Lennart Edsberg

Nada, KTH

Applied Numerical Methods, part 1, DN2221

Computer Lab 1
ODE-systems of LCC-type and stability

In this exercise the following problems are treated:

- linear ODE-systems with constant coefficients (LCC-systems)
- stability of trajectories and critical points

LCC-systems can be solved analytically (see course material) and need therefore no numerical method. LCC-systems are treated in mathematical courses, but are still presented in this course for reviewing purposes since it is easy to obtain their solution with the `expm`-function in Matlab and since it also gives exercise in displaying the solution curves graphically. There are also a few exercises on the important concept of stability of ODE-systems, both the stability of solution curves and equilibrium points.

We start by illustrating how a Matlab program for solution of an LCC-problem can look like:

Introductory example

Given the following second order differential equation:

$$y'' + 2y' + 5y = 0$$

with the initial values $y(0) = 0$ and $y'(0) = 1$. Compute the solution of this ODE-problem on a suitable t-interval.

Rewrite this scalar problem to the standard form, i.e. as a system of first order ODEs $\mathbf{u}' = A\mathbf{u}$, which is solved analytically with the `expm`-function. Input data to the Matlab program solving this problem consists of the matrix A, the initial vector $\mathbf{u}(0)$ and gridpoint data h and N for the points $t_i = ih, i = 0, 1, 2, \ldots, N$, where the solution shall be computed and plotted. The output from the program is the graph of the solution at these gridpoints. In the Matlab program below the solution $\mathbf{u}(t)$ is stored in a resultmatrix `result`. Each column in `result` corresponds to the solution vector at a gridpoint t_i. The first column is the initial vector $\mathbf{u}(0)$.

The plotted result is shown in a graph.

```
A=[0 1;-5 -2];
u0=[0 1]';t0=0;
h=0.1;N=60;
result=[u0];time=[t0];
for k=1:N
  t=k*h;
  u=expm(A*t)*u0;
  result=[result u];
  time=[time t];
end
plot(time,result)
```

This Matlab program plots two curves in the graph. The first row in the `result`-matrix corresponds to the solution $y(t)$, while the second row corresponds to the derivative of the solution $z(t) = y'(t)$, since $\mathbf{u}(t)$ is defined as $(y(t),\, z(t))^T$. The initial values of y och z are different in this case so it is easy to see which curve corresponds to which variable. Otherwise you can use different plot-symbols for different curves.

Observe that the time step h must be chosen with some care if the plotted curve is to be smooth. In the left graph $h = 0.1$ is used as stepsize while the right graph is plotted with $h = 1$. With so few points, however, the line segments building up the curves are clearly seen and the curves gets rough.

With too small time steps, on the other hand say $h = 0.001$, the computation of the `result`-matrix will take unreasonably long time. Hence it is advisable to work out the gridpoint spacing interactively. For other problems it may also be better to use `loglog`, `semilogx` or `semilogy` instead of `plot` to display the important qualitative features of solution curves.

——————— *** ———————

Part 1. Solution of ODE-systems with constant coefficients

Electric circuit

Given the following simple electric circuit with a voltage source of size E, a resistance R, an inductance L and a capacitance C. The components are coupled in series.
Assume that the circuit is first at rest. The switch is pressed at $t = 0$ and a current starts to go through the circuit. Introduce the variable q defined by

the relation $\dot{q} = i$, and the following differential equation can be set up:

$$L\ddot{q} + R\dot{q} + \frac{1}{C}q = E, \quad q(0) = 0, \quad \dot{q}(0) = 0$$

Rewrite this scalar ODE as a system of first order ODEs then write a Matlab program to solve the problem for the following parameter values of E, L, C och R: $E = 10$, $L = C = 0.1$, $R = 1, 10, 100, 1000, 10000$. Observe that this ODE-system is *inhomogenous*.

Plot the solutions $y(t)$ on suitable time intervals and with suitable time steps. Use the `subplot`-command to obtain the three graphs in the same figure.

Part 2. Stability of ODE-systems

a. Stability of the solutions of an ODE-system of LCC-type

In many technical applications where a differential equation is used to model a dynamic process, it is important to see how the solution curves change as a parameter in the model is changed. For a control system it is of great interest to investigate how the the stability properties change as a parameter is changed continuously.

Given the following third order differential equation

$$y''' + 3y'' + 2y' + Ky = 0, \quad y(0) = 1,\ y'(0) = 1,\ y''(0) = 1$$

Investigate how the solution $y(t)$ is changed as K varies in the interval $K \geq 0$.

Rewrite this third order ODE as a system of first order ODEs. We then obtain a system

$$\mathbf{u}' = A(K)\mathbf{u}, \qquad \mathbf{u}(0) = (1, 1, 1)^T$$

Write a Matlab-program showing the solutions $y(t)$ for $K = 0, 1, 4, 8$. Use `subplot` to plot the four graphs in the same figure. Estimate from these plots an approximate value of K for which the solution becomes unstable.

The stability properties can also be shown in a so called *root locus*, often used in control theory to visualize the stability properties for a regulator modeled by an ODE of LCC-type. A root locus is a graph showing the paths of the eigenvalues in the complex plane as a parameter varies. Assume that all eigenvalues start in the left half plane for say $K = 0$, where we have a stable system. When K then increases the eigenvalues will follow continuous paths and for a certain value of K a few eigenvalues eventually enter the right half plane and the system becomes unstable.

Write a Matlab program that plots the root locus when $0 \leq K \leq 10$. Compute accurately the smallest value of K that gives an unstable system. To present the results take a paper copy of your program and the plot of the root locus. Hint: in this part of the lab it is easiest to plot complex numbers in the complex plane. If the number is real, make it complex by adding a small imaginary number, say i*1e-8.

b. Stability of critical points of a nonlinear ODE-system

Given the following system of nonlinear ordinary differential equations

$$\begin{aligned} u_1' &= 5u_1 + 4u_2 - u_1 u_3 \\ u_2' &= u_1 + 4u_2 - u_2 u_3 \\ u_3' &= u_1^2 + u_2^2 - 89 \end{aligned}$$

Write a Matlab-program with the help of which you compute all critical points of the ODE-system and also compute which of these critical points are stable. Hint: use Newton's method to compute the critical points with at least 5 significant digits. There are four critical points and they are situated in the neighbourhood of the following points in R^3: $(8, -5, 2), (-8, 5, 2), (9, 3, 7)$ and $(-9, -3, 7)$.