• The Case for Integrating Computation

Sir Martin Rees, Astronomer Royal at the 2017 APS April Meeting

https://www.facebook.com/apsphysics/videos/10155166016167952
Computational Basics

- First-Order Differential Equations
  - Simple Euler Method will get you pretty far

- Spreadsheet Implementations
1st Order Differential Equation

\[ f(t) = \frac{dN}{dt} \]

By definition,

\[ \lim_{\Delta t \to 0} \frac{N(t + \Delta t) - N(t)}{\Delta t} = \frac{dN}{dt} \]

\[ \frac{N(t + \Delta t) - N(t)}{\Delta t} \approx \frac{dN}{dt} \]

Rearrange to get:

\[ N(t + \Delta t) \approx N(t) + \frac{dN}{dt} \Delta t \]

“Simple Euler Method”
Nuclear Decay

\[ \frac{dN}{dt} = -\lambda N \]

**Exact Solution:**

\[ N(t) = N_0 e^{-\lambda t} \]

**Euler Solution:**

\[ N(t + \Delta t) \approx N(t) + \frac{dN}{dt} \Delta t \]

\[ N(t + \Delta t) \approx N(t) - \lambda N(t) \Delta t \]

\[ N(t + \Delta t) \approx (1 - \lambda \Delta t) N(t) \]
Computational Solution via the “Euler Method”

By definition

\[ a[x(t), v(t), t] = \lim_{\Delta t \to 0} \frac{v(t + \Delta t) - v(t)}{\Delta t}, \]

If the instantaneous acceleration can be approximated by its average value

\[ a(t) \approx a_{avg} = \frac{\Delta v}{\Delta t} \]

\[ a(t) \approx \frac{v(t + \Delta t) - v(t)}{\Delta t} \]

\[ \Rightarrow v(t + \Delta t) \approx v(t) + a(t) \Delta t \]

Similarly,

\[ y(t + \Delta t) \approx y(t) + v(t) \Delta t \]
\[ v(t + \Delta t) \approx v(t) + a(t) \Delta t \]
\[ y(t + \Delta t) \approx y(t) + v(t) \Delta t \]

Diagram:
- Velocity vs. time graph
- Approximate \( v(t + \Delta t) \)
- Exact \( v(t + \Delta t) \)

Approximation:
\[ v_n = v_{n-1} + a_{n-1} \Delta t \]
\[ y_n = y_{n-1} + v_{n-1} \Delta t \]
Accumulation of Local Error in the Euler Method

velocity

accumulated Local Error over n time steps = “Global Error”

\[ v(t + \Delta t) \approx v(t) + a(t)\Delta t \]

\[ y(t + \Delta t) \approx y(t) + v(t)\Delta t \]

“Exact” \( v(t+\Delta t) \)

“Local Error”

Approximate \( v(t+\Delta t) \)
Recall the index notation:

\[ t_n = n\Delta t \]
\[ v(t) \rightarrow v_n \]
\[ y(t) \rightarrow y_n \]
\[ v(t + \Delta t) \rightarrow v_{n+1} \]
\[ y(t + \Delta t) \rightarrow y_{n+1} \]

Now, in general:

\[ a(t) = a[y(t), v(t), t] \]
\[ a_n = a[y_n, v_n, t_n] \]
Falling Sphere with Air Resistance

Apply Newton’s 2\textsuperscript{nd} Law:

\[ \sum F = F_g - F_r = ma \]

\[ F_r = \frac{D \rho A}{2} v^2 \quad \text{and} \quad F_g = mg \]

D = drag coefficient (0.5 for non-spinning sphere)  
\( \rho \) = air density  
A = cross-sectional area (in direction of motion)  
m = mass of sphere

\[ mg - \frac{D \rho A}{2} v^2 = ma \]

\[ a(t) = g - \frac{D \rho A}{2m} [v(t)]^2 \]
Integrate to get *Exact* Solution

\[
\frac{dv}{dt} = a(t) = g - \frac{DqA}{2m} [v(t)]^2
\]

\[
\Rightarrow \quad v(t) = \sqrt{\frac{2mg}{DqA}} \tanh \left( \sqrt{\frac{DqA g}{2m}} t \right) \quad \text{Exact}
\]

Integrate again to get

\[
y(t) = \frac{2m}{DqA} \ln \left[ \cosh \left( \sqrt{\frac{DqAg}{2m}} t \right) \right] \quad \text{Exact}
\]
Hanging Spring

Newton’s 2\textsuperscript{nd} law:

\[ \sum F = mg - k (y + y_{eq}) \]

\[ \Rightarrow \ddot{x}(t) = -\frac{k}{m}y(t) \]

Euler?

Computationally via Euler-Cromer Algorithm

\[ v(t + \Delta t) \approx v(t) + a(t) \Delta t \]

\[ y(t + \Delta t) \approx y(t) + v(t + \Delta t) \Delta t \]
Stable solutions using the Euler approximation

Alan Cromer

Physics Department, Northeastern University, Boston, Massachusetts 02115

(Received 14 July 1980; accepted 9 September 1980)

A minor modification of the standard Euler approximation for the solution of oscillatory problems in mechanics yields solutions that are stable for arbitrarily large number of iterations, regardless of the size of the iteration interval. The period of a nonlinear oscillator converges rapidly to its exact value as the size of the iteration interval is decreased. In two dimensions, closed orbits are given for the two-body Kepler problem and the restricted three-body problem can be iterated indefinitely to produce space-filling orbits. In this new approximation, the difference $\Delta E$ between the initial energy and the energy after $n$ iterations is bounded, oscillatory, and zero when averaged over half a cycle of the motion.

I. INTRODUCTION

The rapidly increasing availability of microcomputers and programmable calculators has stimulated interest in simple numerical methods that students can use to solve

$$
\begin{align*}
    v_{n+1} &= v_n + F_n T \\
    x_{n+1} &= x_n + (1/2)(v_n + v_{n+1}) T
\end{align*}
$$

This approximation seems more reasonable than the FPA, and in fact it does give exact results when $F$ is a constant.

We call this the first-point approximation because, in the equation for $x_{n+1}$, the velocity $v_n$ at the beginning of the iteration interval is used to estimate the average velocity during the interval.

The midpoint approximation (MPA) is

$$
\begin{align*}
    v_{n+1} &= v_n + F_n T \\
    x_{n+1} &= x_n + v_n T
\end{align*}
$$

We have errors that are bounded, oscillatory, and zero when averaged over half a period. The LPA yields stable solutions that can be iterated indefinitely. As a result, an undergraduate—and even a high school student—can obtain very accurate solutions to advanced mechanics problems.

The LPA was discovered quite by accident by Abby Aspel, a student at Newton North High School (Newton, MA). She was working on a computer program for the Kepler problem, and had written the correct program for

IV. CONCLUSIONS

The last point approximation is a simple but powerful method for solving problems of oscillatory motion. The proof of the approximation’s stability is simple enough to give to upperclass undergraduates, and the approximation itself is simple enough to be used by a high school student.

With such an elegant tool at our disposal, the three-body problem may become as much a part of the introductory physics curriculum as the inclined plane.