Differentiation and Integration

You think you know when you learn, are more sure when you can write, even more when you can teach, but certain when you can program.

Alan J. Perlis

Fall 2010
This unit requires the knowledge of differentiation and integration in calculus.

The following topics will be presented:

- Forward difference, backward difference and central difference methods for differentiation.
- Richardson extrapolation technique
- Trapezoid, Simpson’s 3-point and iterative methods for numerical integration
- Romberg’s method for numerical integration.
If function \( f(x) \) is complex, computing \( f'(x) \) is difficult and approximation may be needed.

Suppose we wish to compute \( f'(x_i) \). We may choose a \( x_{i+1} > x_i \) and approximate \( f'(x_i) \) by the slope \( (f(x_{i+1}) - f(x_i)) / (x_{i+1} - x_i) \) — forward difference.
Numerical Differentiation:

• Or, we may choose a \( x_{i-1} < x_i \) and use the slope 
  \[ \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \]
  as an approximation of \( f'(x_i) \)
  — **backward difference.**

The diagram illustrates the backward difference formula with a graph showing the function \( f(x) \) and the derivative \( f'(x_i) \) at point \( x_i \). The slope between the points \( x_{i-1} \) and \( x_i \) approximates the derivative at \( x_i \).
Or, we may choose \( x_{i-1} < x_i \) and \( x_{i+1} > x_i \), both close to \( x_i \), and use slope \( (f(x_{i+1})-f(x_{i-1}))/\left(x_{i+1} - x_{i-1}\right) \) as an approximation of \( f'(x_i) \) — central difference.
Numerical Differentiation: 4/14

- Forward difference, backward difference and central difference methods can be viewed as the use of an *interpolating polynomial of degree 1* that interpolates two chosen points and uses its slope for the derivative of \( f(x) \).
- If we can use interpolating polynomial of degree 1, *why don’t we use degree 2, degree 3, etc?*
- Yes, of course we can. But, higher degree means more points. Normally, three (*i.e.*, degree 2) may be sufficient.
Suppose we wish to compute $f'(x_i)$. We may choose $x_{i+1}$ and $x_{i+2}$, both close to $x_i$, and evaluate $f_i$, $f_{i+1}$ and $f_{i+2}$.

Then, we have three points $(x_i, f_i)$, $(x_{i+1}, f_{i+1})$ and $(x_{i+2}, f_{i+2})$.

An interpolating polynomial of degree 2, $P_2(x)$, can be found, and, use $P_2'(x_i)$ as an approximation of $f'(x_i)$. See next slide.

Newton divided difference is a simple tool for this purpose.
Numerical Differentiation: \( P_2(x) \) and \( f(x) \)

\[ f'(x) + f''(x) + f'''(x) \]

\[ f_{i+1} \quad f_{i+2} \]

\[ x_i \quad x_{i+1} \quad x_{i+2} \]
Numerical Differentiation: 7/14

- Given $x_i < x_{i+1} < x_{i+2}$, divided difference yields $P_2(x)$ as follows:

$$P_2(x) = f[x_i] + f[x_i, x_{i+1}](x - x_i) + f[x_i, x_{i+1}, x_{i+2}](x - x_i)(x - x_{i+1})$$

- Differentiating $P_2(x)$ yields:

$$P_2'(x) = f[x_i, x_{i+1}] + f[x_i, x_{i+1}, x_{i+2}](2x - (x_i + x_{i+1}))$$

- Plugging $x_i$ into $P_2'(x)$ as an approximation of $f'(x_i)$ yields:

$$P_2'(x_i) = f[x_i, x_{i+1}] + f[x_i, x_{i+1}, x_{i+2}](x_i - x_{i+1})$$
Numerical Differentiation: 8/14

• Normally, \( x_i, x_{i+1} \) and \( x_{i+2} \) are \textit{equally spaced} (i.e., \( x_{i+1} - x_i = x_{i+2} - x_{i+1} = \Delta \)). This condition makes computation easier.

• Note that \( f[x_i, x_{i+1}] \) and \( f[x_{i+1}, x_{i+2}] \) are computed as follows:

\[
f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{\Delta} \quad f[x_{i+1}, x_{i+2}] = \frac{f(x_{i+2}) - f(x_{i+1})}{\Delta}
\]

• As a result, \( f[x_i, x_{i+1}, x_{i+2}] \) is

\[
f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} = \frac{f_{i+2} - 2f_{i+1} + f_i}{2\Delta^2}
\]
Numerical Differentiation: 9/14

- Recall that $P_2'(x)$ is

$$P_2'(x_i) = f[x_i, x_{i+1}] + f[x_i, x_{i+1}, x_{i+2}](x_i - x_{i+1})$$

- Plugging $f[x_i, x_{i+1}], f[x_i, x_{i+1}, x_{i+2}]$ and $x_i - x_{i+1} = \Delta$ into $P_2'(x)$, we have

$$P_2'(x_i) = \frac{1}{2\Delta}(-3f_i + 4f_{i+1} - f_{i+2})$$

- This is the three-point forward difference.
If the three chosen points are $x_{i-2} < x_{i-1} < x_i$, the interpolating polynomial of degree 2, $P_2(x)$, is

$$P_2(x) = f[x_{i-2}] + f[x_{i-2}, x_{i-1}](x - x_{i-2}) + f[x_{i-2}, x_{i-1}, x_i](x - x_{i-2})(x - x_{i-1})$$

Differentiating $P_2(x)$ yields:

$$P_2'(x) = f[x_{i-2}, x_{i-1}] + f[x_{i-2}, x_{i-1}, x_i](2x - (x_{i-2} + x_{i-1}))$$

Plugging $x_i$ into $P_2'(x)$ yields

$$P_2'(x_i) = f[x_{i-2}, x_{i-1}] + f[x_{i-2}, x_{i-1}, x_i](2x_i - (x_{i-2} + x_{i-1}))$$
If $x_{i-2}$, $x_{i-1}$ and $x_i$ are equally spaced (i.e., $x_{i-1} - x_{i-2} = \Delta$ and $x_i - x_{i-1} = \Delta$), we have

$$2x_i - (x_{i-2} + x_{i-1}) = (x_i - x_{i-1}) + (x_i - x_{i-2}) = \Delta + 2\Delta = 3\Delta$$

$P_2'(x)$ becomes:

$$P_2'(x_i) = \frac{1}{2\Delta}(f_{i-2} - 4f_{i-1} + 3f_i)$$

This is the three-point backward difference.
Numerical Differentiation: 12/14

- Forward difference uses $x_i < x_{i+1} < x_{i+2}$, and backward difference uses $x_{i-2} < x_{i-1} < x_i$. Is there a “central” difference that uses $x_{i-1} < x_i < x_{i+1}$?

- You certainly can do this because the $P_2(x)$ is:

  $$P_2(x) = f[x_{i-1}] + f[x_{i-1}, x_i](x - x_{i-1}) + f[x_{i-1}, x_i, x_{i+1}](x - x_{i-1})(x - x_i)$$

- Some calculations yield the following (exercise):

  $$P'_2(x_i) = \frac{f_{i+1} - f_{i-1}}{2\Delta}$$

- This does not involve $f_i$ and is the same as the 2-point central difference.
Exercises:

- If the equally spaced data points are $x_{i-1} < x_i < x_{i+1}$, show that $P_2'(x_i)$ is the following:
  \[ P_2'(x_i) = \frac{f_{i+1} - f_{i-1}}{2\Delta} \]

- Given a parabola $f(x) = ax^2 + bx + c$ and three points $s-\Delta$, $s$ and $s+\Delta$, show that
  \[ f'(s) = \frac{f(s + \Delta) - f(s - \Delta)}{2\Delta} \]

Now we know that the three-point central difference does not make much sense for equally spaced data points.
### Numerical Differentiation: 14/14

**Examples:**

- \( f(x) = e^x, \ x = 1 \) and \( \Delta = 0.001 \)
- \( f'(x) = e^x \) and \( f'(1) = e^1 = 2.718281828 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Difference</td>
<td>2.71964142</td>
</tr>
<tr>
<td>Backward Difference</td>
<td>2.71962268</td>
</tr>
<tr>
<td>Central Difference</td>
<td>2.71828228</td>
</tr>
<tr>
<td>3-point Forward</td>
<td>2.71828091</td>
</tr>
<tr>
<td>3-point Backward</td>
<td>2.71828091</td>
</tr>
</tbody>
</table>
Richardson Extrapolation: 1/4

- Richardson has an **extrapolation** scheme that can make the central different method more accurate.

- This extrapolation scheme is very similar to the divided difference scheme:

\[
\begin{align*}
\frac{d_{i,j} - d_{i-1,j}}{4^{j+1}} - 1
\end{align*}
\]
Richardson Extrapolation: 2/4

Richardson extrapolation is computed row-by-row. The 0-th entry is a central difference with reduced step size.

\[ d_{i,j+1} = d_{i,j} + \frac{d_{i,j} - d_{i-1,j}}{4^{j+1} - 1} \]
Richardson Extrapolation: 3/4

- Here is the algorithm of Richardson’s method.
- \(x\): input, \(n\): number of rows, and \(\Delta\): initial step size that will be halved for each row.

\[
\text{DO } i = 0, n \\
\quad d_{i,0} = \frac{f(x + \Delta) - f(x - \Delta)}{2\Delta} \\
\quad \text{DO } j = 0, i-1 \\
\quad \quad d_{i,j+1} = d_{i,j} + \frac{d_{i,j} - d_{i-1,j}}{4^{j+1} - 1} \\
\quad \text{END DO} \\
\Delta = \Delta / 2 \\
\text{END DO}
\]
Richardson Extrapolation: 4/4

Let $f(x) = e^x$. Compute $f'(x)$ at $x=1$ with $n=3$ (3 rows) and $\Delta=1$ (initial step size=1).

$$d_{i,j+1} = d_{i,j} + \frac{d_{i,j} - d_{i-1,j}}{4^{j+1} - 1}$$

row 0 ($\Delta=1$) 3.194528

row 1 ($\Delta=.5$) 2.8329677 $\rightarrow$ 2.7124476

row 2 ($\Delta=.25$) 2.7466855 $\rightarrow$ 2.7179248 $\rightarrow$ 2.71829

row 3 ($\Delta=.125$) 2.7253665 $\rightarrow$ 2.7182602 $\rightarrow$ 2.7182826 $\rightarrow$ 2.7182826

col 0 col 1 col 2 col 3
Numerical Integration: 1/4

- Numerical integration means computing the following in a numerical way (i.e., a value rather than a closed form formula).

\[ \int_{a}^{b} f(x)dx \]

- In fact, the above cannot be integrated precisely in closed form for most functions \( f(x) \). As a result, numerical integration is needed.
One way to compute the integration is to (1) choose a number of data points \((x_0=a, f_0), (x_1, f_1), \ldots, (x_n=b, f_n)\) and (2) find an interpolating polynomial of degree \(n, P_n(x)\).

Then, we use \(P_n(x)\) to replace \(f(x)\):

\[
\int_a^b f(x)dx \approx \int_a^b P_n(x)dx
\]

Since \(P_n(x)\) is a polynomial, its integration is easy to compute. But, this is a tedious and inefficient procedure.
Numerical Integration: 3/4

- In fact, one may divide the interval \([a,b]\) into subintervals for better approximation instead of using the whole interval \([a,b]\).
- If the subintervals are small enough, degree 1 or 2 interpolating polynomials may be good enough.

\[
\int_{a=x_0}^{b=x_4} f(x) \, dx = \sum_{i=0}^{3} \int_{x_i}^{x_{i+1}} f(x) \, dx
\]
While the subintervals do not have to be of equal length, equally spaced points do make computation easier.

Therefore, if we choose to divide the interval $[a,b]$ into $n$ subintervals, each of which has length $\Delta=(b-a)/n$, the division points are $x_0 = a$, $x_1 = x_0+\Delta$, $x_2 = x_0+2\Delta$ ..., $x_i = x_0+i\Delta$, .., $x_n = x_0+n\Delta=b$.

The integration becomes:

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_i+\Delta} f(x)dx$$
The trapezoid bounded by \((a,0), (a,f(a)), (b,f(b))\) and \((b,0)\) has an area close to the area below \(y=f(x)\) bounded by \(x=a\) and \(x=b\).

\[
\text{area} = \frac{b-a}{2} (f(a) + f(b))
\]
Trapezoid Rule: \(2/5\)

- If \([a,b]\) is divided into \(n\) equally spaced intervals with length \(\Delta=(b-a)/n\). Then, \(x_i = x_0 + i\Delta\).
- The area \(A_i\) of the \(i\)-th (\(0 \leq i \leq n-1\)) trapezoid is \((f_i + f_{i+1})\Delta/2\).
- The approximation is the sum of all \(A_i\)'s.
Therefore, the sum of all areas $A_0, A_1, \ldots, A_{n-1}$ as an approximation of the integration is easy to compute.

\[
\int_a^b f(x) \, dx \approx A_0 + A_1 + \cdots + A_{n-1} \\
= \frac{\Delta}{2}(f_0 + f_1) + \frac{\Delta}{2}(f_1 + f_2) + \cdots + \frac{\Delta}{2}(f_{n-1} + f_n) \\
= \frac{\Delta}{2} \left[ f_0 + 2(f_1 + f_2 + \cdots + f_{n-1}) + f_n \right] \\
= \Delta \left[ \frac{f_0 + f_n}{2} + \sum_{i=1}^{n-1} f_i \right]
\]
Recall the following trapezoid rule:

\[ \int_a^b f(x) \, dx \approx \Delta \left[ \frac{f_0 + f_n}{2} + \sum_{i=1}^{n-1} f_i \right] \]

The following is a possible implementation:

```plaintext
! INPUT: a, b, n
Δ = (b-a)/n    ! step size
x = a + Δ     ! x1 here
s = 0.0      ! sum of f(x_1) to f(x_{n-1})
DO i = 1, n-1 ! cumulate each term
    s = s + f(x)
    x = x + Δ
END DO
Result = ((f(a)+f(b))/2+s)*Δ
```
Consider the integration of $e^x$ from 0 to 1. The correct result is $e^1 - e^0 = 1.718282$.

If $[0,1]$ is divided into 4 subintervals, we have $n=4$ and $\Delta=0.25$.

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>EXP $(x_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.25</td>
<td>1.284025</td>
</tr>
<tr>
<td>0.5</td>
<td>1.648721</td>
</tr>
<tr>
<td>0.75</td>
<td>2.117000</td>
</tr>
<tr>
<td>1.0</td>
<td>2.718282</td>
</tr>
</tbody>
</table>

Sum = 5.049747

$$\int_0^1 e^x \, dx \approx 0.25 \left[ \frac{1+2.718182}{2} + 5.049747 \right] \approx 1.7272095$$
Simpson’s 3-Point Rule: $\frac{1}{12}$

- Trapezoid rule is an approximation of $f(x)$ on $[x_i, x_{i+1}]$ with a line (i.e., degree 1 polynomial).
- Simpson’s 3-point rule approximates $f(x)$ on $[x_i, x_{i+2}]$ with a parabola (i.e., degree 2 polynomial) that interpolates $(x_i, f_i)$, $(x_{i+1}, f_{i+1})$ and $(x_{i+2}, f_{i+2})$. 

\[ y = f(x) \quad \text{and} \quad y = P_2(x) \]
Simpson’s 3-Point Rule: $2/12$

- Since the $P_2(x)$ that interpolates $(x_i, f_i)$, $(x_{i+1}, f_{i+1})$ and $(x_{i+2}, f_{i+2})$ can easily be found, we have

$$\int_{x_i}^{x_{i+2}} f(x) \, dx \approx \int_{x_i}^{x_{i+2}} P_2(x) \, dx$$

- Since $P_2(x)$ is a degree 2 polynomial, its integration is very easy.

- How do we find this $P_2(x)$? Should we use Lagrange or Newton divided difference?

- It turns out we don’t need these tools!
Simpson’s 3-Point Rule: \(3/12\)

- We still use equally spaced subintervals.
- Since translation does not change integration result, we translate \(x_{i+1}\) to 0 so that \(x_i = -\Delta\) and \(x_{i+2} = \Delta\). This will simplify our calculation.
- Let the interpolating polynomial be \(c + bx + ax^2\)
Simpson’s 3-Point Rule: $4/12$

The integration of the polynomial $c+bx+ax^2$ from $-\Delta$ to $\Delta$ is easy to compute as follows:

$$\int_{-\Delta}^{\Delta} (c + bx + ax^2) \, dx = cx \bigg|_{-\Delta}^{\Delta} + \frac{b}{2} x^2 \bigg|_{-\Delta}^{\Delta} + \frac{a}{3} x^3 \bigg|_{-\Delta}^{\Delta} = (2\Delta) \left[ c + \frac{a}{3} \Delta^2 \right]$$

$y = f(x)$

$x_i = -\Delta, x_{i+1} = 0, x_{i+2} = \Delta$

$b$ is not used!
Simpson’s 3-Point Rule: \(5/12\)

- How to compute \(c\) and \(a\) in \(P_2(x) = c + bx + ax^2\)?
- From the setup, we have
  
  \[
  f_i = P_2(-\Delta) = c - b\Delta + a\Delta^2 \\
  f_{i+1} = P_2(0) = c \\
  f_{i+2} = P_2(\Delta) = c + b\Delta + a\Delta^2
  \]

- Adding the first and the third equations together and solving for \(a\) yield the following:
  
  \[
a = \frac{1}{2\Delta^2} \left( f_i - 2f_{i+1} + f_{i+2} \right)
  \]
Simpson’s 3-Point Rule: 6/12

What do we have now?

\[
\int_{-\Delta}^{\Delta} f(x)dx \approx (2\Delta) \left[ c + \frac{a}{3} \Delta^2 \right]
\]

\[
c = f_{i+1}
\]

\[
a = \frac{1}{2\Delta^2} (f_i - 2f_{i+1} + f_{i+2})
\]

Plugging \(a\) and \(c\) into the integration yields:

\[
\int_{-\Delta}^{\Delta} f(x)dx \approx \frac{\Delta}{3} \left[ f_i + 4f_{i+1} + f_{i+2} \right]
\]

Note that this result does not depend on the \(x_i\)’s!
Simpson’s 3-Point Rule: $7/12$

- Compute $\int_0^1 e^x dx = e^1 - e^0 = 1.718281828...$
- We need 3 equally spaced points $x_0=0$, $x_1=0.5$ and $x_2=1$. Thus, $\Delta=0.5$.

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>$f_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>1.648721271</td>
</tr>
<tr>
<td>1.</td>
<td>2.718281828</td>
</tr>
</tbody>
</table>

\[
\int_{x_i}^{x_{i+2}} f(x) dx \approx \frac{\Delta}{3} [f_i + 4f_{i+1} + f_{i+2}]
\]

$\times 4$ + $\times (\Delta/3)$ = 1.718861152
Simpson’s 3-Point Rule: 8/12

- For a general integration problem, we need to divide $[a,b]$ into an even number of subintervals, and apply Simpson’s 3-point rule to two consecutive ones.
- More precisely, apply Simpson’s 3-point rule to $[x_0, x_1, x_2]$, $[x_2, x_3, x_4]$, $[x_4, x_5, x_6]$, ..., $[x_{n-2}, x_{n-1}, x_n]$. Note that Simpson’s 3-point rule only depends on the $f_i$’s and the length of subinterval $\Delta$. For convenience, we shall use $n = 2m$, where $n$ is the number of subintervals, and $\Delta = (b-a)/n$. 
**Simpson’s 3-Point Rule: 9/12**

- Consider the following results from integrating two consecutive subintervals.

\[
\begin{align*}
[x_0, x_1, x_2] & \quad \frac{\Delta}{3} [f_0 + 4f_1 + f_2] \\
[x_2, x_3, x_4] & \quad \frac{\Delta}{3} [f_2 + 4f_3 + f_4] \\
[x_4, x_5, x_6] & \quad \frac{\Delta}{3} [f_4 + 4f_5 + f_6] \\
& \vdots \\
[x_{2m-4}, x_{2m-3}, x_{2m-2}] & \quad \frac{\Delta}{3} [f_{2m-4} + 4f_{2m-3} + f_{m-2}] \\
[x_{2m-2}, x_{2m-1}, x_{2m}] & \quad \frac{\Delta}{3} [f_{2m-2} + 4f_{2m-1} + f_{2m}]
\end{align*}
\]

- Odd indices \( f_i \)'s have coefficient 4
- Even indices \( f_i \)'s have coefficient 2
- \( f_0 \) and \( f_{2m} \) appear exactly once
Simpson’s 3-Point Rule: 10/12

● The following is a computation scheme:

```
0 1 2 3 4 5 6 7 ........ 2m-4 2m-3 2m-2 2m-1 2m
```

$\times 2$

$\times 4$

$m$ terms

$m-1$ terms

$x(\Delta/3)$

final result
Simpson’s 3-Point Rule: 11/12

- In summary, the sum is the following:
\[
\frac{\Delta}{3} \left[ (f_0 + f_{2m}) + 4 \sum_{i=1}^{m} f_{2i-1} + 2 \sum_{i=1}^{m-1} f_{2i} \right]
\]

- Input:
  - interval: \([a,b]\)
  - \(n=2m\): # of divisions
  - Output in Result

```plaintext
\Delta = (b-a)/n
step = \Delta + \Delta
m = n/2

 odd terms
odd = 0.0
x = a + \Delta
DO i = 1, m, 2
  odd = odd + f(x)
x = x + step
END DO

 even terms
even = 0.0
x = a + \Delta + \Delta
DO i = 2, m-1, 2
  even = even + f(x)
x = x + step
END DO

Result = (f(a)+f(b)+4*odd + 2*even)*\Delta/3
```
Simpson’s 3-Point Rule: \(12/12\)

- Integrate \(\frac{1}{1+x^2}\) from 0 to 1. The answer is \(0.78539816\ldots\)

\[
\int_0^1 \frac{1}{1+x^2} \, dx = \tan^{-1}(1) - \tan^{-1}(0) = \frac{\pi}{4} = 0.785398163\ldots
\]

\(\Delta=0.25\)

<table>
<thead>
<tr>
<th>index</th>
<th>(x_i)</th>
<th>(f_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>0.94117647</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>0.75</td>
<td>0.64</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\(\times 4\) -> \(6.32470588\)

\(\times 2\) -> \(1.6\)

\(\times (\Delta/3)\) -> \(0.785392156\)
Iterative Methods

- An iterative method increases the number of subdivisions until the process converges.
- Since trapezoid and Simpson 3-point rules are simple, we shall look at how they can be modified to become “iterative.”
- We may start with a coarse subdivision of \([a,b]\), and compute the integration.
- If two successive integration results are close to each other, stop.
- Otherwise, refine the subdivision and do again!
Iterative Trapezoid Method: 1/4

- Initially, we have one subinterval \([a, b]\), and \(\Delta = b-a\).
- The integration is \(I_0 = \Delta \times (f(a) + f(b))/2\)
- In the next iteration, the length of each subinterval is halved (i.e., \(\Delta = \Delta/2\)) and the number of subintervals is doubled.
- Thus, if the previously computed result \(I_n\) with \(n\) subintervals is not very different from the newly computed result \(I_{2n}\) with \(2n\) subintervals, then stop. Otherwise, start the next iteration.
Iterative Trapezoid Method: 2/4

- Trapezoid method has a simple relationship between $I_n$ and $I_{2n}$.
- It uses the following formula:
  \[
  \int_a^b f(x)dx \approx \Delta \left[ \frac{f_0 + f_n}{2} + \sum_{i=1}^{n-1} f_i \right]
  \]
- Let $K_n$ be the previous sum, then $K_{2n}$ can be obtained by adding new values.

Only need to update this sum!
**Iterative Trapezoid Method:** \( 3/4 \)

The integral \( \int_{0}^{1} \frac{1}{1 + x^2} \, dx \approx \frac{\pi}{4} \)

\[
\pi / 4 \approx 0.785398163
\]

\[
\Delta = 0.5 \\
\Delta = 0.25 \\
\Delta = 0.125
\]

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \Delta = 0.5 )</th>
<th>( \Delta = 0.25 )</th>
<th>( \Delta = 0.125 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.125</td>
<td></td>
<td></td>
<td>0.984615384</td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td>0.94117647</td>
<td></td>
</tr>
<tr>
<td>0.375</td>
<td></td>
<td>0.876712328</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.625</td>
<td></td>
<td>0.719101123</td>
<td></td>
</tr>
<tr>
<td>0.75</td>
<td>0.64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.875</td>
<td></td>
<td>0.566371681</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Prev. Sum:** 0 0.8 2.381176471

**New Sum:** 0.8 1.58117647 3.146800518

**This Sum:** 0.8 2.381176471 5.527976989

**Integration:** 0.775 0.782794117 0.784747123

\[
f(0) + f(1) = 0.75
\]
Iterative Trapezoid Method: 4/4

The following is a possible algorithm:

\begin{verbatim}
! Initialization
! Int  - integration
! This  - new integration
! Prev - previous sum
! Next  - new sum

Fixed = (f(a) + f(b))/2
Int   = Fixed
Prev  = 0
\Delta = b-a
Intervals = 1

DO
    \Delta2 = \Delta/2
    x = a + \Delta2
    Next = 0
    DO i = 1, Intervals
        Next = Next + f(x)
        x = x + \Delta
    END DO
    This = (Fixed+Prev+Next)*\Delta2
    IF (|This - Int| < \varepsilon) EXIT
    Prev = Next
    Int = This
    \Delta = \Delta2
    Intervals = Intervals*2
END DO
\end{verbatim}
Iterative Simpson Method: 1/3

- Simpson method can also be made iterative.
- All newly added points have *odd* indices!
- All original points have *even* indices!

\[
\int_{a=x_0}^{b=x_2^{2m}} f(x) = \frac{\Delta}{3} \left[ (f_0 + f_{2m}) + 4 \sum_{i=1}^{m} f_{2i-1} + 2 \sum_{i=1}^{m-1} f_{2i} \right]
\]

![Diagram of Simpson's method](image.png)
**Iterative Simpson Method: 2/3**

\[
\int_0^1 \frac{1}{1 + x^2} \, dx \approx \frac{\pi}{4}
\]

\[
f(0) + f(1) = 1.5
\]

\[
\pi / 4 \approx 0.785398163
\]

<table>
<thead>
<tr>
<th></th>
<th>$\Delta=0.5$</th>
<th>$\Delta=0.25$</th>
<th>$\Delta=0.125$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.984615384</td>
</tr>
<tr>
<td>0.125</td>
<td></td>
<td>0.94117647</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td></td>
<td></td>
<td>0.876712328</td>
</tr>
<tr>
<td>0.375</td>
<td>0.8</td>
<td>0.719101123</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>0.625</td>
<td></td>
<td></td>
<td>0.566371681</td>
</tr>
<tr>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>2.38117647</td>
<td></td>
</tr>
<tr>
<td>Previous</td>
<td>0</td>
<td>0.8</td>
<td>3.146800516</td>
</tr>
<tr>
<td>This Odd</td>
<td>0.8</td>
<td>1.58117647</td>
<td></td>
</tr>
<tr>
<td>Integration</td>
<td>0.78333…</td>
<td>0.785392156</td>
<td>0.785398125</td>
</tr>
</tbody>
</table>
Iterative Simpson Method: 3/3

\[
\text{DO} \\
\text{New\_Even} = \text{Even} + \text{Odd} \\
\text{New\_Odd} = 0 \\
\Delta 2 = \Delta / 2 \\
x = a + \Delta 2 \\
\text{DO} \ i = 1, \text{Intervals} \\
\quad \text{New\_Odd} = \text{New\_Odd} + f(x) \\
\quad x = x + \Delta \\
\text{END DO} \\
\text{New\_Int} = \left( \text{Fixed} + 4 \times \text{New\_Odd} + 2 \times \text{New\_Even} \right) \times \Delta 2 / 3 \\
\text{IF} \ (|\text{New\_Int} - \text{Int}| < \varepsilon) \text{ EXIT} \\
\text{Even} = \text{New\_Even} \\
\text{Odd} = \text{New\_Odd} \\
\text{Int} = \text{New\_Int} \\
\text{Intervals} = \text{Intervals} + \text{Intervals} \\
\Delta = \Delta 2 \\
\text{END DO}
\]
Romberg’s Method: 1/4

Romberg’s method for integration is similar to Richardson’s method for differentiation.

Romberg’s method extrapolates the results from two successive values computed with the iterative trapezoid (or Simpson) method.

\[ r_{i,j+1} = r_{i,j} + \frac{r_{i,j} - r_{i-1,j}}{4^{j+1} - 1} \]

\( I_n: \) trapezoid method using \( n \) subintervals

use Richardson's formula
Romberg’s Method: 2/4

Romberg’s method requires an update from $I_n$ to $I_{2n}$, where $I_k$ is the integration from $k$ intervals.

$I_n$ is computed as follows:

$$I_n = \Delta_n \left[ \frac{f_0 + f_n}{2} + \sum_{i=1}^{n-1} f_i \right]$$

$I_{2n}$ is computed as

$$I_{2n} = \Delta_{2n} \left[ \frac{f_0 + f_1}{2} + (\text{original}) + (\text{new}) \right]$$

Since $\Delta_{2n} = \Delta_n / 2$, we have

$$I_{2n} = I_n / 2 + \Delta_{2n} \times (\text{new})$$
Romberg Method: 3/4

- The left is Romberg’s method
- \( n \): number of rows
- Result is in \( r_{n,n} \)

\[
\begin{align*}
\Delta &= b-a \\
r_{0,0} &= (f(a) + f(b)) \cdot \Delta/2 \\
\text{intervals} &= 1 \\
\text{DO} \ i = 1, \ n \\
\Delta^2 &= \Delta/2 \\
x &= a + \Delta \\
\text{sum} &= 0 \\
\text{DO} \ k = 1, \ \text{intervals} \\
\text{sum} &= \text{sum} + f(x) \\
x &= x + \Delta \\
\text{END DO} \\
r_{i,0} &= r_{i-1,0}/2 + \Delta^2 \cdot \text{sum} \\
\text{DO} \ j = 0, \ i-1 \\
r_{i,j+1} &= r_{i,j} + \frac{r_{i,j} - r_{i-1,j}}{4^{j+1} - 1} \\
\text{END DO} \\
\Delta &= \Delta^2 \\
\text{intervals} &= \text{intervals} \cdot 2 \\
\text{END DO}
\end{align*}
\]

trapezoid rule

Richardson extrapolation
Romberg Method: 4/4

● The following is an example:

\[
\int_{0}^{1} \frac{1}{1 + x^2} \, dx = \tan^{-1}(1) - \tan^{-1}(0) = \frac{\pi}{4} = 0.785398163...
\]

0.75

0.775 → 0.7833333

0.7827941 → 0.78539216 → 0.78552943

0.7847471 → 0.7853981 → 0.78539854 → 0.78539645

0.7852354 → 0.7853982 → 0.7853982 → 0.7853982 → 0.7853982
The End