NATIONAL UNIVERSITY OF SINGAPORE
Department of Mathematics

MA1506 Laboratory 2 (scilab)
Semester II 2010/2011

The aim of lab 2 is to demonstrate some tools available in scilab that help us to better understand solutions of differential equations. In Part A, we will graph direction fields and use them to visualize solutions to differential equations. In Part B, we shall learn how to use a numerical solver to approximate solutions to differential equations.

Part A: Direction Fields

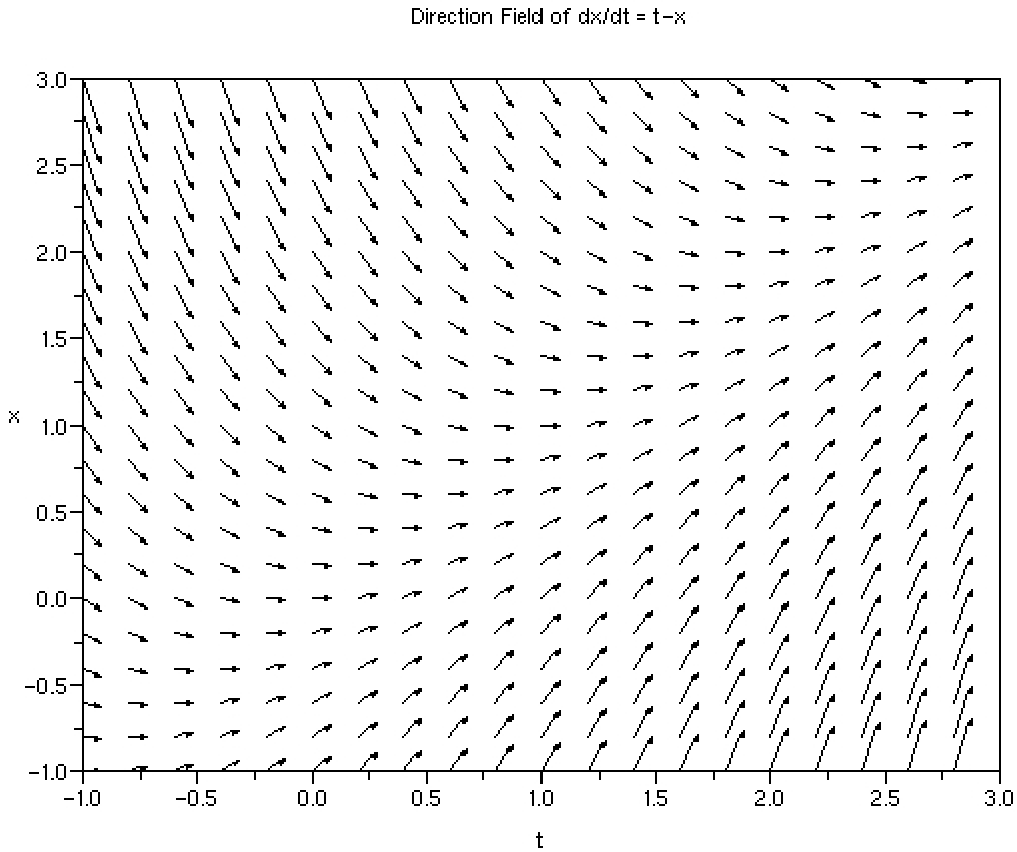
We should understand that it may not always be possible to solve any given first order differential equation. However, it is relatively easy to obtain the direction field of the differential equation and deduce qualitative information about the equation without solving it. The **direction field** (or slope field) of a first order differential equation of the form

$$\frac{dx}{dt} = f(t, x),$$

is obtained by drawing through each point in the (t, x) plane a short line segment with slope $x' = f(t, x)$.
For example, consider the equation $\frac{dx}{dt} = t - x$. To graph its direction field, we execute the following series of commands:

```
--> deff('[xdot]=f(a,x)', 'xdot=[1 ; x(1) - x(2)]')
--> fchamp(f, 0, -1:0.2:3, -1:0.2:3)
--> mtlb_axis ( [-1 3 -1 3]);
--> xlabel('t')
--> ylabel('x')
--> title( 'Direction Field of dx/dt = t - x')
```

Before we study the resulting graph, let us explain the commands line by line. The deff command creates a function f in the variables a and x. We shall not be using a but it needs to be there. The variable x actually has two components $x(1)$ and $x(2)$, which represents the horizontal and vertical axis respectively. In our case, $x(1) = t$ and $x(2) = x$. Hence you can see that we are creating a function $t - x$. The fchamp command creates a direction field for our function f. (The 0 parameter is related to a which is not used.) The remaining two parameters dictate the size of the field. In this case, a rectangular 21×21 grid of uniformly distributed points in the rectangle (actually a square) from -1 to 3 on both the horizontal and vertical axis.

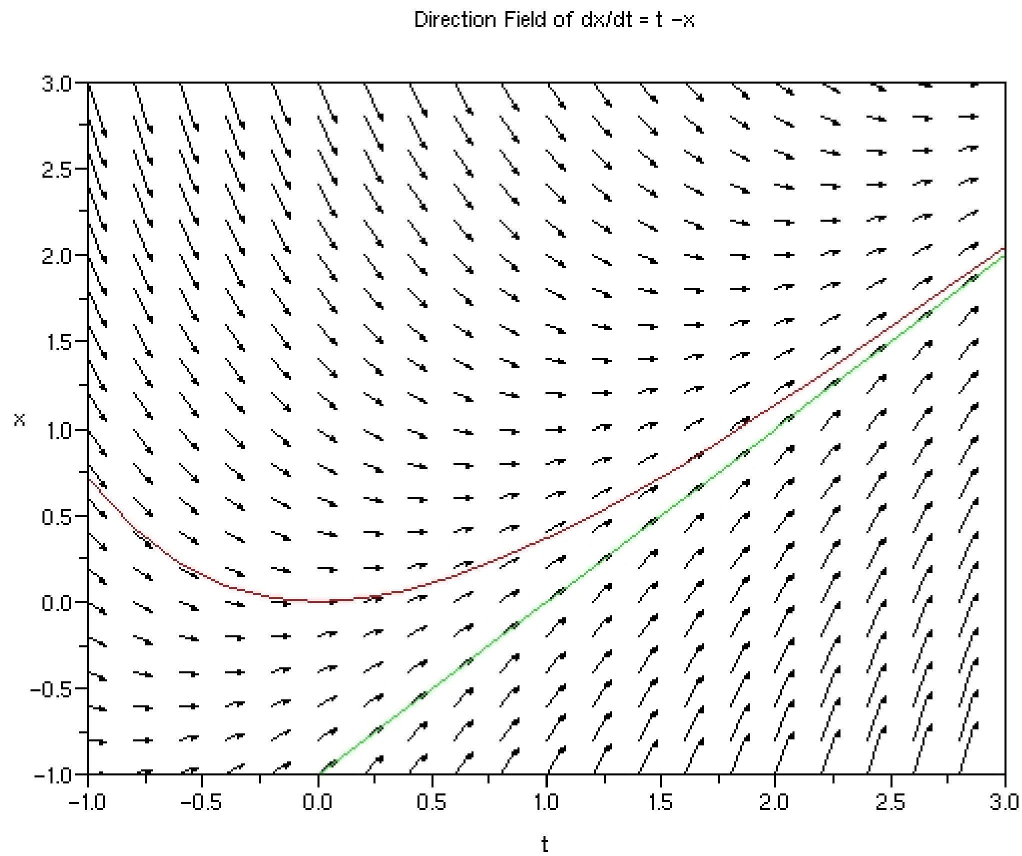


Let us now study the resulting graph. Note that $\frac{dx}{dt}$ seem to approach 1 as t gets large. This implies that solutions $x(t)$ seem to grow as t gets large. There does not appear to be any **equilibrium solutions**. Recall that equilibrium solutions $x(t)$ are those that do not change over time. Hence $x(t)$ should equal a constant. This implies $\frac{dx}{dt} = 0$. In a direction field, equilibrium solutions correspond to lines with slope 0, i.e. horizontal lines. As our differential equation is easy to solve, we know that the solutions are given by

$$x(t) = t - 1 + Ce^{-t}.$$

Let us plot two solutions corresponding to $C = 0$ and $C = 1$ with

```
--> t=-1:0.2:3;
--> x1=t-1;
--> x2=t-1 + exp(-t);
--> plot(t,x1,'g')
--> plot(t,x2,'r')
```



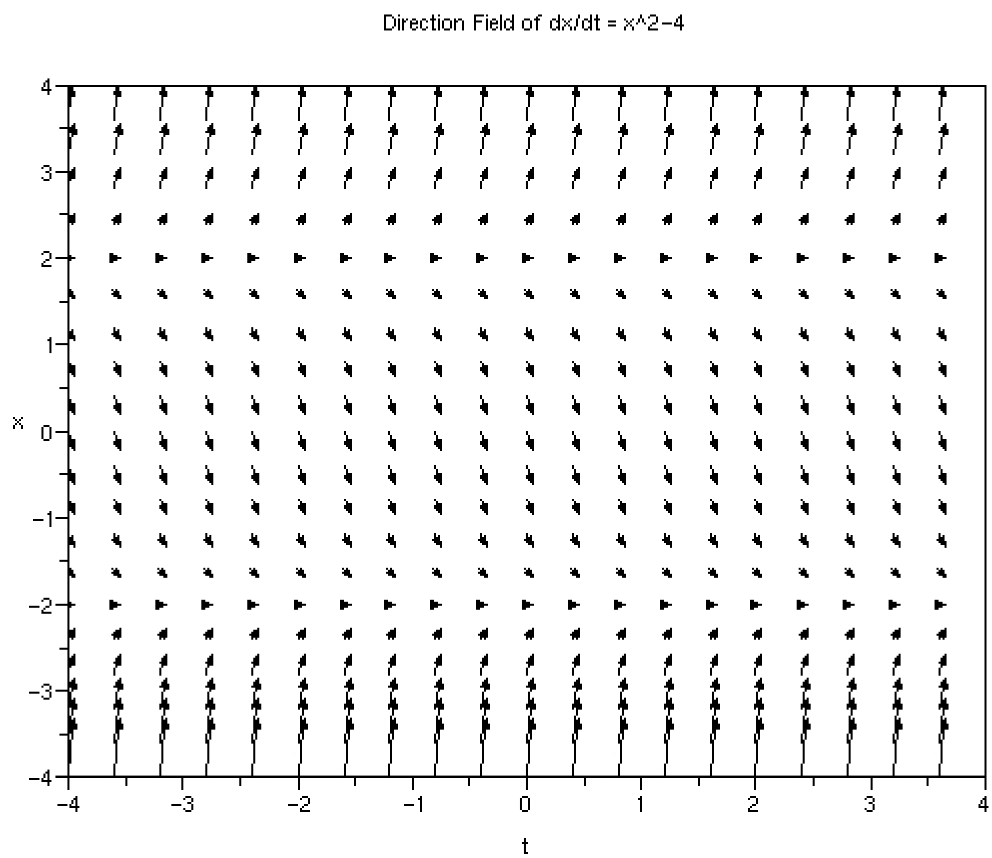
In conclusion, if we start with any initial point $x(t)$ on the graph and trace out the **trajectory** of the path, we will obtain the curve of a particular solution. Try this by starting with $x(0) = 1$ to compute the corresponding value of C, and plot the resulting particular solution into the direction field.

For our second example, let us consider the equation $\frac{dx}{dt} = x^2 - 4$. We graph the direction field with

```
--> deff('[xdot]=f(a,x)', 'xdot=[1 ; x(2).^2 -4]') // only the de changes
--> fchamp(f, 0, -4:0.4:4, -4:0.4:4,1.5)
--> mtlb_axis ( [-4 4 -4 4])
--> xlabel('t')
--> ylabel('x')
--> title( 'Direction Field of dx/dt = x^2-4')
```

Note that the only change to our function f is the part defining $x^2 - 4$. The // is the comment character. Scilab ignores everything after that. For the fchamp command, we added a fifth parameter which enlarges the size of the arrow head to make it easier to discern the equilibriums. If this parameter is not specified, the default size is 1.

Now, it is rather tedious to type the long string of commands every time we want to graph a direction field. We can improve the process by making use of a scilab script. Scripts are basically text files which end with a .sci extension, and contain a list of scilab commands. Click on → Editor to call out a new window. Type the list of commands above line by line (without the "−− >"). Now click → File → Save as, and name it "dfield.sci". Remember to save the file in an appropriate directory, either your h: drive or into your own thumbdrive. To run your script file simply click → File → Exec and locate your script. If you wish to modify your script, you can use → File → Open, to do so.



By maximizing the figure window if necessary, observe that there are two horizontal lines at $x = -2$ and $x = 2$. These are precisely the equilibrium solutions. Note that solutions near $x = 2$ tend to move away whereas solutions near $x = -2$ tend to converge. Hence $x = -2$ is a **stable** equilibrium whereas $x = 2$ is unstable. Besides these two, what do the graphs of other solutions look like?

Let us solve for $x(t)$, first assuming $x \neq 2$ or -2:

$$\frac{dx}{dt} = x^2 - 4 \quad \Longrightarrow \quad \int \frac{dx}{x^2 - 4} = \int 1dt$$
$$\Longrightarrow \quad \log|x-2| - \log|x+2| = 4t + C.$$

Recalling the *no-crossing principle*, i.e. the solution curves should not cross, we have three cases:

$$\frac{x-2}{x+2} = \begin{cases} Ae^{4t} & \text{if } x > 2, \\ -Ae^{4t} & \text{if } -2 < x < 2, \\ Ae^{4t} & \text{if } x < -2. \end{cases}$$

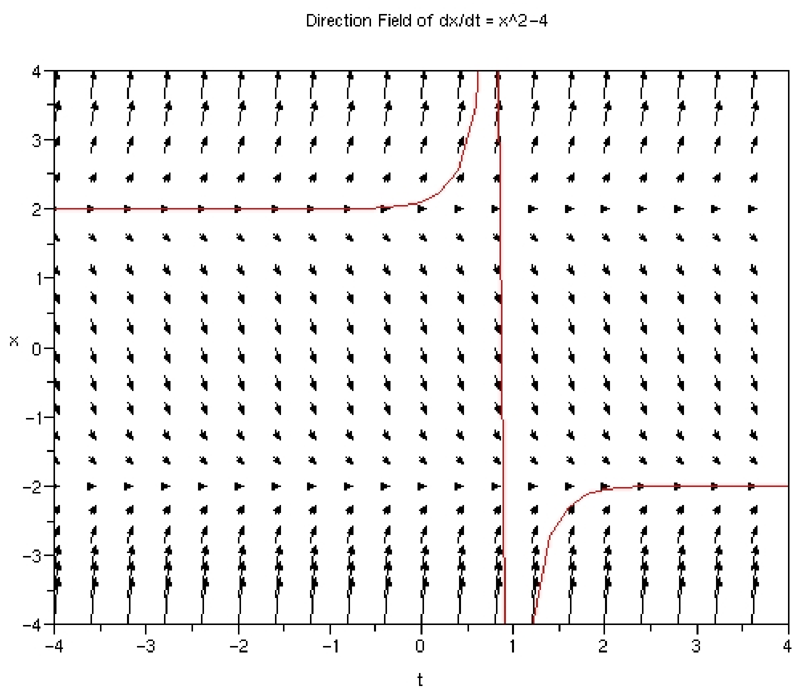
Here, A is a positive constant.
In the first case $x > 2$, some rearrangement gives us

$$x = \frac{2 + 2Ae^{4t}}{1 - Ae^{4t}}.$$

Beginning with the condition, $x(0) = 2.1$, we calculate $A = 1/41 \approx 0.0244$. Suppose we try to plot this graph with

```
--> t1=-4:0.2:4;    //we smaller intervals for a smoother curve
--> x1 = 2*(1+0.0244*exp(4*t1))./(1-0.0244*exp(4*t1));
--> plot(t1,x1,'r')
--> mtlb_axis([-4 4 -4 4]);
```

We will obtain:



You should be suspicious when you see that vertical red line and the portion of the curve that lies below $x = -2$.

Exercise 2A

1. This solution curve clearly violates the *no-crossing principle*. What went wrong?

2. Use appropriate initial values of $x(0)$ to plot at least 6 solution curves on the direction field. (Hint: For each solution curve, restrict your range of t so that only the correct portion of the curve appears.)

3. Plot the direction field for
$$\frac{dy}{dx} = 3\sin y + y - 2$$
on a suitably large rectangle. Discuss the behaviour as $x \to \infty$ and find the equilibriums.

Part B: Numerical Solvers

Consider the following example of a differential equation which cannot be easily solved by known techniques.
$$\frac{dx}{dt} = t + x^2.$$
While the direction field helps us visualize the family of solutions, we sometimes need numerical values. In such cases, it is often very useful to use numerical methods to approximate solutions. We shall learn how to do this using scilab's numerical solver called **ode** which solves initial value problems (IVP) of the following form:
$$\frac{dx}{dt} = f(t,x), \qquad a \le x \le b,\ x(a) = x_0.$$
Note that the initial condition should always be supplied because we are approximating a unique particular solution and not the whole family of solutions.
Let us start with the following simple IVP first.
$$\frac{dx}{dt} = \frac{t}{x}, \qquad 0 \le t \le 2,\ x(0) = 1.$$

```
-->function xdot = f(t,x)
-->xdot = t/x
-->endfunction
-->t=0:0.1:2;
-->x=ode(1, 0, t, f);
-->plot(t,x)
```

The first three lines defines a function $f(t, x)$ in terms of variables t and x. Note that there is no need to use "./" for dividing in functions. The next command creates an array of points, called t, where we will approximate solutions to the d.e. with the ode command. The first two parameters of the ode command are the initial values $x_0 = 1$ and $t_0 = 0$, i.e. $x(t_0) = x_0$. The last command then plots the points t against the approximate values of $x(t)$.
Since this d.e. is separable, we can solve the IVP to get $x(t) = \sqrt{t^2+1}$ and compare the two graphs.

```
--> X = sqrt( t.^2+1);
--> plot(t, X, 'r')
```

Note that you actually cannot see any difference graphically. To see how good are the approximations, we use the following commands and note how many decimal places the two sets of values agree to. (Use format(20) if necessary.)

```
-->[X ; x]
```

We can also use ode to approximate solutions to IVP involving second order differential equations.

$$x'' = f(t, x, x'), \qquad , a \leq x \leq b, x(a) = \alpha_0, x'(a) = \alpha_1.$$

Consider the following past year midterm question:

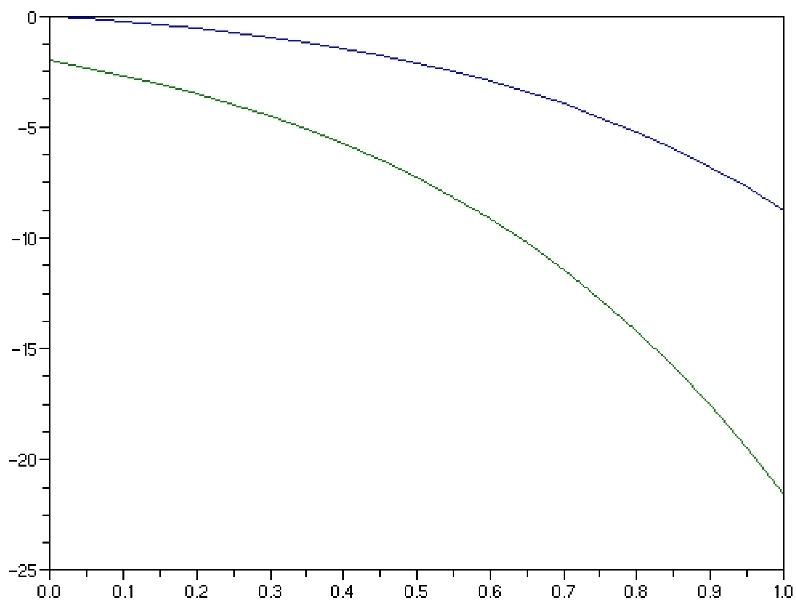
$$x'' - 3x' + 2x = te^t, \qquad x(0) = 0, x'(0) = -2.$$

Since the solution is $x = e^t - e^{2t} + (-\frac{1}{2}t^2 - t)e^t$, we have $x(1) = -e^2 - \frac{1}{2}e \approx -8.7482$.
To solve this numerically, we write our new function as called myfunction.

```
-->function xdot = myfunction(t,x)
-->xdot = [x(2) ; 3*x(2) - 2*x(1)+t*exp(t)] //only need to change the part after ;
-->endfunction
-->t=0:0.05:1;
-->x0 = [0 ; -2];
-->sol=ode(x0, 0, t, myfunction);
-->plot(t,sol)
-->sol(1,$)
```

Launch your browser variable window to check that you now have an array called **t** which contains increasing values from 0 to 1, and a double array **sol** which contains two rows of values. The last value in the first row of **sol** is exactly the solution that we want! We obtain its value with the command **sol(1,$)**. The first argument indicates the row and the second argument indicates the column. **$** meant we want the value in the last column.

Let us now examine the commands we used. We created a new function called "myfunction" with two variables **t** and **x**. The variable **x** actually contains two values, **x(1)** which represent x in our d.e. and **x(2)** which represents x'. As indicated by the comments, we are only interested in the portion after the semi-colon ;. If we write our second order d.e. as $x'' = f(t, x, x')$ where t, x, x' are replaced by **t, x(1), x(2)** respectivel, then we input precisely the expression for $f(t, x, x')$ after the semi-colon.
The ode command approximates the solution to myfunction on the interval $0 \leq t \leq 1$, with the two initial conditions given by x0. The numerical solutions are stored in the double array **sol** where the first row contains x and the second row contains x'. You should be able to tell which solution curve is x or x' by looking at the initial values at $t = 0$.

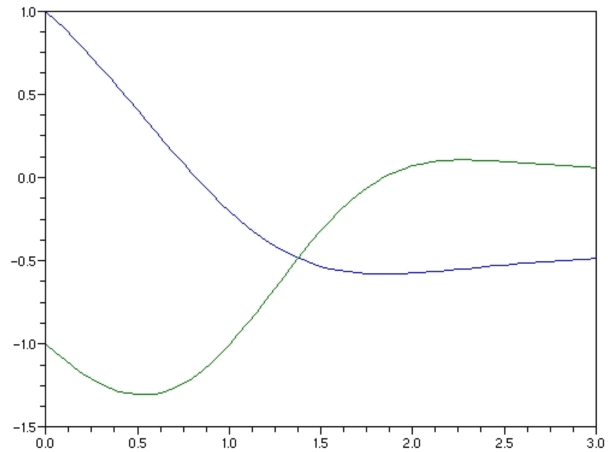


Let us solve another example, this time involving a differential equation with non constant coefficients.

$$x'' + t^2 x' + x = 0, \qquad x(0) = 1, x'(0) = -1,$$

for $0 \leq t \leq 3$. We write the set of commands as a script file.

```
function xdot = myfunction(t,x)
xdot = [x(2) ; -t^2*x(2) - x(1)]   //change the function
endfunction
t=0:0.05:3;    // change the range of t
x0 = [1; -1];  // change the initial value
sol=ode(x0, 0, t, myfunction);
plot(t,sol)
```



Use the initial values to decide which curve is $x(t)$ and which is $x'(t)$.

Exercise 2B

1. Let $f(x)$ be a solution of the differential equation $f'(x) = \frac{x+1}{\sqrt{x}}$, $x > 0$, such that $f(1) = 2$. Find $f(4)$. (This was a past year midterm question. Use ode to get an approximate answer.)

2. Use ode to find solutions to

$$\frac{dx}{dt} = t + x^2, \qquad x(0) = 1.$$

   (Hint: You might get an error. Try plotting your graph on the rectangle, $0 \leq t \leq 1, 0 \leq x \leq 100$.)

3.

$$x'' + 2x' + x = 2 + e^{2t}, \qquad x(0) = 37/9, x'(0) = -7/9.$$

   Find $x(1)$.

—The End—