NUMERICAL DIFFERENTIATION

Used mainly to numerically solve differential equations (next chapter).

In addition to **truncation** error, we must also watch out for **round off** errors.

The **main idea**: select a few nodes (at and near x_0), evaluate $y(x)$ at these, fit interpolating polynomial, differentiate that instead.

Example:

Derive formula for $y'(x_0)$, using $x_0 - h$, x_0 and $x_0 + h$ as nodes.

Interpolating polynomial:

$$\frac{(x-x_0)(x-x_0-h)}{(-h)\cdot(-2h)}\,y(x_0-h)+$$
$$\frac{(x-x_0+h)(x-x_0-h)}{h\cdot(-h)}\,y(x_0)+$$
$$\frac{(x-x_0+h)(x-x_0)}{2h\cdot h}\,y(x_0+h)$$

Differentiate, then substituting $x = x_0$:

$$\begin{aligned} y'(x_0) &\simeq \frac{y(x_0-h)}{-2h}+\frac{y(x_0+h)}{2h} \\ &= \frac{y(x_0+h)-y(x_0-h)}{2h} \end{aligned}$$

(the usual slope between the end points).

Taylor expanding the RHS yields:

$$y'(x_0)+\frac{y'''(x_0)}{6}h^2+\frac{y^{v}(x_0)}{120}h^4+....$$

The truncation error:

$$\frac{y'''(x_0)}{6}h^2+\frac{y^{v}(x_0)}{120}h^4+....$$

Unfortunately, we cannot reduce h indefinitely (round off error).

Formula demonstration:

Approximate $y'(1)$ where $y(x) = \exp(x^2)$. (The exact value is $2e = 5.43656\,3656$).

h	$\frac{\exp[(1+h)^2]-\exp[(1-h)^2]}{2h}$	Error:
0.1	5. 52788 333	0.0913197
0.01	5. 43746 980	0.0009061
0.001	5. 43657 250	0.0000088
0.0001	5. 43656000	0.0000037
0.00001	5. 43655000	0.0000136
0.000001	5. 43650000	0.0000636

The error should decrease by a factor of 100 in each step - does it?

Richardson Extrapolation (new name for the old Romberg).
Example:
(same as before, h reduced more slowly):

h	$R_i = \frac{\exp[(1+h)^2]-\exp[(1-h)^2]}{2h}$	$S_i = \frac{4R_{i+1}-R_i}{3}$	$T_i = \frac{16S_{i+1}-S_i}{15}$
$\frac{1}{4}$	6.03135 7050	**5.4**2938 7349	**5.43657** 7469
$\frac{1}{8}$	**5.**57987 9776	**5.436**12 8086	**5.43656 3**886
$\frac{1}{16}$	**5.4**7206 6010	**5.4365**3 6649	**5.43656 3**708
$\frac{1}{32}$	**5.4**4541 8990	**5.43656** 2016	
$\frac{1}{64}$	**5.43**877 6260		

$\frac{64T_{i+1}-T_i}{63}$
5.43656 3669
5.43656 3704

(as close as we can get to the exact answer).

Higher-Degree (more nodes) **Formulas**

To obtain a more accurate formula for $y'(x_0)$, we will now use $x = x_0$, $x_0 \pm h$ *and* $x_0 \pm 2h$ as nodes.

Two ways of simplifying the derivation:

1. Assume that $x_0 = 0$ and $h = 1$ (the old X scale), then transform back to x.

   Interpolating polynomial:

$$\frac{(X+1)(X-1)(X-2)}{(-1)\times(-3)\times(-4)}Y(-2) + \frac{(X+2)(X-1)(X-2)}{1\times(-2)\times(-3)}Y(-1) + \frac{(X+2)(X+1)(X-2)}{3\times2\times(-1)}Y(1) + \frac{(X+2)(X+1)(X-1)}{4\times3\times1}Y(2)$$

   Differentiate, then set $X = 0$:

$$\frac{(-1)(-2)+1(-2)+1(-1)}{-12}Y(-2) + \frac{(-1)(-2)+2(-2)+2(-1)}{6}Y(-1) + \frac{1(-2)+2(-2)+2\cdot1}{-6}Y(1) + \frac{1(-1)+2(-1)+2\cdot1}{12}Y(2) = \frac{1}{12}Y(-2) - \frac{8}{12}Y(-1) + \frac{8}{12}Y(1) - \frac{1}{12}Y(2)$$

   Transform back to x:

$$y'(x_0) \simeq \frac{y(x_0-2h) - 8\,y(x_0-h) + 8\,y(x_0+h) - y(x_0+2h)}{12h}$$

2. Knowing that the formula must be anti-symmetric (for any odd derivative), i.e.

$$-c_2\,Y(-2) - c_1Y(-1) + c_1Y(1) + c_2Y(2)$$

and must be correct for $Y(X) = X$, X^3 (it's **automatically** correct for 1, X^2 and X^4):

$$\begin{aligned} 2c_2 + c_1 + c_1 + 2c_2 &= 1 \\ 8c_2 + c_1 + 8c_1 + c_2 &= 0 \end{aligned}$$

implying $c_1 = -8c_2$ and $c_2 = -\frac{1}{12}$ (same answer as before).

Taylor expanding:

$$\begin{aligned}
&y(x_0 - 2h) = \\
&y(x_0) - 2h\,y'(x_0) + \frac{4h^2}{2}y''(x_0) - \frac{8h^3}{6}y'''(x_0) + \frac{16h^4}{24}y^{iv}(x_0) - ... \\
&y(x_0 - h) = \\
&y(x_0) - h\,y'(x_0) + \frac{h^2}{2}y''(x_0) - \frac{h^3}{6}y'''(x_0) + \frac{h^4}{24}y^{iv}(x_0) - ... \\
&y(x_0 + h) = \\
&y(x_0) + h\,y'(x_0) + \frac{h^2}{2}y''(x_0) + \frac{h^3}{6}y'''(x_0) + \frac{h^4}{24}y^{iv}(x_0) + ... \\
&y(x_0 + 2h) = \\
&y(x_0) + 2h\,y'(x_0) + \frac{4h^2}{2}y''(x_0) + \frac{8h^3}{6}y'''(x_0) + \frac{16h^4}{24}y^{iv}(x_0) + ...
\end{aligned}$$

Our new formula thus yields:

$$y'(x_0) - \frac{h^4}{30}y^{v}(x_0) + ...$$

(a lot smaller truncation error).

Example: (same derivative, one more time):

h	$R_i = \frac{-\exp[(1+2h)^2]+8\exp[(1+h)^2]-8\exp[(1-h)^2]+\exp[(1-2h)^2]}{12h}$
$\frac{1}{4}$	$\mathbf{5}.30723\,9257$
$\frac{1}{8}$	$\mathbf{5}.\mathbf{4}2938\,7358$
$\frac{1}{16}$	$\mathbf{5}.\mathbf{436}12\,8081$
$\frac{1}{32}$	$\mathbf{5}.\mathbf{4365}3\,6647$

$T_i = \frac{16R_{i+1} - R_i}{15}$	$\frac{64T_{i+1} - T_i}{63}$
$\mathbf{5}.\mathbf{43}753\,0565$	$\mathbf{5}.\mathbf{43656}\,2332$
$\mathbf{5}.\mathbf{43657}\,7463$	$\mathbf{5}.\mathbf{43656}\,\mathbf{36}67$
$\mathbf{5}.\mathbf{43656}\,\mathbf{3}885$	

(reasonably accurate, even without Richardson).

Nonsymmetric Spacing

Choice of nodes may be restricted (say, to one side of x_0).

Example:
Approximate $y'(x_0)$, using x_0, x_0+h and x_0+2h as nodes.
In terms of X:

$$c_0 Y(0) + c_1 Y(1) + c_2 Y(2)$$

Make it exact for $Y(X) = 1$, X and X^2:

$$\begin{aligned} c_0 + c_1 + c_2 &= 0 \\ c_1 + 2c_2 &= 1 \\ c_1 + 4c_2 &= 0 \end{aligned}$$

which yields $c_2 = -\frac{1}{2}$, $c_1 = 2$, $c_0 = -\frac{3}{2}$.
The resulting formula:

$$y'(x_0) \simeq \frac{-3y(x_0) + 4y(x_0+h) - y(x_0+2h)}{2h}$$

Its right hand side expands to

$$y'(x_0) - \frac{h^2}{3} y'''(x_0) - \frac{h^3}{4} y^{iv}(x_0) - ...$$

The main error term is proportional to h^2 (since we have three nodes).
Example:
Applied to our benchmark problem, this yields:

h	$R_i = \frac{-3\exp(1)+4\exp[(1+h)^2]-\exp[(1+2h)^2]}{2h}$	$S_i = \frac{4R_{i+1}-R_i}{3}$
$\frac{1}{16}$	**5.** 35149 250	**5. 43**909 0859
$\frac{1}{32}$	**5. 4**1719 127	**5. 436**85 1443
$\frac{1}{64}$	**5. 43**193 640	

$\frac{8S_{i+1}-S_i}{7}$

5. 43653 1527

(performs worse than the symmetric rule).

Higher Derivatives

To approximate $y''(x_0)$, we first derive a basic three-node (symmetric) formula.

The interpolating polynomial is the same, differentiating it *twice* yields the following result:

$$\frac{y(x_0-h) - 2y(x_0) + y(x_0+h)}{h^2}$$

Taylor expanding:

$$y''(x_0) + \frac{y^{iv}(x_0)}{12} h^2 + ...$$

The round-off error is now substantially bigger than in the $y'(x_0)$ case.
Example:

Approximate the second derivative of $\exp(x^2)$ at $x = 1$ (equal to $6e = 16.30969\,097$):

h	$R_i = \frac{\exp[(1-h)^2]-2\exp(1)+\exp[(1+h)^2]}{h^2}$	$S_i = \frac{4R_{i+1}-R_i}{3}$	$\frac{16S_{i+1}-S_i}{15}$
$\frac{1}{16}$	$\mathbf{16.3}77100$	$\mathbf{16.309}653$	$\mathbf{16.3097}12$
$\frac{1}{32}$	$\mathbf{16.3}26515$	$\mathbf{16.3097}08$	
$\frac{1}{64}$	$\mathbf{16.3}13910$		

The second stage of Richardson extrapolation no longer improves (due to round-off error), unless 15-digit accuracy used:

h	$R_i = \frac{\exp[(1-h)^2]-2\exp(1)+\exp[(1+h)^2]}{h^2}$
$\frac{1}{16}$	$\mathbf{16.3}7709\,985$
$\frac{1}{32}$	$\mathbf{16.3}2651\,323$
$\frac{1}{64}$	$\mathbf{16.3}1389\,467$

$S_i = \frac{4R_{i+1}-R_i}{3}$	$\frac{16S_{i+1}-S_i}{15}$
$\mathbf{16.3096}5\,102$	$\mathbf{16.30969\,09}8$
$\mathbf{16.30968}\,848$	

Finally, we derive a formula to approximate $y'''(x_0)$.

Minimum of four nodes is needed (always the order of the derivative plus one).

We will choose them at $x_0 \pm h$ and $x_0 \pm 2h$.

In the X scale, the formula must read (utilizing its antisymmetry):

$$-c_2\,Y(-2) - c_1\,Y(-1) + c_1\,Y(1) + c_2\,Y(2)$$

Furthermore, it must yield the exact answer with $Y(X) = X$ and X^3:

$$\begin{aligned} 2c_2 + c_1 + c_1 + 2c_2 &= 0 \\ 8c_2 + c_1 + c_1 + 8c_2 &= 6 \end{aligned}$$

which implies $c_1 = -2c_2$ and $c_2 = \frac{1}{2}$.

We thus obtain

$$Y'''(0) \simeq \frac{Y(2) - 2Y(1) + 2Y(-1) - Y(-2)}{2}$$

or, going back to $y(x)$:

$$y'''(x_0) \simeq \frac{y(x_0 + 2h) - 2y(x_0 + h) + 2y(x_0 - h) - y(x_0 - 2h)}{2h^3}$$

The right hand side expands to

$$y'''(x_0) + \frac{y^v(x_0)}{4}\,h^2 + ...$$

Example:

Approximate the third derivative of $\exp(x^2)$ at $x = 1$ (the exact answer is $20e = 54.36563\,657$).

We will carry out the computation using 20-digit accuracy:

h	$R_i = \frac{\exp[(1+2h)^2]-2\exp[(1+h)^2]+2\exp[(1-h)^2]-\exp[(1-2h)^2]}{2h^3}$
$\frac{1}{32}$	$\mathbf{54.}\,57311\,583$
$\frac{1}{64}$	$\mathbf{54.\,4}1742\,711$
$\frac{1}{128}$	$\mathbf{54.\,37}857\,926$

$S_i = \frac{4R_{i+1}-R_i}{3}$	$\frac{16S_{i+1}-S_i}{15}$
$\mathbf{54.\,3655}3\,087$	$\mathbf{54.\,36563\,65}9$
$\mathbf{54.\,36562}\,998$	

Summary:

To derive an m-node (m must be bigger than n) formula for $y^{(n)}(x_0)$:

1. In the X scale, find the values of the c-coefficients by making the formula exact for $y = 1$, X, X^2, ... X^m.

2. For symmetric spacing, the c-coefficients are either symmetric (n even - use 1, X^2, ...only) or antisymmetric (n odd - use X, X^3, ...only).

3. Convert to x scale by changing y_0 to $y(x_0)$, y_1 to $y(x_0 + h)$, etc., and dividing by h^n.

4. The truncation error has terms proportional to h^{m-n}, h^{m-n+1}, h^{m-n+2}, etc. in the general (non-symmetric) case. When the nodes are chosen symmetrically, the odd-power (in h) terms disappear.

The **round off error** is due to subtracting numbers of very similar size (this happens in all these formulas, but is more pronounced for higher derivatives). For example, computing $y(x_0 + h) - y(x_0 - h)$, when $y = \exp(x^2)$, $x_0 = 1$, and $h = 0.0001$ yields:

$$\begin{array}{r} 2.718553670 \\ -2.718010013 \\ = 0.000543657 \end{array}$$

Computing $y(x_0 + h) + y(x_0 - h) - 2y(x_0)$ becomes, using 10 digits, almost impossible:

$$\begin{array}{r} 5.436563683 \\ -5.436563656 \\ = 0.000000027 \end{array}$$