Applied differential equations

TW244 - Lecture 05

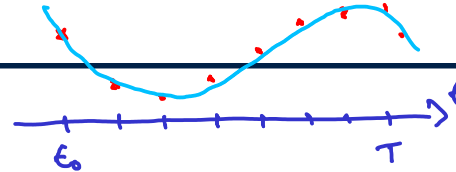
1st-order DEs: Numerical methods

Prof Nick Hale - 2020

SCIENCE
NATUURWETENSKAPPE
EYOBUNZULULWAZI

Numerical methods

2.6 Numerical methods
Approximate solutions



Consider the first-order initial value problem (IVP)

$$\frac{dy}{dt} = f(t, y) \qquad \text{with} \qquad y(t_0) = y_0.$$

We're interested in a solution $y = y(t)$ for $t \geq t_0$.

What to do if we cannot find a function analytically (i.e., in closed form)?

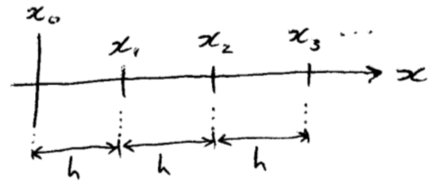
We can approximate it numerically!

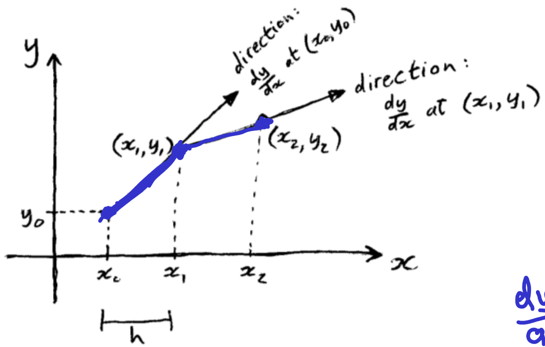
This gives an approximation to the solution, but this is often good enough.

2.6 Numerical methods

Euler's method (pronounced "Oiler", no "Yuler"!)

Divide the x-axis in to equally-spaced subintervals and let $y_i \approx y(t_i)$.





Calculate $\frac{dy}{dt}$ at (t_0, y_0) and step in that direction to (t_1, y_1).

Repeat at (t_1, y_1), etc.

This is the basis of Euler's method.

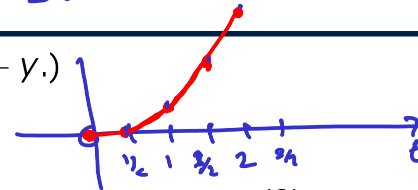
$$\frac{dy}{dt} = \lim_{h\to 0} \frac{y(t+h) - y(t)}{h}$$

In general we want y_n such that $\frac{y_{n+1}-y_n}{h} = f(t_n, y_n)$ $\therefore$ $y_{n+1} = y_n + hf(t_n, y_n)$.

Euler's method: Example

$y_{n+1} = y_n + hf(t_n, y_n)$

Example: $\frac{dy}{dt} = t - y$ with $y(0) = 0$. (i.e., $f(t, y) = t - y$.)



Let's choose $h = \frac{1}{2}$, then:

$$
\begin{aligned}
y_0 &= 0 & & & & & y(0) \\
y_1 &= y_0 + hf(t_0, y_0) &&= 0 + \tfrac{1}{2}(0 - 0) &&= 0 &\approx\ & y(\tfrac{1}{2}) \\
y_2 &= y_1 + hf(t_1, y_1) &&= 0 + \tfrac{1}{2}(\tfrac{1}{2} - 0) &&= 0.25 &\approx\ & y(1) \\
y_3 &= y_2 + hf(t_2, y_2) &&= 0.25 + \tfrac{1}{2}(1 - 0.25) &&= 0.625 &\approx\ & y(\tfrac{3}{2}) \\
y_4 &= y_3 + hf(t_3, y_3) &&= 0.625 + \tfrac{1}{2}(\tfrac{3}{2} - 0.625) &&= 1.0625 &\approx\ & y(2) \\
&\vdots
\end{aligned}
$$

Check: the exact solution of this IVP DE is $y = t - 1 + e^{-t}$.

Exact: $y(2) = 2 - 1 + e^{-2} = 1.1353$

Numerical: $y(2) \approx y_4 = 1.0625$.

These calculations are tedious to do by hand...

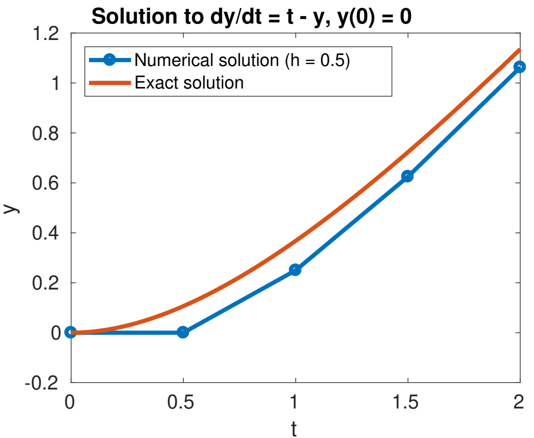
but computers are ideal for performing such computations!

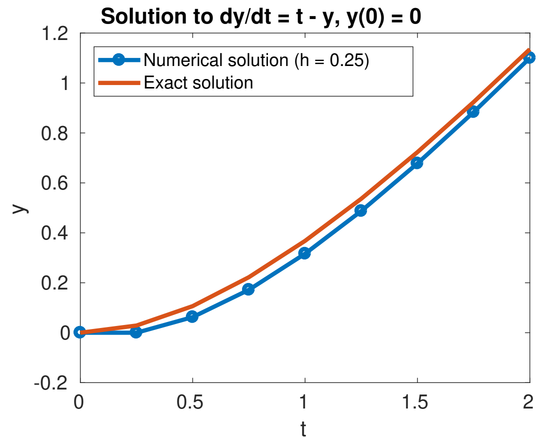
MATLAB text file for the implementation of Euler

```
t = []; y = [];                     % Initialise t and y
f = @(t,y) t - y;                   % Define rhs
h = .5;                             % Step size
a = 0; b = 2;                       % Solution interval
n = (b-a)/h;                        % Number of steps
t(1) = 0; y(1) = 0;                 % Initial values x0 and y0.
% WARNING! The (1) in the above means THE FIRST ENTRY IN THE VECTOR
% not necessarily THE VALUE AT TIME 1. Also note that since MATLAB
% starts counting from 1 (not 0) so x(1) = x0 and y(1) = y0.

% The main loop:
for i = 1:n
    y(i+1) = y(i) + h*f(t(i),y(i)); % Euler formula
    t(i+1) = t(i) + h;              % Update t
end
plot(t, y, '-o', 'LineWidth', 3);   % Plot circles. Thick line.
axis([0 2 -.2 1.2])                 % Set the axis limits [xmin, xmax, ymin, ymax]

% Plot the exact solution:
t = linspace(0, 2, 100);            % 100 equally spaced points in [0,2]
ytrue = t - 1 + exp(-t);            % Note exp(t) NOT e^t or exp(1)^t !
hold on                             % Use hold on to plot multiple lines
plot(t, ytrue, 'LineWidth', 3)      % Plot just thick line.
hold off                            % Remember to turn hold off!

% Add a legend and title, etc
xlabel('t'), ylabel('y')            % Axis labels
title('Solution to dy/dt = t - y, y(0) = 0')
legend(['Numerical solution (h = ' num2str(h) ')'], 'Exact solution', 'Location', 'NW')
set(gca, 'FontSize', 14)            % Change the font size on the figure
```

Euler's method: Smaller step sizes

We can (usually*) improve the accuracy in our approximations by choosing a smaller step size, h, but at the cost of more calculations.





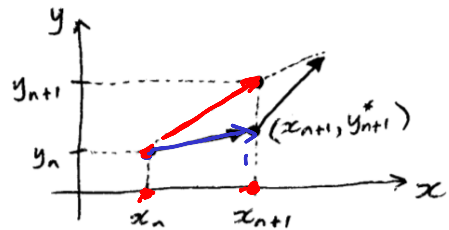
*Take TW324 next year to find out more details.

2.6 Numerical methods

Explicit Trapezium

Modified Euler (or "improved Euler"). See p. 365 of textbook.

1. Calculate $\frac{dy}{dt} = \underbrace{f(t_n, y_n)}_{\text{slope } A}$ and step to y^*_{n+1}.
2. Calculate $\frac{dy}{dt} = \underbrace{f(t_{n+1}, y^*_{n+1})}_{\text{slope } B}$
3. Take an average of the two slopes: slope $C = (A + B)/2$
4. Step from y_n to y_{n+1} using slope C.



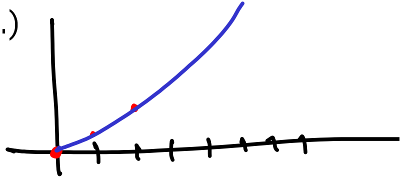
This gives the modified Euler scheme:

$$\begin{aligned} y^*_{n+1} &= y_n + hf(t_n, y_n) \\ y_{n+1} &= y_n + \tfrac{1}{2}h(f(t_n, y_n) + f(t_{n+1}, y^*_{n+1})) \end{aligned}$$

This is also known as the 'explicit trapezium rule' (see TW324).

Modified Euler: Example

Example: $\frac{dy}{dt} = t - y$ with $y(0) = 0$. (i.e., $f(t, y) = t - y$.)



Let's choose $h = \frac{1}{2}$, then:

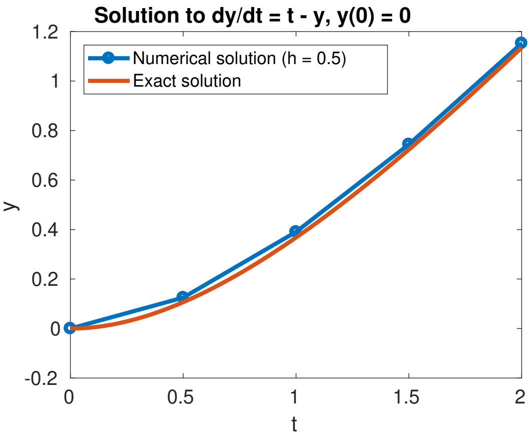
$$
\begin{aligned}
y_0 &= 0 \\
y_1^* &= y_0 + hf(t_0, y_0) && = 0 + \tfrac{1}{2}(0 - 0) = 0 \\
y_1 &= y_0 + \tfrac{h}{2}\left[f(t_0, y_0) + f(t_1, y_1^*)\right] && = 0 + \tfrac{1}{4}[(0 - 0) + (\tfrac{1}{2} - 0)] = 0.125 \\
y_2^* &= y_1 + hf(t_1, y_1) && = 0.125 + \tfrac{1}{2}(\tfrac{1}{2} - 0.125) = 0.3125 \\
y_2 &= y_1 + \tfrac{h}{2}\left[f(t_1, y_1) + f(t_2, y_2^*)\right] && = 0.125 + \tfrac{1}{4}[(\tfrac{1}{2} - 0.125) + (1 - 0.3125)] \\
& && = 0.390625 \\
&\vdots \\
y_4 &= 1.152588\ldots \qquad \approx y(2)
\end{aligned}
$$

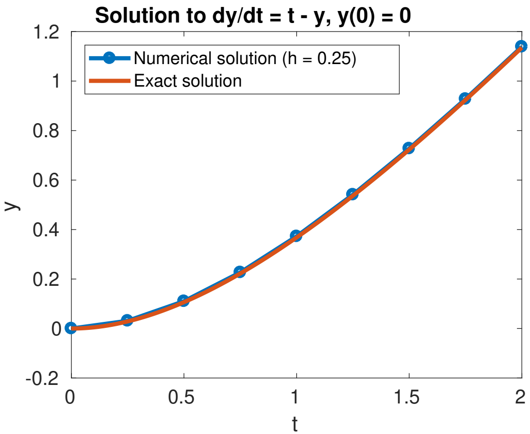
Recall the exact solution of this IVP DE is $y = t - 1 + e^{-t}$ with

$$y(2) = 2 - 1 + e^{-2} = 1.1353$$

Modified Euler: Example

Again, we can (usually) take h smaller to improve accuracy:





Exercise: Implement the modified Euler code in MATLAB (or Python).

- Whilst it is useful to understand how numerical methods such as the two above work and to have some experience in coding them yourself, in practice one typically uses general purpose software for the task. For example, later in the course we will be using MATLAB's `ode45` function.
- Numerical methods for the solution of DEs (and the computation of integrals and many other topics) are covered in more detail in the `TW324` course next semester:

  http://appliedmaths.sun.ac.za/TW324/

- Question for consideration: We argued that some DEs cannot be solved analytically, which is why we need numerical methods. But if all problems can be solved numerically, do we need analytical solutions?