2.6 A Numerical Method (Euler's Method)

a lecture for MATH F302 Differential Equations

Ed Bueler, Dept. of Mathematics and Statistics, UAF

Fall 2023

for textbook: D. Zill, *A First Course in Differential Equations with Modeling Applications*, 11th ed.

where we stand

- we now have methods for generating by-hand solutions to first-order differential equations:
  - 2.2 separable equations: $y' = g(x)h(y)$
  - 2.3 linear equations: $y' + P(x)y = f(x)$
  - 2.4 exact equations: $M\,dx + N\,dy = 0$ where $\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}$
- there are further methods . . . such as in section 2.5
  - but we are skipping §2.5; its methods are weak
- where do we stand?:
  - *there are some problems we can do . . .*
  - *but often our by-hand calculus/algebra techniques don't work*
- this situation is permanent

example 1

- Example 1. solve the initial value problem

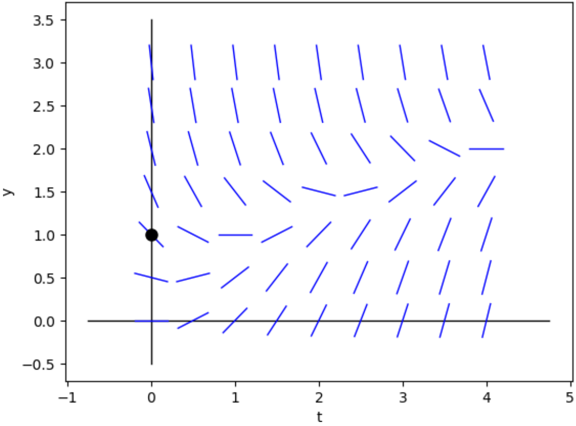
$$\frac{dy}{dt} = t - y^2, \qquad y(0) = 1$$

in particular, find $y(4)$

Solution version 0: *Explain why 2.2–2.4 methods don't apply.*

example 1, cont.

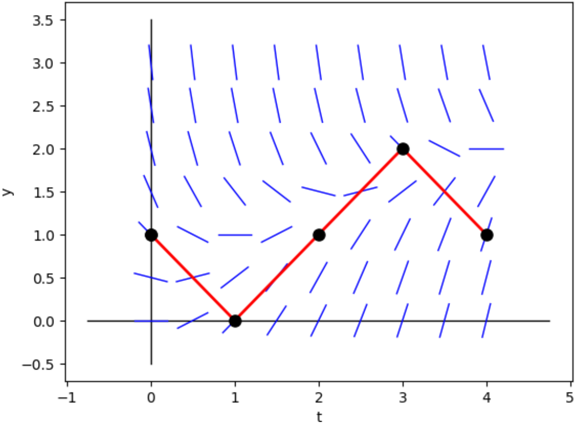
Solution version 1: *Solve it using a direction field and a pencil.*



- this is only approximate

example 1, cont. cont.

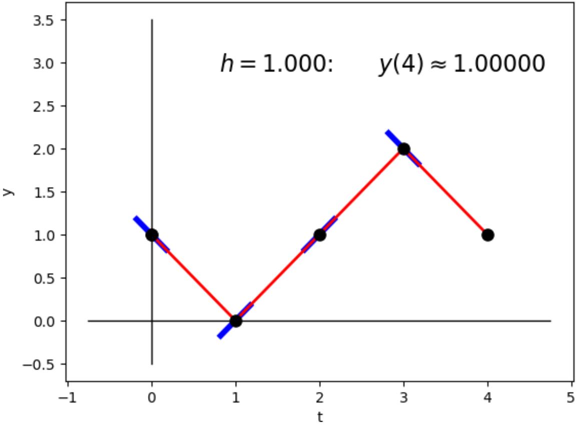
Solution version 2: *Make a computer follow the direction field.*



- this is still only approximate because we go straight

example 1, cont. cont. cont.

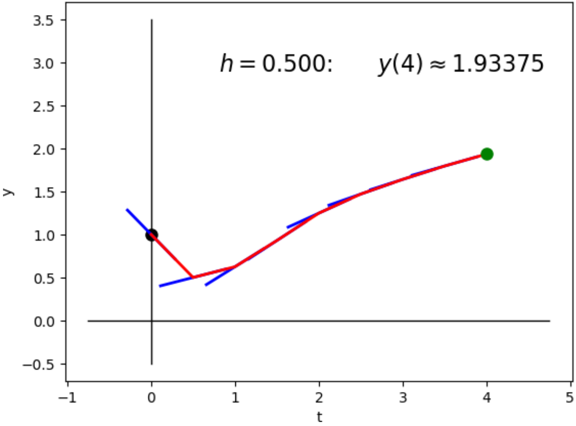
Solution version 3: *The direction field is not actually needed.*



- this is the same as previous

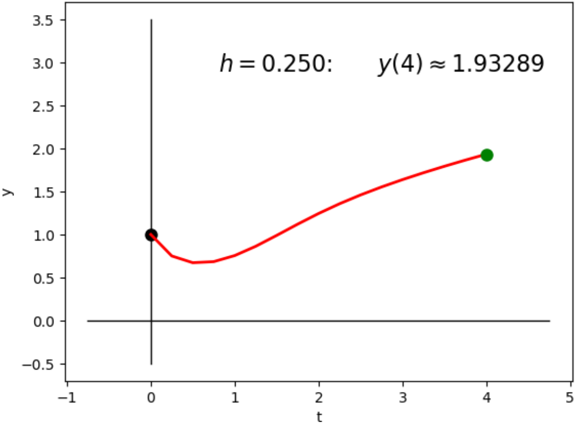
example 1, cont. cont. cont. cont.

Solution version 4: *Do it more accurately by smaller steps*



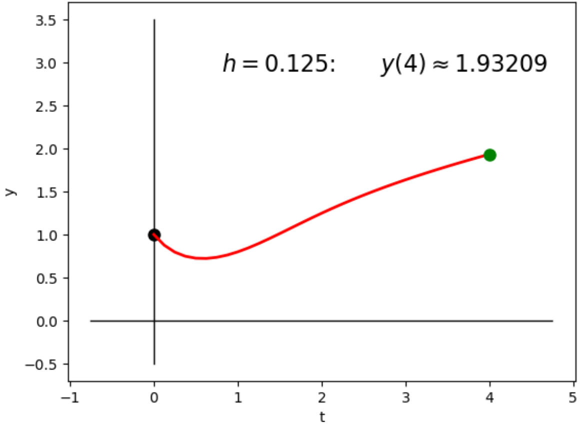
- the blue slope lines are not really needed . . .

Solution version 5: *Smaller steps.*



- this is *still* only approximate

Solution version 6: *Smaller. (Make the computer do more work.)*



- this looks like a *solution* not a direction field

Euler's method

- the idea of following the direction field, in a straight line for a short distance, and repeating, is *Euler's method*
- for the general DE $\frac{dy}{dx} = f(x, y)$, Euler's method is

$$y_{n+1} = y_n + h\, f(x_n, y_n) \qquad (*)$$

  - $h \neq 0$ is a step size *you must choose*
  - the next x-value is always h away from the last: $x_{n+1} = x_n + h$
  - $(*)$ is a formula to understand *and* memorize
  - ... and put in computer programs
- in the previous slides we had $f(x, y) = x - y^2$, starting values $(x_0, y_0) = (0, 1)$, and four values of h: $h = 1, 0.5, 0.25, 0.125$

a derivation of Euler's method

easy to derive it from the direction field of $\frac{dy}{dx} = f(x, y)$, as follows:

- suppose we are at a point (x_n, y_n)
  - this might be the initial point (x_0, y_0)
- the slope is $m = f(x_n, y_n)$ so the line we want is

$$y - y_n = f(x_n, y_n)(x - x_n)$$

- we want to move to a new location $x_{n+1} = x_n + h$ so $x - x_n = h$ and $y = y_{n+1}$
- thus

$$y_{n+1} - y_n = f(x_n, y_n)\, h$$

- i.e. $y_{n+1} = y_n + h\, f(x_n, y_n)$

measuring accuracy

- assume we are solving an ODE IVP: $\frac{dy}{dx} = f(x, y)$, $y(x_0) = y_0$
- if we also know the exact solution $y(x)$ then we can measure (evaluate) the error in the approximation

$$y_n \approx y(x_n)$$

  - "y_n" is the number produced by Euler's method
  - "$y(x_n)$" is the exact solution at the x-value x_n
- picture this:

measuring accuracy

- context:

$$\frac{dy}{dx} = f(x, y), \qquad y(x_0) = y_0$$
$$y(x) = \text{(exact solution, a formula)}$$
$$y_n = \text{(numbers you compute from Euler's method)}$$

- there are two common ways to report the error:

1. *absolute error* $= |y(x_n) - y_n|$
2. *relative error* $= \dfrac{|y(x_n) - y_n|}{|y(x_n)|}$

caveat about measuring accuracy

- you can only compute absolute or relative error *if* the exact solution is known
- . . . but usually the reason we use a numerical method like Euler's is because the exact solution is *not* known
    - in real applications we do not know the exact solution
- thus: examples where the absolute or relative error is computable are automatically "toy examples"

example 2

- Example 2: for the ODE IVP

$$y' = y, \quad y(0) = 1$$

(a) find the exact solution
(b) use Euler's method to get an approximation of $y(1)$ which is accurate to four digits?
   - try $h = 0.1$ first, and then $h = 0.05$?

(c) show in a table: x_n, y_n, the exact value $y(x_n)$, the absolute error, and the relative error

example 2, cont.

- so one can proceed by hand, but its tedious work . . .
- this is the original purpose for which computers were designed
- I used Matlab code below
  - posted as `simpleeuler.m` at the Codes tab on the public site

```
h = 0.1;                  % change to e.g. h=0.05
N = 10;                   % change to e.g. N=20
x = 0;
y = 1;
for n = 1:N+1
    exact = exp(x);
    [x, y, exact, abs(y-exact), 100*abs(y-exact)/abs(exact)]
    y = y + h * y;        % this is Euler's method
    x = x + h;
end
```

example 2, cont. cont.

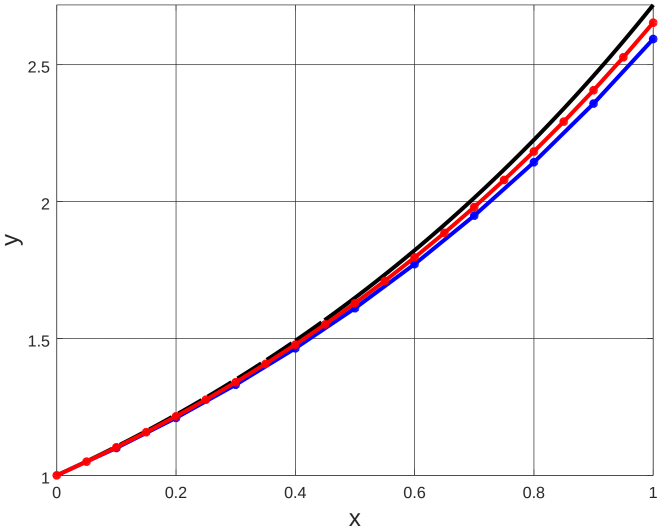
- the code produces the table below when $h = 0.1$ and we take $N = 10$ steps ... giving 4.58% relative error at $x = 1$

x_n	y_n	actual value	abs. error	rel. error
0.00	1.0000	1.0000	0.0000	0.00
0.10	1.1000	1.1052	0.0052	0.47
0.20	1.2100	1.2214	0.0114	0.93
0.30	1.3310	1.3499	0.0189	1.40
0.40	1.4641	1.4918	0.0277	1.86
0.50	1.6105	1.6487	0.0382	2.32
0.60	1.7716	1.8221	0.0506	2.77
0.70	1.9487	2.0138	0.0650	3.23
0.80	2.1436	2.2255	0.0820	3.68
0.90	2.3579	2.4596	0.1017	4.13
1.00	2.5937	2.7183	0.1245	4.58

example 2, cont.[3]; $h = 0.05$, $N = 20$ case

x_n	y_n	actual value	abs. error	rel. error
0.00	1.0000	1.0000	0.0000	0.00
0.05	1.0500	1.0513	0.0013	0.12
0.10	1.1025	1.1052	0.0027	0.24
0.15	1.1576	1.1618	0.0042	0.36
0.20	1.2155	1.2214	0.0059	0.48
0.25	1.2763	1.2840	0.0077	0.60
0.30	1.3401	1.3499	0.0098	0.72
0.35	1.4071	1.4191	0.0120	0.84
0.40	1.4775	1.4918	0.0144	0.96
0.45	1.5513	1.5683	0.0170	1.08
0.50	1.6289	1.6487	0.0198	1.20
0.55	1.7103	1.7333	0.0229	1.32
0.60	1.7959	1.8221	0.0263	1.44
0.65	1.8856	1.9155	0.0299	1.56
0.70	1.9799	2.0138	0.0338	1.68
0.75	2.0789	2.1170	0.0381	1.80
0.80	2.1829	2.2255	0.0427	1.92
0.85	2.2920	2.3396	0.0476	2.04
0.90	2.4066	2.4596	0.0530	2.15
0.95	2.5270	2.5857	0.0588	2.27
1.00	2.6533	2.7183	0.0650	2.39

example 2, cont.[4]



- for $h = 0.001$ and $N = 1000$ I get 0.05% rel. error:

$$y_{1000} = 2.71692 \approx 2.71828 = y(1)$$

another derivation of Euler's method

- start with the DE
$$\frac{dy}{dx} = f(x, y)$$
- remember what a derivative is!:
$$\lim_{h \to 0} \frac{y(x+h) - y(x)}{h} = f(x, y(x))$$
- think: $y(x)$ is current value and $y(x+h)$ is next value
- drop the limit and adopt this as a method:
$$\frac{y_{n+1} - y_n}{h} = f(x_n, y_n)$$
  - at this point y_n and $y(x_n)$ mean different things!
- rewrite as Euler's method before: $y_{n+1} = y_n + hf(x_n, y_n)$

are there better methods?

- yes!
- here is a derivation, by picture, of the "explicit midpoint rule":

are there better methods?

- yes!
- Euler's method is only first order; it makes errors proportional to the step size h
- see the Wikipedia page for "midpoint method"
  - the explicit and implicit midpoint methods are "second order"
  - they can get the same accuracy in 10 steps that Euler does in 100 steps
- see the Wikipedia page for "Runge-Kutta methods"
  - the original RK method is "fourth order" so it can get the same accuracy in 10 steps that Euler does in 10000 steps
  - we will return to this in Chapter 9

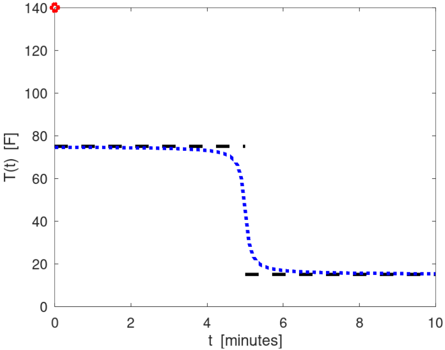
example 3

- Example 3: consider the ODE IVP

$$\frac{dT}{dt} = -0.3\,(T - T_m(t))\,, \qquad T(0) = 140$$

where $$T_m(t) = 75 - \left(\frac{60}{\pi}\right)\left(\frac{\pi}{2} + \arctan(10(t-5))\right)$$

- what situation does this model?

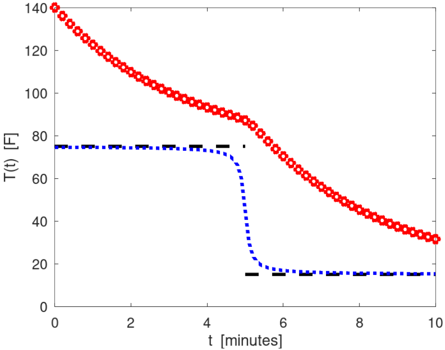


example 3: Newton's law of cooling

- consider: ODE IVP $\frac{dT}{dt} = -0.3\,(T - T_m(t))$, $T(0) = 140$, where $T_m(t)$ is a step down from 75 to 15 at $t = 5$
- approximate solution using Euler's method on $t \in [0, 10]$:

$$T_{n+1} = T_n + h\,[-0.3\,(T - T_m(t_n))]$$

- try $N = 50$ steps of length $h = 10/50 = 0.2$ minutes?



expectations

- to learn this material, just listening to a lecture is *not* enough!
- also:
  - *read* section 2.6 in the textbook
  - *try out* the Euler's method codes at the Codes tab
  - *do* Homework 2.6