Mathematics 255—January 18, 1996

Summary of numerical methods

In this course we shall work with three different methods of solving differential equations by numerical approximation. In choosing among them there is a trade-off between simplicity and efficiency. Euler's method is relatively simple to understand and to program, for example, but almost hopelessly inefficient. The third and final method we shall use, the order 4 method of Runge-Kutta, is very efficient, but rather difficult to understand and even to program. The improved Euler's method lies somewhere in between these two on both grounds.

Each one produces a sequence of approximations to the solution over a range of x-values, and proceeds from an approximation of y at one value of x to an approximation at the next in one more or less complicated step. In each method one step goes from x_n to $x_{n+1} = x_n + h$. The methods differ in how the step from y_n to y_{n+1} is performed. More efficient single steps come about at the cost of higher complexity for one step. But for a given desired accuracy, the overall savings in time is good.

Method	The step from (x_n, y_n) to (x_{n+1}, y_{n+1})	Overall error
Euler's	$s_* = f(x_n, y_n)$ $y_{n+1} = y_n + \Delta \cdot s_*$ $x_{n+1} = x_n + \Delta$	proportional to Δ
Improved Euler's (RK of order 2)	$s_* = f(x_n, y_n)$ $y_* = y_n + \Delta \cdot s_*$ $x_{n+1} = x_n + \Delta$ $s_{**} = f(x_{n+1}, y_*)$ $y_{n+1} = y_n + \frac{\Delta}{2}(s_* + s_{**})$	proportional to Δ^2
Runge-Kutta of order 4	$s_* = f(x_n, y_n)$ $y_* = y_n + \frac{\Delta}{2} s_*$ $x_* = x + \frac{\Delta}{2}$ $s_{**} = f(x_*, y_*)$ $y_{**} = y_n + \frac{\Delta}{2} s_{**}$ $s_{***} = f(x_*, y_{**})$ $x_{n+1} = x + \Delta$ $y_{***} = y_n + \Delta \cdot s_{***}$ $s_{****} = f(x_{n+1}, y_{***})$ $y_{n+1} = y_n + \Delta \left(\frac{s_* + 2s_{**} + 2s_{***} + s_{****}}{6} \right)$	proportional to Δ^4

The purpose of using $*$ here is that the numbers like y_* etc. are not of interest beyond the immediate calculation they are involved in.

In the case of the simplest kind of differential equation

$$y' = f(x)$$

these reduce to the **rectangle** method, the **trapezoidal** rule, and **Simpson's** rule for finding a numerical approximation to the integral

$$\int_{x_0}^{x_f} f(x)\,dx$$

When implementing these by hand, which one has to do every now and then, you should set the calculations out in tables, or on a spreadsheet, like this for the Improved Euler's method applied to the differential equation

$$y' = 2xy - 1$$

with step size $\Delta = 0.1$.

x	y	$s_* = f(x, y)$	$y_* = y + \Delta \cdot s_*$	$s_{**} = f(x + \Delta, y_*)$
0.000000	1.000000	-1.000000	0.900000	-0.820000
0.100000	0.909000	-0.818200	0.827180	-0.669128
0.200000	0.834634	-0.666147	0.768019	-0.539189

etc.

Of course this is redundant and tedious work, and ideally suited to a computer. Here is a simple Pascal program to implement Euler's method for the same differential equation.

```
program euler;

var x, y, xf, h:  real;
    i, N : integer;

function f(x, y:real):  real;

begin
  f := 2*x*y - 1;
end;

function next_y(x, y, h:  real):real;

begin
  next_y := y + h*f(x, y);
end;

begin
  write('Initial x?  y?  Final x?  N? ');
  read(x, y, xf, N);
  h := (xf - x)/N;
    writeln;
  for i := 1 to N do begin
    writeln(x:8:6, ' ', y:8:6);
    y := next_y(x, y, h);
    x := x + h;
  end;
  writeln(x:8:6, ' ', y:8:6);
end.
```