Math 114, solutions to Assignment 11

Isabel Lugo

November 21, 2007

These are the solutions to the tenth homework assignment.

1 Section 10.1, # 12.

We note that the answer can't be B, because in B we have $y' = -2xy$, so $y' = 0$ when $x = 0$. Thus any solution curve to $y' = -2xy$ crosses the y-axis horizontally, but it does not. Similarly, any solution to $y' = 1 + xy$ (which is equation A) has y' positive in the first quadrant, so any solution curve to this equation m ust be increasing in the first quadrant; our curve is not. By elimination, the answer is C. To check this, note that in the second quadrant $y' = 1 - 2xy$ is positive, so any solution curve to C is increasing in the second quadrant; furthermore, when $x = 0$ we have $y' = 1$. Our curve satisfies both of these.

2 Section 18.4, # 12.

Solve $x^2y'' + xy' + x^2y = 0, y(0) = 1, y'(0) = 0$ by using ower series.

Answer. Let $y = \sum_{n=0}^{\infty} c_n x^n$. Then we have

$$y' = \sum_{n=1}^{\infty} nc_n x^{n-1}, y'' = \sum_{n=2}^{\infty} n(n-1)c_n x^{n-2}.$$

Thus the left-hand side of the original differential equation is

$$x^2 \sum_{n=2}^{\infty} n(n-1)c_n x^{n-2} + x \sum_{n=1}^{\infty} nc_n x^{n-1} + x^2 \sum_{n=0}^{\infty} c_n x^n.$$

Pulling the powers of x inside the sums gives

$$\sum_{n=2}^{\infty} n(n-1)c_n x^n + \sum_{n=1}^{\infty} nc_n x^n + \sum_{n=0} c_n x^{n+2}.$$

Finally, we shift the index on the last sum to get

$$\sum_{n=2}^{\infty} n(n-1)c_n x^n + \sum_{n=1}^{\infty} nc_n x^n + \sum_{n=2} c_{n-2} x^n$$

and combine these all into a single sum. The second sum contributes a term $c_1 x$; all three sums contribute terms in $x^2, x^3, \ldots$. We have

$$c_1 x + \sum_{n=2}^{\infty} \left[n(n-1)c_n + nc_n + c_{n-2} \right] x^n = 0.$$

So $c_1 = 0$, and $n^2 c_n + c_{n-2} = 0$ for all n. We can rewrite this to give $c_n = -c_{n-2}/n^2$; since $c_1 = 0$ we get $c_3 = c_5 = c_7 = \cdots = 0$. For the even terms, we have

$$c_2 = \frac{-1}{2^2} c_0, c_4 = \frac{-1}{4^2} c_2 = \frac{1}{2^2 \cdot 4^2} c_0, \cdots$$

and in general

$$c_{2k} = \frac{(-1)^k}{(2 \cdot 4 \cdot \ldots \cdot 2k)^2} c_0 = \frac{(-1)^k}{4^k (k!)^2} c_0.$$

Plugging these into the expression for y gives

$$y = c_0 \left(\sum_{k=0}^{\infty} \frac{(-1)^k}{4^k (k!)^2} x^k \right)$$

and setting $y = 0$, all terms but the $k = 0$ term go away, and we get $c_0 = 1$.

3 Numerical estimation

The following Maple code does Euler's method (without the plotting that Dr. Pemantle's code incorporates):

```
f := (y,t) -> t*(1-t/(1+y^2));
eulermethod := proc(t0,tf,y0,h)
t := t0; y := y0;
while t < tf do
y := y + f(y,t)*h; t := t+h;
od;
RETURN(y); end;
eulermethod(0, 1, 1, 1/16);
```

The procedure `eulermethod` takes four inputs – the initial and final times, the initial y-value, and the step size, in that order. (We use the step size 1/16 in this example.) Successively halving the step size, we see:
with step size 1/2, we get $f(1) \approx 1.1875000000$
with step size 1/4, $f(1) \approx 1.277056828$
with step size 1/8, $f(1) \approx 1.321757917$
with step size 1/16, $f(1) \approx 1.344180781$
with step size 1/32, $f(1) \approx 1.355420458$
with step size 1/64, $f(1) \approx 1.361048558$
with step size 1/128, $f(1) \approx 1.363864832$

with step size $1/256$, $f(1) \approx 1.365273540$

with step size $1/512$, $f(1) \approx 1.365978046$

with step size $1/1024$, $f(1) \approx 1.366330318$ and we can see that this is approaching somewhere around 1.37, to two decimal places.

The actual value is about 1.3666826. One way to get this more exact value (which is beyond the scope of this course) is to note that the error in Euler's method is proportional to h. Let $f_n(1)$ be the result of the Euler method with step size 2^{-n}; thus we have $f_1(1) = 1.1875, f_2(1) = 1.2770\ldots$, and so on. Then we have

$$f_n(1) = f(1) + ah + bh^2 + \ldots$$

for some constants $a, b, \ldots$ and $h = 2^{-n}$. We thus have

$$f_n(1) \approx f(1) + a/2^n, f_{n-1}(1) \approx f(1) + a/2^{n-1}$$

and if we take these approximations as equalities, we see $2f_n(1) - f_{n-1}(1) = f(1)$. In reality, this is not true, but it's a more accurate approximation. Computing, $f_n(1) - f_{n-1}(1)$ for, say, $n = 8, 9, 10$ gives values around 1.3666826. (This is an example of the technique known as *Richardson extrapolation*, which is covered in numerical analysis courses.) In practice, the Euler method is not used for numerical computation, because there are other methods which are more accurate; the most typical such method is the *fourth-order Runge-Kutta method*.

(A solution to the Picard iteration problem may follow.)