall try \( L = 1 \)

\[
|f(x,y) - f(x',y')| = |y - y'|
\leq 1 \cdot |y - y'|
\]

So we see that we satisfy the Lipschitz Condition with a Constant \( L = 1 \).

Euler’s Method

- The technique works by using applying \( f \) at the current point \((x_n,y_n)\) to get an estimate of \( y' \) at that point.

This technique for solving D.E.'s is known as Euler’s Method.

- It is simple, slow and inaccurate, with experimentation showing that the error is \( O(h) \).

Truncation Errors I

Definitions.

- The error introduced at each step is called the Local Truncation Error.
- The error introduced at any given point, as a result of accumulating all the local truncation errors up to that point, is called the Global Truncation Error.

In the diagram above, the local truncation error is \( y(x_{n+1}) - y_{n+1} \).

Numerically solving \( y' = f(x,y) \)

- We assume we are trying to find values of \( y \) for \( x \) ranging over the interval \([a,b]\).
- We start with the one point where we have the exact answer, namely the initial condition \( y_0 = y(x_0) \).
- We generate a series of \( x \)-points from \( a = x_0 \) to \( b \), separated by a small step-interval \( h \):

\[
\begin{align*}
x_0 &= a \\
x_i &= a + i \cdot h \\
h &= \frac{b-a}{N} \\
x_N &= b
\end{align*}
\]

- We want to compute \( \{y_i\} \), the approximations to \( \{y(x_i)\} \), the true values.

Euler’s Method

Example.

In our example, we have

\[
y' = y \\
y(x,y) = y \\
y_{n+1} = y_n - h \cdot y_n
\]

At each point after \( x_0 \), we accumulate an error, because we are using the slope at \( x_n \) to estimate \( y_{n+1} \), which assumes that the slope doesn’t change over interval \([x_n, x_{n+1}]\).

Truncation Errors II

We can estimate the local truncation error \( y(x_{n+1}) - y_{n+1} \), by assuming the value \( y_n \) for \( x_n \) is exact as follows: as follows:

\[
y(x_{n+1}) = y(x_n) + h
\]

Using Taylor Expansion about \( x = x_n \)

\[
y(x_{n+1}) = y(x_n) + h y'(x_n) + \frac{h^2}{2} y''(\xi)
\]

Assuming \( y_0 \) is exact \( (y(x_n) = y(x_n)) \), so \( y'(x_n) = f(x_n,y_n) \)

\[
y(x_{n+1}) = y_n + h f(x_n,y_n) + \frac{h^2}{2} y''(\xi)
\]

Now looking at \( y_{n+1} \) by definition of the Euler method:

\[
y_{n+1} = y_n + h f(x_n,y_n)
\]

We subtract the two results:

\[
y(x_{n+1}) - y_{n+1} = \frac{h^2}{2} y''(\xi)
\]
Improved Differentiation Techniques II

- Using the slope \( y'(x_n, y_n) = f(x_n, y_n) \) at \( x_n \), the Euler approximation is:

\[
\frac{y_{n+1} - y_n}{h} \approx f(x_n, y_n)
\]

- Considering the slope \( y'(x_{n+1}, y_{n+1}) = f(x_{n+1}, y_{n+1}) \) at \( x_{n+1} \), we can propose this new approximation:

\[
\frac{y_{n+1} - y_n}{h} \approx f(x_{n+1}, y_{n+1})
\]

- The trouble is: we don’t know \( y_{n+1} \) in \( f \) (because this is what we are looking for!).
- So instead we use \( y_n^{(k)} \), the Euler’s approximation of \( y_{n+1} \):

\[
\frac{y_{n+1} - y_n}{h} \approx f(x_{n+1}, y_n^{(k)})
\]

Runge-Kutta Techniques I

- We can repeat the Heun’s approach by considering the approximations of slopes in the interval \([x_n, x_{n+1}]\).
- This leads to a large class of improved differentiation techniques which evaluate \( f \) many times at each \( h \)-step, in order to get better error performance.
- This class of techniques is referred to collectively as Runge-Kutta techniques, of which Heun’s Method is the simplest example.
- The classical Runge-Kutta technique evaluates \( f \) four times to get a method with global truncation error of \( O(h^4) \).
Runge-Kutta Techniques II

Runge-Kutta's technique using 4 approximations. It is computed using approximations of the slope at $x_n$, $x_{n+1}$ and also two approximations at mid interval $x_n + \frac{h}{2}$:

$$\frac{y_{n+1} - y_n}{h} = \frac{1}{6} (f_1 + 2f_2 + 2f_3 + f_4)$$

with

$$f_1 = f(x_n, y_n)$$
$$f_2 = f \left( x_n + \frac{h}{2}, y_n + \frac{h}{2} f_1 \right)$$
$$f_3 = f \left( x_n + \frac{h}{2}, y_n + \frac{h}{2} f_2 \right)$$
$$f_4 = f \left( x_n + h, y_n + h \cdot f_3 \right)$$

It can be shown that the global truncation error is $O(h^4)$.