Physics through Computational Thinking

Dynamics through Numerical Methods

Auditya Sharma and Ambar Jain
Dept. of Physics, IISER Bhopal

Outline

In this lecture you will

1. solve 2nd order differential equation for damped oscillation using Euler's method

Euler's Method

- For the differential equation of the type

$$\dot{x}(t) = f(x, t) \qquad x(t_0) = x_0 \tag{1}$$

- we implemented Euler's method using the **Module/function** below

```
euler[func_, xi_, ti_, tf_, nMax_] := Module[{h, datalist, prev},
  h = (tf - ti) / nMax // N;
  For[datalist = {{ti, xi}},
   Length[datalist] ≤ nMax,
   AppendTo[datalist, prev + {h, h func @@ prev}],
   prev = Last[datalist];
  ];
  Return[datalist];
 ]
```

- where, the input arguments are:
  - **func**: a function of t, x. $\dot{x} = \text{func}[t, x]$
  - **ti**: initial time or start time for computation
  - **tf**: final time or end time for computation
  - **xi**: initial value of x at $t = t_i$.
  - **nMax**: number of time interval (or step size).
- The expected **local error** in computation is of the order of h^2: $O(h^2)$
- The expected **global error** in the computation is of the order of h: $O(h)$
- We implemented the following **err** function to compute the mean global error given by equation

$$\text{err} = \frac{1}{N} \sum_{i=1}^{N} |x_i - F(t_i)| \tag{2}$$

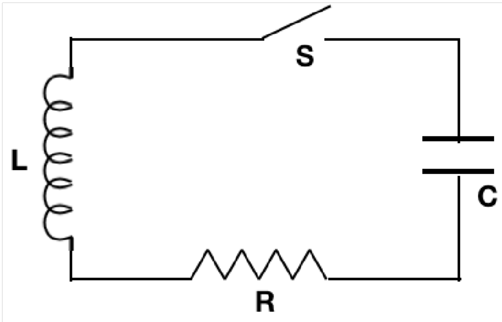
```
err[dataset_, func_] := Module[{tlist, xlist, Fxlist},
  tlist = dataset[[ ;; , 1]];       (*Extract each time value*)
  xlist = dataset[[ ;; , 2]];       (*Extract each x value*)
  Fxlist = func /@ tlist;           (*Apply func to each time value to get list of func[ti]*)
  Return[xlist - Fxlist // Abs // Mean];
 ]
```

- Calculate the mean global error for various value of h for fixed t_i and t_f

```
In[]:= f[t_, x_] = -x t;
       ff[t_] = e^(-t^2/2);
       Table[{5.0/10^n, err[euler[f, 1, 0, 5, 10^n], ff]/(5.0/10^n)}, {n, 1, 4}]

Out[]= {{0.5, 0.0746396}, {0.05, 0.0639748}, {0.005, 0.0631697}, {0.0005, 0.0630899}}

In[]:= err[euler[f, 1, 0, 5, 50], ff]

Out[]= 0.00648814
```

- We see that the global error scales at h, thus the error is considered as $O(h)$.

Reducing Higher Order ODE to a set of first order ODEs



- The equations we are usually interested in solving are equation of motion, which are usually second order differential equation, such as

$$\frac{d^2 Q}{d t^2} = -\frac{Q}{L C} - \frac{R}{L}\frac{d Q}{d t} \tag{3}$$

- associated with a suitable initial value condition like

$$\begin{aligned} Q(0) &= Q_0 \\ \dot{Q}(0) &= 0 \end{aligned} \tag{4}$$

- In order to apply a suitable numerical method, such as Euler’s method we will turn this IVP into a set of coupled first order ODEs:

$$\begin{aligned} \frac{d Q}{d t} &= I \\ \frac{d I}{d t} &= -\frac{Q}{L C} - \frac{R}{L} I \end{aligned} \tag{5}$$

- Now this is set of two coupled ODEs of order 1, in terms of dynamical quantities Q and I, with initial condition:

$$\begin{aligned} Q(0) &= Q_0 \\ I(0) &= 0 \end{aligned} \tag{6}$$

- We need to solve them simultaneously using Euler’s method.

Generalization

- Any n^{th}order ODE can be converted to coupled n first order ODE. Let's say ODE has the form:

$$\text{ODE}: \quad \frac{d^n x}{d t^n} = f\left(t, x, x^{(1)}, x^{(2)}, \ldots, x^{(n-1)}\right) \tag{7}$$

- then, we can define

$$\begin{gathered} y = \dot{x} \\ z = \dot{y} \\ \vdots \\ \dot{s} = w \\ \dot{w} = f(t, x, y, z, \ldots, s, w) \end{gathered} \tag{8}$$

- which is a set of coupled ODEs

Define and Translate LCR circuit eqn for Numerical Evaluation

- Remember the four steps of Computational Thinking: **Define**, **Translate**, **Compute** and **Interpret**
- In this problem we will define/identify and translate the problem into a form that it can be solved on the computer
- First step is to non-dimensionalize the equation of motion for LCR circuit, thus reducing the number of parameters in the problem
- Second step is to write the second order ODE into a set of first order ODEs (as shown in the previous section)

Problem: For the IVP given by equation below

$$\frac{d^2 Q}{dt^2} = -\frac{Q}{LC} - \frac{R}{L}\frac{dQ}{dt}$$
$$Q(0) = Q_0 \qquad (9)$$
$$\dot{Q}(0) = 0$$

(a) Non-dimensionalize the equation by choosing suitable scales for Q, I and t, expressing the equation in dimensionless quantities.

(b) How many free parameters are left in the equation after non-dimensionalization?

(c) Write the non-dimensionalized equation as a set of first order ODEs.

(d) Express the solution of this equation in terms of your dimensionless quantities:

$$Q(t) = Q_0\, e^{\frac{-R}{2L}t} \cos\left(\sqrt{\frac{1}{LC} - \frac{R^2}{4L^2}}\; t\right) \qquad (10)$$

Solution

$$Q \text{ scale}: \quad Q_0$$
$$t \text{ scale}: \quad L/R \qquad (11)$$
$$I \text{ scale}: \quad Q_0 R/L$$

Making the transformation:

$$\begin{aligned} Q &\longrightarrow Q_0 Q \\ t &\longrightarrow \frac{L}{R} t \\ I &\longrightarrow \frac{Q_0 R}{L} I \end{aligned} \tag{12}$$

we get

$$\begin{gathered} \frac{Q_0}{(L/R)^2} \frac{d^2 Q}{d t^2} = -\frac{Q_0 Q}{L C} - \frac{R}{L} \frac{Q_0}{L/R} \frac{d Q}{d t} \\ \Rightarrow \frac{d^2 Q}{d t^2} = -\frac{L}{R^2 C} Q - \frac{d Q}{d t} \end{gathered} \tag{13}$$

In the equation above, Q and t are dimensionless. Therefore, $I = dQ/dt$ is also dimensionless. After non-dimensionalization, there is only one free parameter $L/(R^2 C)$ which is the ratio of two competing time scales in the problem.
Writing as first order ODE we get

$$\begin{gathered} \frac{d Q}{d t} = I \\ \frac{d I}{d t} = -\frac{L}{R^2 C} Q - I \\ Q(0) = 1 \\ I(0) = 0 \end{gathered} \tag{14}$$

The solution, in terms of dimensionless Q and t can be written as:

$$Q(t) = e^{-t/2} \cos\left(\sqrt{\frac{L}{R^2 C} - \frac{1}{4}}\ t\right) \tag{15}$$

As expected, solution also depends only on one parameter: $L/(R^2 C)$ which should be greater than $\frac{1}{4}$ for the solution to be valid.

Solving Coupled ODEs using Euler's Method

- Lets take the following set of coupled ODEs, which is most general case for ODE, as higher order ODEs can always be brought in similar form.

$$\begin{aligned}\dot{x} &= f(t, x, y, z)\\ \dot{y} &= g(t, x, y, z)\\ \dot{z} &= h(t, x, y, z)\end{aligned} \tag{16}$$

- Define a column vectors X and F as

$$X = \begin{pmatrix} t \\ x \\ y \\ z \end{pmatrix} \qquad F = \begin{pmatrix} 1 \\ f \\ g \\ h \end{pmatrix} \tag{17}$$

- Then the coupled ODEs can be written as

$$\dot{X} = F \tag{18}$$

- It is straightforward to generalize this to arbitrary dimensions as long as their is only one dynamical variable t.
- Euler's method is given by

$$\begin{aligned}X_0 &= X_{\text{initial}}\\ X_{n+1} &= X_n + h\,F(X_n)\end{aligned} \tag{19}$$

- Here is its implementation. Notice the use of **Through** function

```
Through[{f, g, h}[x]]
```

```
{f[x], g[x], h[x]}
```

```
Through[{f, g, h}@x]
```

```
{f[x], g[x], h[x]}
```

```
Through[{f, g, h} @@ {x, y}]

{f1[x, y], g[x, y], h[x, y]}

eulerGen[Func_, X0_, tf_, nMax_] := Module[{h, datalist, prev, next, rate},
  h = (tf - X0[[1]]) / nMax // N;
  For[datalist = {X0},
   Length[datalist] ≤ nMax,
   AppendTo[datalist, next],
   prev = Last[datalist];
   rate = Through[Func @@ prev];
   next = prev + h rate;
  ];
  Return[datalist];
 ]
```

Application to 2nd Order ODEs: LCR circuit

- We want to solve the IVP:

$$\begin{gathered}\frac{dQ}{dt} = I \\ \frac{dI}{dt} = -\frac{L}{R^2 C} Q - I \\ Q(0) = 1 \\ I(0) = 0\end{gathered} \tag{20}$$

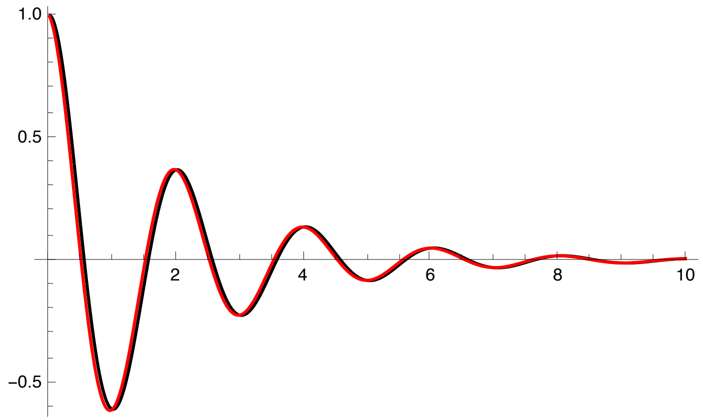
- Implementation: Lets take the ratio $w = L/(R^2 C)$

```
w = 10;
beta = Sqrt[w - 1/4];
2.0 π / beta
id[t_, charge_, current_] = 1;
chargeDot[t_, charge_, current_] = current;
currentDot[t_, charge_, current_] = -w charge - current;
initial = {0, 1, 0};
```

```
2.01223
```

```
data = eulerGen[{id, chargeDot, currentDot}, initial, 10, 10 000];
```

```
Show[ListPlot[data[[ ;; , 1 ;; 2]], Joined → True, PlotMarkers → None, PlotRange → Full],
 Plot[e^(-t/2) Cos[beta t], {t, 0, 10}, PlotRange → Full, PlotStyle → Red]]
```



```
Show[ListPlot[data[[ ;; , {1, 3}]], Joined → True, PlotMarkers → None, PlotRange → Full],
 Plot[-1/2 e^(-t/2) Cos[t beta] - e^(-t/2) beta Sin[t beta], {t, 0, 20}, PlotStyle → Red, PlotRange → Full]]
```