Bonus: Euler's method applied to e^x

Example 143. Consider the IVP $y'=y$, $y(0)=1$. Approximate the solution $y(x)$ for $x\in[0,1]$ using Euler's method with 4 steps. In particular, what is the approximation for $y(1)$?

Comment. Of course, the real solution is $y(x)=e^x$. In particular, $y(1)=e\approx 2.71828$.

Solution. The step size is $h=\frac{1-0}{4}=\frac{1}{4}$. We apply Euler's method with $f(x,y)=y$:

$$\begin{aligned}
x_0&=0 & y_0&=1\\
x_1&=\frac{1}{4} & y_1&=y_0+f(x_0,y_0)h=1+\frac{1}{4}=\frac{5}{4}=1.25\\
x_2&=\frac{1}{2} & y_2&=y_1+f(x_1,y_1)h=\frac{5}{4}+\frac{5}{4}\cdot\frac{1}{4}=\frac{5^2}{4^2}=1.5625\\
x_3&=\frac{3}{4} & y_3&=y_2+f(x_2,y_2)h=\frac{5^2}{4^2}+\frac{5^2}{4^2}\cdot\frac{1}{4}=\frac{5^3}{4^3}\approx 1.9531\\
x_4&=1 & y_4&=y_3+f(x_3,y_3)h=\frac{5^3}{4^3}+\frac{5^3}{4^3}\cdot\frac{1}{4}=\frac{5^4}{4^4}\approx 2.4414
\end{aligned}$$

In particular, the approximation for $y(1)$ is $y_4\approx 2.4414$.

Comment. Can you see that, if instead we start with $h=\frac{1}{n}$, then we similarly get $x_i=\frac{(n+1)^i}{n^i}$ for $i=0,1,\ldots,n$? In particular, $y(1)\approx y_n=\frac{(n+1)^n}{n^n}=\left(1+\frac{1}{n}\right)^n\to e$ as $n\to\infty$. Do you recall how to derive this final limit?

Example 144. (cont'd) Consider the IVP $y'=y$, $y(0)=1$. Approximate the solution $y(x)$ for $x\in[0,1]$ using Euler's method with n steps for several values of n. In each case, what is the approximation for $y(1)$?

Solution. Since the real solution is $y(x)=e^x$ so that, in particular, the exact solution is $y(1)=e\approx 2.71828$. We proceed as we did in Example 143 in the case $n=4$ and apply Euler's method with $f(x,y)=y$:

$$\begin{aligned}
x_{n+1} &= x_n+h\\
y_{n+1} &= y_n+h\underbrace{f(x_n,y_n)}_{=y_n}=(1+h)y_n
\end{aligned}$$

We observe that it follows from $y_{n+1}=(1+h)y_n$ that $y_n=(1+h)^n y_0$. Since $y_0=1$ and $h=\frac{1-0}{n}=\frac{1}{n}$, we conclude that

$$x_n=1,\quad y_n=\left(1+\frac{1}{n}\right)^n.$$

[For instance, for $n=4$, we get $x_4=1$, $y_4=\left(\frac{5}{4}\right)^4\approx 2.4414$ as in Example 143.]

In particular, our approximation for $y(1)$ is $\left(1+\frac{1}{n}\right)^n$.

Armin Straub
straub@southalabama.edu

Here are a few values spelled out:

$$\begin{aligned}
n=1: &\quad \left(1+\frac{1}{n}\right)^n = 2 \\
n=4: &\quad \left(1+\frac{1}{n}\right)^n = 2.4414... \\
n=12: &\quad \left(1+\frac{1}{n}\right)^n = 2.6130... \\
n=100: &\quad \left(1+\frac{1}{n}\right)^n = 2.7048... \\
n=365: &\quad \left(1+\frac{1}{n}\right)^n = 2.7145... \\
n=1000: &\quad \left(1+\frac{1}{n}\right)^n = 2.7169... \\
n\to\infty: &\quad \left(1+\frac{1}{n}\right)^n \to e = 2.71828...
\end{aligned}$$

We can see that Euler's method converges to the correct value as $n\to\infty$. On the other hand, we can see that it doesn't converge impressively fast. That is why, for serious applications, one usually doesn't use Euler's method directly but rather higher-order methods derived from the same principles (such as Runge–Kutta methods).

Interpretation. Note that we can interpret the above values in terms of compound interest. We start with initial capital of $y(0)=1$ and we are interested in the capital $y(1)$ after 1 year if we receive interest at an annual rate of 100%:

- If we receive a single interest payment at the end of the year, then $y(1)=2$ (case $n=1$ above).
- If we receive quarterly interest payments of $\frac{100\%}{4}=25\%$ each, then $y(1)=(1.25)^4=2.441...$ (case $n=4$).
- If we receive monthly interest payments of $\frac{100\%}{12}=\frac{1}{12}$ each, then $y(1)=2.6130...$ (case $n=12$).
- If we receive daily interest payments of $\frac{100\%}{365}=\frac{1}{365}$ each, then $y(1)=2.7145...$ (case $n=365$).

It is natural to wonder what happens if interest payments are made more and more frequently. Well, we already know the answer! If interest is compounded continuously, then we have e in our bank account after one year.

Taylor methods

(Taylor method of order k) The following is an order k method for solving IVPs:

$$\begin{aligned}
x_{n+1} &= x_n + h \\
y_{n+1} &= y_n + f(x_n,y_n)h + \frac{1}{2}f'(x_n,y_n)h^2 + \cdots + \frac{1}{k!}f^{(k-1)}(x_n,y_n)h^k
\end{aligned}$$

where $f^{(n)}(x,y)$ is short for $\frac{\mathrm{d}^n}{\mathrm{d}x^n}f(x,y(x))$ (expressed in terms of f and its partial derivatives).

For instance. $f'(x,y)=\frac{\mathrm{d}}{\mathrm{d}x}f(x,y(x))=f_x(x,y)+f_y(x,y)y'(x)=f_x(x,y)+f_y(x,y)f(x,y)$

Especially for higher derivatives, it is easier to compute these for specific f. See next example.

Comment. As for Euler's method, being an order k method means that the method has a global error that is $O(h^k)$ (while the local truncation error is $O(h^{k+1})$; note that we can see this because we truncate the Taylor expansion of $y(x)$ after h^k so that the next term is $O(h^{k+1})$).

Example 145. Spell out the Taylor method of order 2 for numerically solving the IVP

$$y' = \cos(x)y, \quad y(0) = 1.$$

Solution. The Taylor method of order 2 is based on the Taylor expansion

$$y(x+h) = y(x) + y'(x)h + \tfrac{1}{2}y''(x)h^2 + O(h^3),$$

where we have a local truncation error of $O(h^3)$ so that the global error will be $O(h^2)$.
From the DE we know that $y'(x) = \cos(x)y$, which is $f(x,y)$. We differentiate this to obtain

$$\begin{aligned} y''(x) &= \frac{\mathrm{d}}{\mathrm{d}x}\cos(x)y = -\sin(x)y + \cos(x)y' = -\sin(x)y + \cos^2(x)\, y \\ &= (-\sin(x) + \cos^2(x))y, \end{aligned}$$

which is $f'(x,y)$. Hence, the Taylor method of order 2 takes the form:

$$\begin{aligned} y_{n+1} &= y_n + f(x_n, y_n)h + \tfrac{1}{2}f'(x_n, y_n)h^2 \\ &= y_n + \cos(x_n)y_n\, h + \tfrac{1}{2}((-\sin(x_n) + \cos^2(x_n))y_n)h^2 \end{aligned}$$

For any choice of h, we can therefore compute $(x_1, y_1), (x_2, y_2), \ldots$ starting with (x_0, y_0) by the above recursive formula combined with $x_{n+1} = x_n + h$.

Example 146. Spell out the Taylor method of order 3 for numerically solving the IVP

$$y' = \cos(x)y, \quad y(0) = 1.$$

Solution. The Taylor method of order 3 is based on the Taylor expansion

$$y(x+h) = y(x) + y'(x)h + \tfrac{1}{2}y''(x)h^2 + \tfrac{1}{6}y'''(x)h^3 + O(h^4),$$

where we have a local truncation error of $O(h^4)$ so that the global error will be $O(h^3)$.
From the DE we know that $y'(x) = \cos(x)y$, which is $f(x,y)$. As in the previous example, we differentiate this to obtain

$$\begin{aligned} y''(x) &= \frac{\mathrm{d}}{\mathrm{d}x}\cos(x)y = -\sin(x)y + \cos(x)y' = -\sin(x)y + \cos^2(x)\, y \\ &= (-\sin(x) + \cos^2(x))y, \end{aligned}$$

which is $f'(x,y)$. Once more differentiating this, we obtain

$$\begin{aligned} y'''(x) &= \frac{\mathrm{d}}{\mathrm{d}x}(-\sin(x) + \cos^2(x))y \\ &= (-\cos(x) - 2\cos(x)\sin(x))y + (-\sin(x) + \cos^2(x))y' \\ &= (\cos^2(x) - 3\sin(x) - 1)\cos(x)y, \end{aligned}$$

which is $f''(x,y)$. Hence, the Taylor method of order 3 takes the form:

$$\begin{aligned} y_{n+1} &= y_n + f(x_n, y_n)h + \tfrac{1}{2}f'(x_n, y_n)h^2 + \tfrac{1}{6}f''(x_n, y_n)h^3 \\ &= y_n + \cos(x_n)y_n\, h + \tfrac{1}{2}((-\sin(x_n) + \cos^2(x_n))y_n)h^2 + \tfrac{1}{6}((\cos^2(x_n) - 3\sin(x_n) - 1)\cos(x_n)y_n)h^3 \end{aligned}$$

Comment. The exact solution of this IVP is $y(x) = e^{\sin(x)}$. We use this in the next example for comparison.

Example 147. Python Let us use Python to approximate the solution $y(x)$ of the IVP from the previous example for $x \in [0, 2]$.

```
>>> from math import e, cos, sin

>>> def taylor_3_cosy(x0, y0, xmax, n):
        h = (xmax - x0) / n
        ypoints = [y0]
        for i in range(n):
            y0 = y0 + cos(x0)*y0*h + 1/2*(cos(x0)**2-sin(x0))*y0*h**2 + \
                 1/6*(cos(x0)**2-3*sin(x0)-1)*cos(x0)*y0*h**3
            x0 = x0 + h
            ypoints.append(y0)
        return ypoints
```

Since the exact solution is $y(x) = e^{\sin(x)}$, we have $y(2) = e^{\sin(2)} \approx 2.48258$.

```
>>> taylor_3_cosy(0, 1, 2, 4)

   [1, 1.625, 2.3475297541746047, 2.7350418255304874, 2.476391322837691]
```

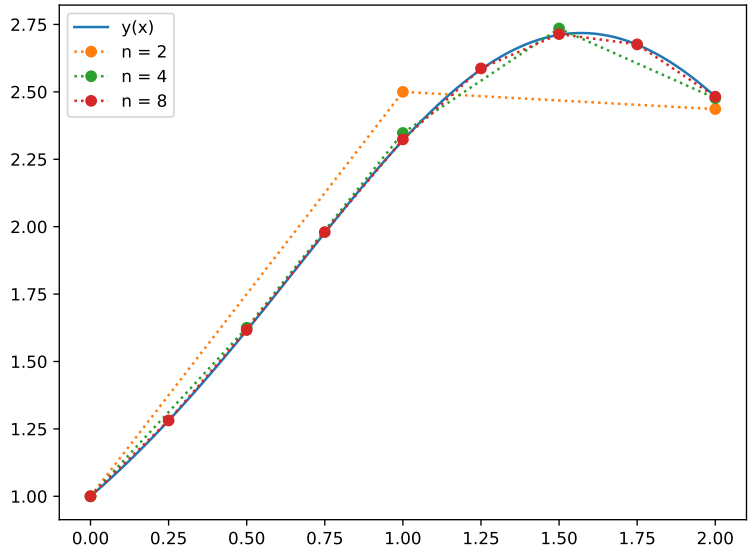
For comparison, the exact values of the solution at these four steps are:

```
>>> [e**sin(i/2) for i in range(5)]

   [1.0, 1.6151462964420837, 2.319776824715853, 2.7114810176821584, 2.4825777280150003]
```

The following convincingly illustrates that the error is indeed $O(h^3)$.

```
>>> [taylor_3_cosy(0, 1, 2, 10**n)[-1] - e**sin(2) for n in range(5)]

   [2.5174222719849997, -0.0002461575553160955, -1.6375769584797695e-07,
   -1.5647971807197791e-10, 2.0339285811132868e-13]
```

The following is a plot of the exact solution together with our approximations when using 2, 4 and 8 steps. Already for 4 steps, we obtain an approximation that, at least visually, is remarkably good.



On the other hand, for comparison, the following plot shows the corresponding approximations when using Euler's method instead.