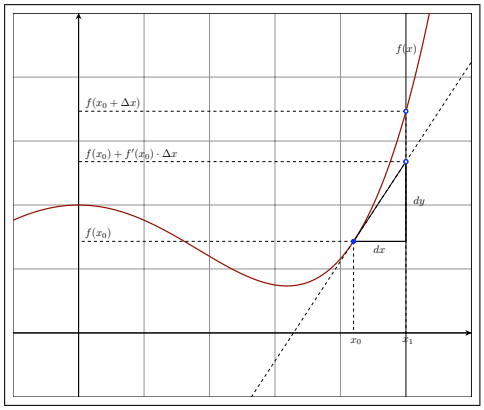
Project 4

Euler's Method and Sky Diving

Background

The derivative f' of a function f at $x = x_0$ gives the slope of the tangent line at the point $(x_0, f(x_0))$. One can use the tangent line to estimate values of f for values of x near x_0 and see whether the function is increasing or decreasing.



We observe that near $x = x_0$ the value of dy is a "good" approximation to the true value of Δy; i.e., the tangent line is "close" to the graph of f when x is close to x_0. Leibniz's notation for the derivative at a, $\frac{dy}{dx} = f'(a)$, was chosen to suggest that $dy = dx \cdot f'(a)$. So our new value $y + \Delta y$ is approximated by $y + dy = f(a) + dx \cdot f'(a)$. Certainly, the actual value of $y + \Delta y$ could be larger or smaller than our estimate—that depends on the function. The smaller the change in x, however, the smaller the error should be.

We have already used the local linearity of differentiable functions to develop Newton's method to approximate the zeroes of a function. A second, and historically important application, interpolation is used to extend the tables. If one knows $f(a)$ and

can determine $f'(a)$, then values of $f(a + \Delta x)$ can be estimated by:

$$f(a + \Delta x) \approx f(a) + \Delta x \cdot f'(a)$$

This technique is particularly valuable when values for f are difficult to calculate.

Local linearity is also used to solve initial value problems:

> Suppose we have an easy way to compute $f'(x)$. Suppose further we have one point, the initial value, that lies on the graph of $y = f(x)$. From this information we "reconstruct" f.

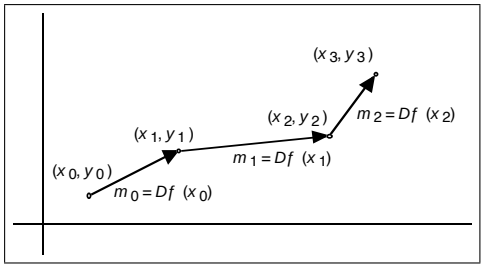
In this project, we adopt a frequently used strategy. We weaken our notion of what it is to reconstruct f. Instead of insisting on a formula for f, we produce a table of values $[x, f(x)]$. Typically, such data arrays contain anywhere from 20 to $20,000$ points (x_i, y_i).

Before starting to calculate, we need to decide on the size of the table. This decision is based on the needs of the end user of the data, the equipment and time available, and the mathematical nature of the problem. The size of the table determines the range of x-values and the total number of entries. Often, the x-values are equally spaced to simplify the formulas. Then the domain and the number of entries together determine Δx, the distance between successive x-values in the table.

We construct the table of values as follows:

> Our initial point is $(x_0, f(x_0))$. We calculate $f'(x_0)$. The tangent line to f is given by
>
> $$y = f'(x_0) \cdot (x - x_0) + f(x_0).$$
>
> Taking $x_1 = x_0 + \Delta x$ and substituting into the equation of the tangent line, we get the value
>
> $$y_1 = f'(x_0) \cdot (x_1 - x_0) + f(x_0)$$
>
> which simplifies to
>
> $$y_1 = f'(x_0) \cdot \Delta x + f(x_0).$$
>
> Knowing the equality $y_1 = f(x_1)$ is at best approximate, we accept the value and enter it in the table. Now start over from the point $(x_1, f(x_1))$. We next calculate $f'(x_1)$. The corresponding tangent line to f is given by
>
> $$y = f'(x_1) \cdot (x - x_1) + f(x_1).$$
>
> Taking $x_2 = x_1 + \Delta x$ and substituting into the equation of the tangent line, we get the value
>
> $$y_2 = f'(x_1) \cdot (x_2 - x_1) + f(x_1)$$
>
> which simplifies to
>
> $$y_2 = f'(x_1) \cdot \Delta x + f(x_1).$$
>
> Again we accept the equality $y_2 = f(x_2)$ for use in the table. To produce the next point, we calculate $f'(x_2)$ and move along the new tangent line to reach (x_3, y_3).

The process is continued until we have filled the table. Successively generating new values of f in a table by following the tangent lines is known as *Euler's method*. The diagram below illustrates the construction of the first three points in the table generated by Euler's method.

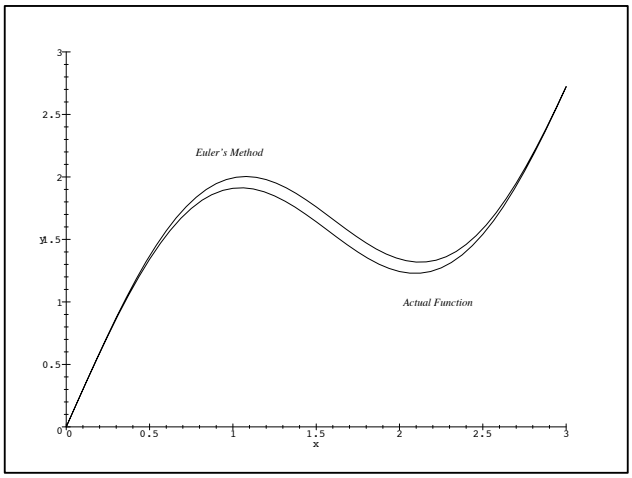


To give you some feel for the accuracy possible with Euler's method, the figure below shows the graph of the function f along with the graph of the table produced from the initial value problem:

$$f'(x) = 1 + 2\cos(2x)$$
$$(x_0, y_0) = (0, 0)$$

with the table size determined as:

Range of table:	x values from $x = 0$ to $x = 3$
Number of table entries:	51 (50 *new values* + *the initial point*)
Step size:	$\Delta x = \frac{3-0}{50} = 0.06$



Note in this example, on the interval where the function itself is concave down, Euler's method over-approximates the change in f. On the interval in which the function is concave up, Euler's method under-approximates the change in f. Over the entire interval $[0, 3]$, the two errors appear to cancel for this initial value problem.

Let's consider an initial value problem in detail:

Suppose $f'(x) = x^2 - y^2$ and $f(0.5) = 1$.
Construct a table for $f(x)$ over the interval from $x = 0.5$ to $x = 1.0$.

To begin our construction, we take $\Delta x = 0.1$; this will produce five new values and complete a table of 6 entries.

Question 1 *How does the choice of the value for Δx affect the size of the table generated?*

Set up the initial values and functions in Maple by entering :

```
> df := (x,y) -> x^2 - y^2;
```

$$df := (x, y) \to x^2 - y^2$$

```
> x[0] := 0.5;
> y[0] := 1.0;
```

$$x_0 := .5$$

$$y_0 := 1.0$$

```
> dx := 0.1;
```

$$dx := .1$$

Calculate the table entries by typing:

```
> x[1] := x[0] + dx;
> y[1] := y[0] + df(x[0],y[0])*dx;
```

$$x_1 := .6$$

$$y_1 := .925$$

```
> x[2] := x[1] + dx;
> y[2] := y[1] + df(x[1],y[1])*dx;
```

$$x_2 := .7$$

$$\mathrm{f}(\,.7\,) := .8754375$$

```
> x[3] := x[2] + dx;
> y[3] := y[2] + df(x[2],y[2])*dx;
```

$$x_3 := .8$$

$$y_3 := .8477984184$$

```
> x[4] := x[3] + dx;
> y[4] := y[3] + df(x[3],y[3])*dx;
```

$$x_4 := .9$$

$$y_4 := .8399222026$$

```
> x[5] := x[4] + dx;
> y[5] := y[4] + df(x[4],y[4])*dx;
```

$$x_5 := 1.0$$

$$y_5 := .8503752720$$

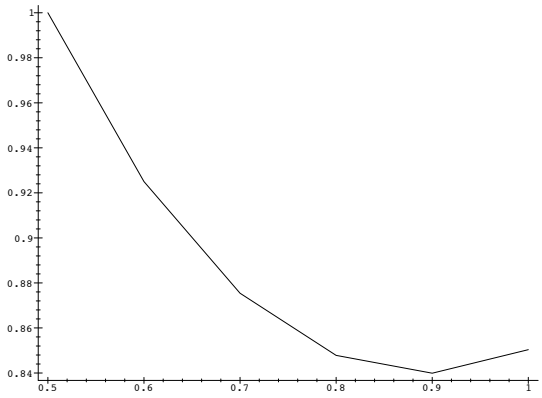
Tedious, wasn't it? To put our values in a table, we will use Maple's sequence command.

```
> Table := [seq([x[i], y[i]], i=0..5)];
```

$$Table := [[.5, 1.0], [.6, .925], [.7, .8754375], [.8, .8477984184], [.9, .8399222026], [1.0, .8503752720]]$$

`Table` can now be plotted[1]

```
> plot(Table);
```



Or printed nicely in a matrix.

```
> Matrix(6, 2, [Table]);
```

$$\begin{bmatrix} .5 & 1.0 \\ .6 & .925 \\ .7 & .8754375 \\ .8 & .8477984184 \\ .9 & .8399222026 \\ 1.0 & .8503752720 \end{bmatrix}$$

[1] Merging two images makes a very informative plot. Try:
`plots[display]({plot(Table), plot(Table, style=point, symbol=diamond)});`.

Clearly, in extending the table to $x = 2.5$ we could spend inordinate time doing arithmetic. We can automate Euler's method using the following Maple constructs.

Define the derivative function and set the initial values:

```
> df := (x,y) -> x^2 - y^2;
```

$$df := (x, y) \to x^2 - y^2$$

```
> x[0] := 0.5;
> y[0] := 1.0;
```

$$x_0 := .5$$

$$y_0 := 1.0$$

```
> dx := 0.1;
```

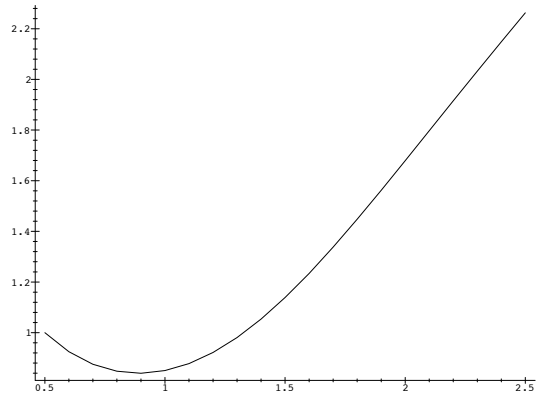
$$dx := .1$$

```
> for i from 0 to 19 do
    x[i+1] := x[i] + dx;
    y[i+1] := y[i] + df(x[i],y[i])*dx;
    end do:
> NewTable := [seq([x[i], y[i]], i=0..20)]:
    Matrix(21, 2, NewTable); [2]
```

$$\begin{bmatrix} .5 & 1.0 \\ .6 & .925 \\ .7 & .8754375 \\ .8 & .8477984184 \\ .9 & .8399222026 \\ 1.0 & .8503752720 \\ 1.1 & .8780614617 \\ 1.2 & .9219622687 \\ 1.3 & .9809608262 \\ 1.4 & 1.053732412 \\ 1.5 & 1.138697212 \\ 1.6 & 1.234034078 \\ 1.7 & 1.337750067 \\ 1.8 & 1.447792543 \\ 1.9 & 1.562182218 \\ 2.0 & 1.679140890 \\ 2.1 & 1.797189477 \\ 2.2 & 1.915200475 \\ 2.3 & 2.032401189 \\ 2.4 & 2.148335730 \\ 2.5 & 2.262801089 \end{bmatrix}$$

```
> plot(NewTable);
```

[2] Enter `interface(rtablesize=50)` to tell Maple to display large matrices.



We have avoided several possible problems. In using this procedure you must be careful that all of the initial values are entered as floating point numbers (they must contain a decimal point). Care must be taken in entering parentheses and brackets where required. Minor typing mistakes can result in warnings about "recursive definitions" — heed them unless you have a perverse desire to crash the machine and lose your work.

Project Report

The velocity of a sky diver is modeled using Newton's Second Law by the system:

$$\begin{aligned} v'(t) &= 9.8 - k \cdot [v(t)]^{1.1} && \textit{The units are acceleration, m/sec}^2 \\ v(0) &= v_0 && \textit{Initial velocity is in m/sec.} \end{aligned}$$

The first term in the derivative is the acceleration due to the gravity of the earth. The second term is due to air resistance — an important quantity to sky divers.

For your models, take the coefficient of resistance $k = 2.1$ and the initial velocities $v_0 = 0.0$ m/sec and $v_0 = 4.8$ m/sec. Use Euler's method to determine the velocity of the sky diver after 10 seconds. Produce at least three tables of different sizes. The first table should have 21 values.

Question 2 *Euler's method when used with a small number of points (which gives a large Δx) produces a jagged graph. What size table is needed before the graph "smooths"?*

Extension

1. Recall that velocity v is the derivative of displacement s. Use your reconstruction of v from one or more of your tables above to reconstruct $s(t)$, the distance fallen by the sky diver.

2. The `for` ... `from` ... `by` ... `to` ... `do` ... `od` structure used in Euler's method can be entered as a Maple procedure with the variable `n` replacing the 19. This has the effect of changing `dx` and creating larger or smaller tables. Remember, for each new table, `dx` must be recalculated.

3. Consider the effect of a parachute opening after 60 seconds.

Report Requirements

A minimal project report will include:

- English responses to Questions 1 and 2.
- The annotated Maple statements and calculations used to generate the velocity table.
- A plot of the generated points (x_i, y_i).
- A discussion of the graphs in relation to your physical intuition of sky diving.