Q1: In the absence of analytic tools, how can one "see" a solution to an IVP?

Q2: How do computer solution and ODEs?

Model:
\[ \frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \]

where both \( f, y \) are cont "around" \((t_0, y_0) \in \mathbb{R}^2\).

- **New** \((t_0, y_0)\) soln is unique.
- **Slope field** about at \((t_0, y_0)\)
  - gives us \( y'(t) \) for soln \( y(t) \) not solving \( y(t_0) = y_0 \).
  - eqn of line tangent to \( y(t) \)
    - at \((t_0, y_0)\) is
    \[ y - y_0 = y'(t_0)(t - t_0) \text{ or } \]
\[ y = y_0 + y'(t_0)(t - t_0) \]

like in Calc I, if we keep \( t \) close to \( t_0 \), then this tangent line is "good" approx to the actual solution \( y(t) \).
Idea: 1. Follow first line for a short time:

Let $t_1 = t_0 + h$ for small $h$.

Then $y_1 = y_0 + y'(t_0)(t_1 - t_0) \quad \text{call this } (x_1, y_1)$

$= y_0 + f(t_0, y_0)\; h$

2. Call the short line segment or do the line from $(t_0, y_0)$ to $(t_1, y_1)$ your approximated soln.

3. Take $(t_1, y_1)$ as your new starting point and repeat 1 & 2 to generate $(t_2, y_2)$:

$t_2 = t_1 + h$

$y_2 = y_1 + f(t_1, y_1)\; h$

Notes:

① Doing this for a small range of $t$ will not produce a solution, but will produce something "near" the actual solution passing through $(t_0, y_0)$.

② The amount one is "off" on the $y$ is called error, and is measurable.

③ This kind of computation is called Euler for computing
Euler Method

Given $y' = f(t, y)$, $y(t_0) = y_0$, with unique solutions,

one can approximate $y(t)$ on $[a, b]$, where $t_0 = a$

\[ t = t_0, t_1, \ldots, t_{n-1}, t_n = b, \quad t_i - t_{i-1} = h \]

1. Partition $[a, b]$ into step size $h$
2. Evaluate $y_{i+1} = y_i + h \cdot f(t_i, y_i)$
3. Connect points $(t_i, y_i)$ by straight lines.

Q: What are the issues of the Euler Method?

1. Step size is too large, approximate solution is far away from real one.
2. Step size is too small, computation is expensive
3. Computers cannot think (yet)....
- Computers cannot "see" asymptotes:
  ex. Graph \( f(x) = \frac{1}{x-2} \)

- Graph \( y = \ln x \)

- Draw phase portrait of \( y' = -\frac{1}{y}, \quad y(0) = -3 \)
  
  1. What is \( y \)-intercept?
  2. Find \( t \) where \( y(t) = 1 \) (Hint: see 2 of them)

  3. See ODE phase portraits of this solution for different solution methods.
Near where the vector field is very steep, (almost vertical), strange things can happen. This is because the step size horizontally may be small, but vertically may be great!

See JODE.
How to generalize the Euler Method

Euler Method: \( y_{n+1} = y_n + h \cdot f(t_n, y_n) \)

Integrating \( f(t, y(t)) \) over the interval \([t_n, t_{n+1}]\) would yield \( y(t_{n+1}) \) precisely, since \( y' = f(t, y(t)) \), and

\[
y(t) = y(t_n) + \int_{t_n}^{t} f(s, y(s)) \, ds
\]

By FTC.

Instead, we estimate the integral:

\[
y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s)) \, ds \approx y(t_n) + \frac{f(t_n, y(t_n)) \cdot (t_{n+1} - t_n)}{h}
\]

Improving the Euler Method involves only finding better estimates of the integral.
Instead, estimate $y_{n+1}$ via a standard Euler Bound:

$$y_{n+1} = y_n + \frac{h}{2} \left( f(t_n, y_n) + f(t_{n+1}, y_n + h f(t_n, y_n)) \right)$$

The Runge–Kutta methods are simply better, more computationally expensive versions of this estimation process.
Slope Field Applet.

eqn #1: \( \frac{dy}{dx} = -\frac{x}{y} \)

Min. x: \(-\pi^4\)  Max. x: \(\pi^4\)
Min. y: \(-10\)  Max. y: 10
Num. of segs: x 20  y 20

Show: Slope, Solution, Line, Pol, Tilt, Ax
Init. Cond. Euler
Step: 0.1
Add init. cond.: x 0.0  y 0.0

Last error: 

Submit | Show Table | Clear All
eqn #1: \( \frac{dy}{dx} = -\frac{x}{y} \)

Min. x: -\( \pi \cdot 4 \)  
Max. x: \( \pi \cdot 4 \)

Min. y: -10  
Max. y: 10

Num. of segs: x 20  
y 20

Show: Slope, Solution, Line, Poly, Tit, Asr  
Init. Cond.  
Mod. Eu  
Step: 0.1

Add init. cond. x: -12.0078652  
y: 8.85496183

Last error:  

Show All Errors, Print, Frame