BENG 221 Mathematical Methods in Bioengineering

Lecture 4

Tutorial

Analytic and Numerical Methods in ODEs

Class notes
Gert Cauwenberghs, Dept. of Bioengineering, UC San Diego

4.1

Overview

Summary

We will review *analytic and numerical techniques* for solving ODEs, using pencil and paper, and implemented in Matlab. These simple techniques lay the foundations for solving more complex systems of PDEs in the coming weeks.

By way of example we study the dynamics of a *ring oscillator*, a circular chain of three inverters with identical capacitive loading. 4.2

Contents

1 Example: Ring Oscillator Dynamics

Some circuit elements

$$i_{out} = g(v_{in}) \approx -Gv_{in} \tag{1}$$

$$v = \frac{Q}{C} = \frac{1}{C}\int_{-\infty}^{t} i \, dt \tag{2}$$

4.4

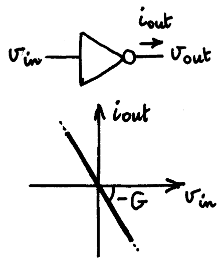


Figure 1: An inverting transconductor (inverter) converts and input voltage to an output current, with gain $-G$.

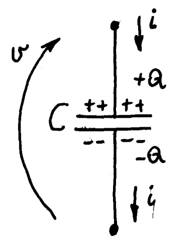


Figure 2: A capacitor converts charge, or integrated current, to voltage with gain $1/C$.

Ring Oscillator

$$\begin{aligned} C\,\frac{dv_1}{dt} &= g(v_3) \approx -G\,v_3 \\ C\,\frac{dv_2}{dt} &= g(v_1) \approx -G\,v_1 \\ C\,\frac{dv_3}{dt} &= g(v_2) \approx -G\,v_2 \end{aligned} \tag{3}$$

$$\begin{aligned} v_1(0) &= v_{10} \\ v_2(0) &= v_{20} \\ v_3(0) &= v_{30} \end{aligned} \tag{4}$$

4.5

2 Analytic ODE Solution

Eigenvalues

Ring oscillator ODE dynamics in matrix notation:

$$\frac{d\mathbf{v}}{dt} = \mathbf{A}\mathbf{v} \qquad \mathbf{v}(0) = \mathbf{v}_0 \tag{5}$$

with

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \quad \mathbf{v}(0) = \begin{pmatrix} v_{10} \\ v_{20} \\ v_{30} \end{pmatrix} \tag{6}$$

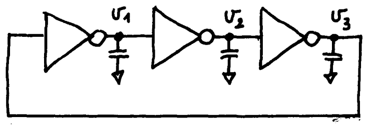
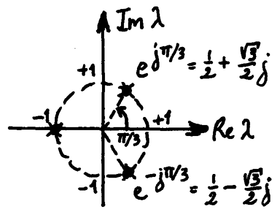


Figure 3: A 3-inverter ring oscillator with capacitive loading.



where $G/C \equiv 1$ with no loss of generality.

Eigenvectors $\mathbf{x}_i$ and corresponding eigenvalues λ_i of $\mathbf{A}$ satisfy $\mathbf{A}\ \mathbf{x}_i = \lambda_i\ \mathbf{x}_i$, or $\det(\mathbf{A} - \lambda_i\ \mathbf{I}) = 0$, which reduces to $\lambda_i^{\ 3} + 1 = 0$, with three solutions:

$$\lambda_i = (-1)^{\frac{1}{3}} = \begin{cases} -1 \\ e^{+j\pi/3} = \frac{1}{2} + j\frac{\sqrt{3}}{2} \\ e^{-j\pi/3} = \frac{1}{2} - j\frac{\sqrt{3}}{2} \end{cases} \tag{7}$$

4.6

Eigenvectors

The corresponding eigenvectors are:

$$\begin{aligned} \lambda_1 &= -1: & \mathbf{x}_1 &= \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ \lambda_2 &= e^{+j\pi/3} = \frac{1}{2} + j\frac{\sqrt{3}}{2}: & \mathbf{x}_2 &= \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ e^{+j2\pi/3} \\ e^{-j2\pi/3} \end{pmatrix} \\ \lambda_3 &= e^{-j\pi/3} = \frac{1}{2} - j\frac{\sqrt{3}}{2}: & \mathbf{x}_3 &= \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ e^{-j2\pi/3} \\ e^{+j2\pi/3} \end{pmatrix} \end{aligned} \tag{8}$$

The eigenvectors form a complex orthonormal basis:

$$\mathbf{x}_i^{\ *}\ \mathbf{x}_j = \delta_{ij}, \qquad i,\ j = 1, \ldots 3 \tag{9}$$

where $\mathbf{x}_i^{\ *}$ is the complex conjugate transpose of $\mathbf{x}_i$. 4.7

Eigenmodes

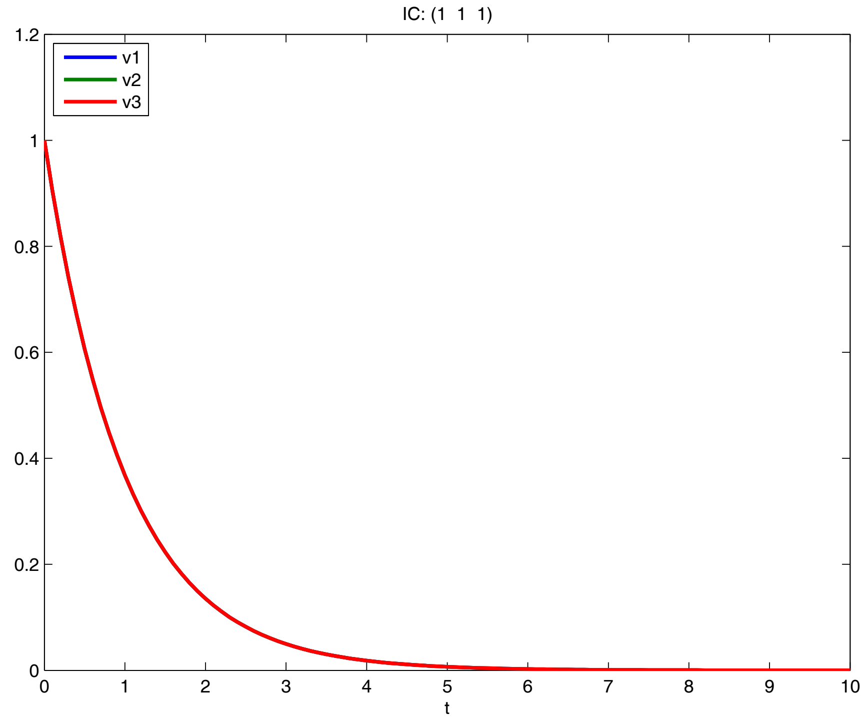


Figure 4: Ring oscillator ODE solution for $\mathbf{v}(0) = (1,1,1)^T$.

The general solution is the superposition of eigenmodes (see Lecture 1):

$$
\begin{aligned}
\mathbf{v} &= \sum_{i=1}^{3} c_i \, \mathbf{x}_i \, e^{\lambda_i t} \\
&= c_1 \, e^{-t} \, \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \\
&\quad\; c_2 \, e^{\frac{1}{2}t} \, e^{j\frac{\sqrt{3}}{2}t} \, \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ e^{+j2\pi/3} \\ e^{-j2\pi/3} \end{pmatrix} + \\
&\quad\; c_3 \, e^{\frac{1}{2}t} \, e^{-j\frac{\sqrt{3}}{2}t} \, \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ e^{-j2\pi/3} \\ e^{+j2\pi/3} \end{pmatrix}
\end{aligned} \tag{10}
$$

$\mathbf{v}(t)$ is real, and so c_2 and c_3 must be complex conjugate. Therefore, the second and third eigenmodes are oscillatory with an exponentially rising carrier. The first eigenmode is a decaying exponential common-mode transient. 4.8

First Eigenmode– Common-mode Decaying Exponential 4.9

Second/third Eigenmode– Exponentially Rising Three-phase Oscillations 4.10

Initial Conditions

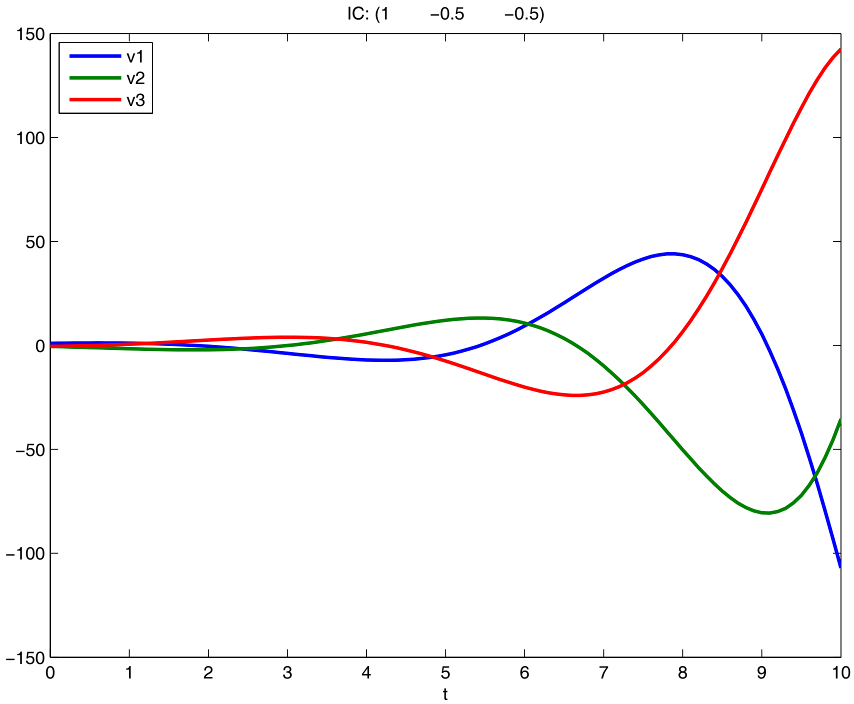


Figure 5: Ring oscillator ODE solution for $\mathbf{v}(0) = (1, -\frac{1}{2}, -\frac{1}{2})^T$.

The IC constrained solution is obtained by virtue of the orthonormality of the eigenvectors (see also Lecture 1):

$$\mathbf{v} = \sum_{i=1}^{n} \mathbf{x}_i^* \ \mathbf{v}(0) \ \mathbf{x}_i \ e^{\lambda_i t} \tag{11}$$

which, using the identity $e^{+j\alpha} + e^{-j\alpha} = 2\ cos(\alpha)$, leads to:

$$\begin{aligned} v_1 \quad = \quad & \frac{e^{-t}}{3} \left(v_{10} + v_{20} + v_{30}\right) + \\ & \frac{2\ e^{t/2}}{3} \ (v_{10} \cos(\frac{\sqrt{3}}{2} t) + \\ & \qquad v_{20} \cos(\frac{\sqrt{3}}{2} t + \frac{2\pi}{3}) + v_{30} \cos(\frac{\sqrt{3}}{2} t - \frac{2\pi}{3})) \end{aligned} \tag{12}$$

and identical expressions for v_2 and v_3 under ordered permutation of the indices (consistent with the ring symmetry). 4.11

3 Numerical Verification

Matlab Implementation

Using the eigenvector-eigenvalue decomposition of $\mathbf{A}$ in matrix form:

$$\mathbf{A}\ \mathbf{X} = \mathbf{X}\ \mathbf{s} \tag{13}$$

where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ and $\mathbf{s} = diag(\lambda_1, \lambda_2, \lambda_3)$, the solution (11) can be expressed in matrix form:

$$\mathbf{v} = \mathbf{X}\ diag(\mathbf{X}^*\ \mathbf{v}(0)) e^{diag(\mathbf{s})t} \tag{14}$$

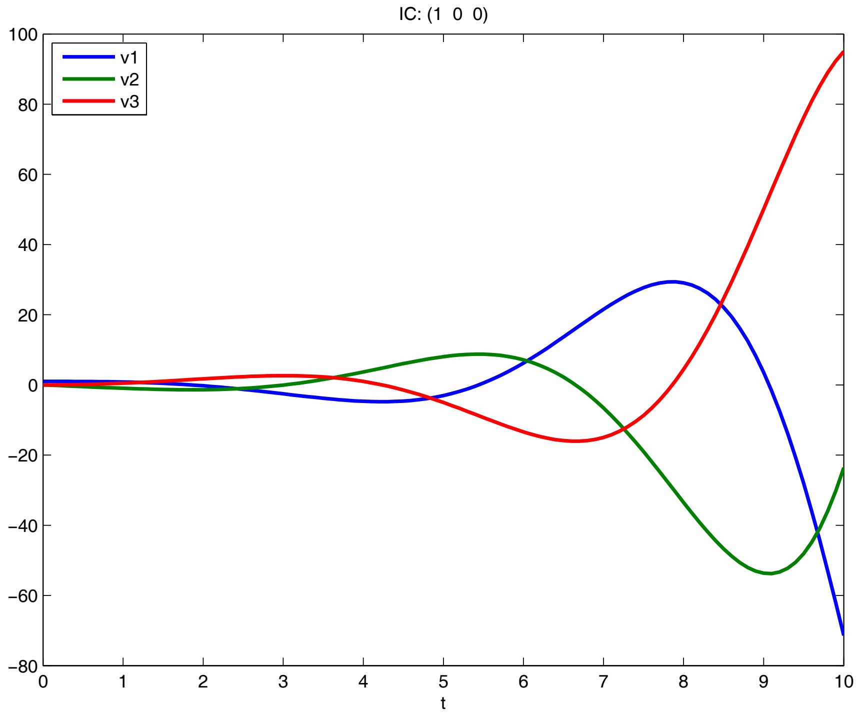


Figure 6: Ring oscillator ODE solution for $\mathbf{v}(0) = (1,0,0)^T$.

for efficient matlab implementation:

```
[X, s] = eig(A);
V = X * diag(X' * V0) * exp(diag(s) * t);
```
4.12

Initial Conditions 4.13

4 Numerical Simulation

Euler Integration

Euler numerical integration produces approximate solutions to:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t),t) \tag{15}$$

at discrete time intervals $t = n\Delta t$, by finite difference approximation of the derivative:

$$\frac{d\mathbf{x}}{dt}(t) = \frac{1}{\Delta t}(\mathbf{x}(t+\Delta t) - \mathbf{x}(t)) + \mathcal{O}(\Delta t^2) \tag{16}$$

leading to the recursion:

$$\mathbf{x}(t+\Delta t) \approx \mathbf{x}(t) + \Delta t\ \mathbf{f}(\mathbf{x}(t),t). \tag{17}$$

Matlab *Euler* example (ring oscillator):

```
Ve = V0; % Euler approximation, initialize to IC
Vs = V0; % Euler state variable, initialize
for te = tstep:tstep:trange
   Vs = Vs + A * Vs * tstep;
   Ve = [Ve, Vs];
end
```
4.14

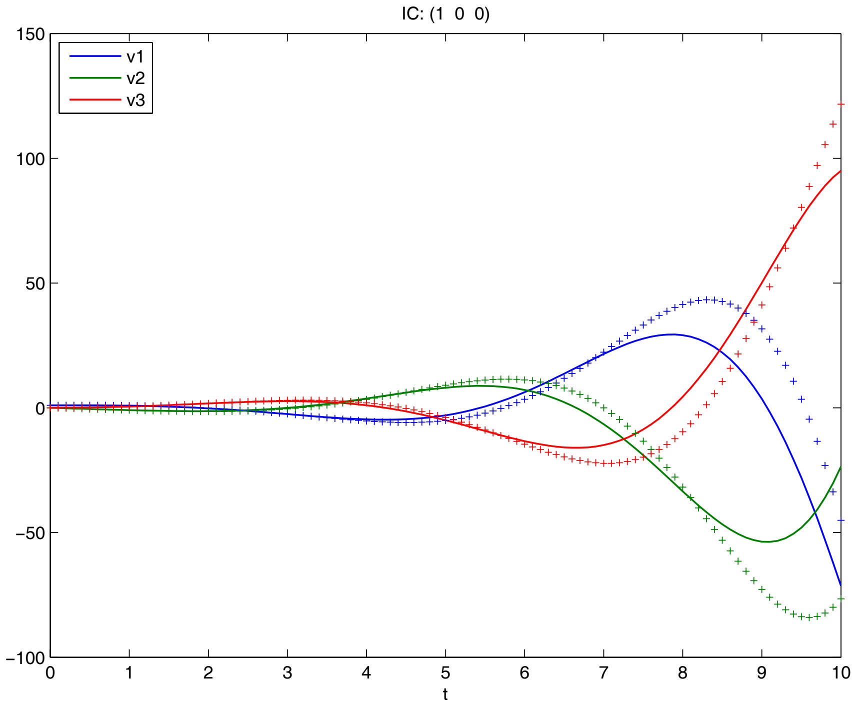


Figure 7: Ring oscillator ODE simulation using Euler integration.

Euler Integration

4.15

Crank-Nicolson Integration

Better numerical ODE methods exist that use higher-order finite difference approximations of the derivative. *Crank-Nicolson* is a second order method that approximates (15) more accurately using a centered version of the finite difference approximation of the derivative:

$$\frac{d\mathbf{x}}{dt}(t+\frac{\Delta t}{2}) = \frac{1}{\Delta t}(\mathbf{x}(t+\Delta t)-\mathbf{x}(t)) + \mathscr{O}(\Delta t^3) \tag{18}$$

leading to a recursion:

$$\begin{aligned} \mathbf{x}(t+\Delta t) &\approx \mathbf{x}(t) + \Delta t\ \mathbf{f}(\mathbf{x}(t+\frac{\Delta t}{2}), t+\frac{\Delta t}{2}) \\ &\approx \mathbf{x}(t) + \frac{\Delta t}{2}(\mathbf{f}(\mathbf{x}(t),t) + \mathbf{f}(\mathbf{x}(t+\Delta t), t+\Delta t)). \end{aligned} \tag{19}$$

Matlab *CN* example (ring oscillator):

```
G = (eye(3) - A * tstep / 2) \ (eye(3) + A * tstep / 2);
Vc = V0; % CN approximation, initialize to IC
Vs = V0; % CN state variable, initialize
for te = tstep:tstep:trange
   Vs = G * Vs;
   Vc = [Vc, Vs];
end
```

4.16

Crank-Nicolson Integration

4.17

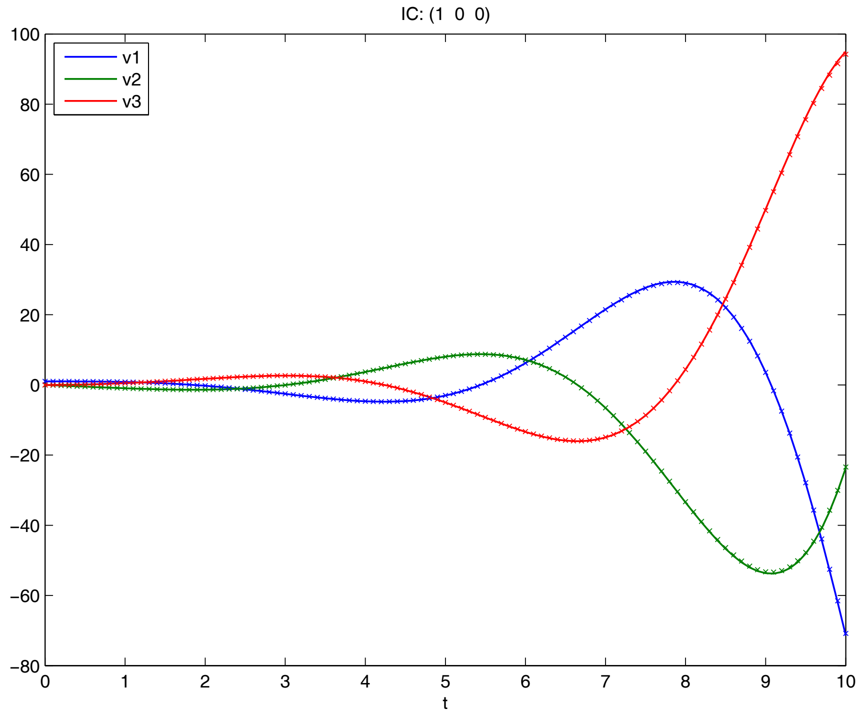


Figure 8: Ring oscillator ODE simulation using Crank-Nicolson integration.

Higher-Order Integration

Crank-Nicolson (19) is *implicit* in that the solution at time step $t+\Delta t$ is recursive in the state variable $\mathbf{x}(t+\Delta t)$, rather than forward *e.g.*, given in previous values of the state variable $\mathbf{x}(t)$. More advanced methods, such as *Runge-Kutta* and several of Matlab's built-in ODE solvers, are explicit and/or higher order, allowing faster integration although possibly at the expense of numerical stability.

Matlab *ode23* example (ring oscillator):

```
OdeOptions = odeset('RelTol', 1e-9); % accuracy, please
dVdt = @(t,V) A * V;
[tm, Vm] = ode23(dVdt, [0 trange], V0);
```
4.18

Explicit Forward Second-Order Integration 4.19

Matlab Built-in *ode23* Solver 4.20

Further Reading

Bibliography

References

[1] R. Haberman, *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems,* 4th Ed., 2004, Ch. 6.

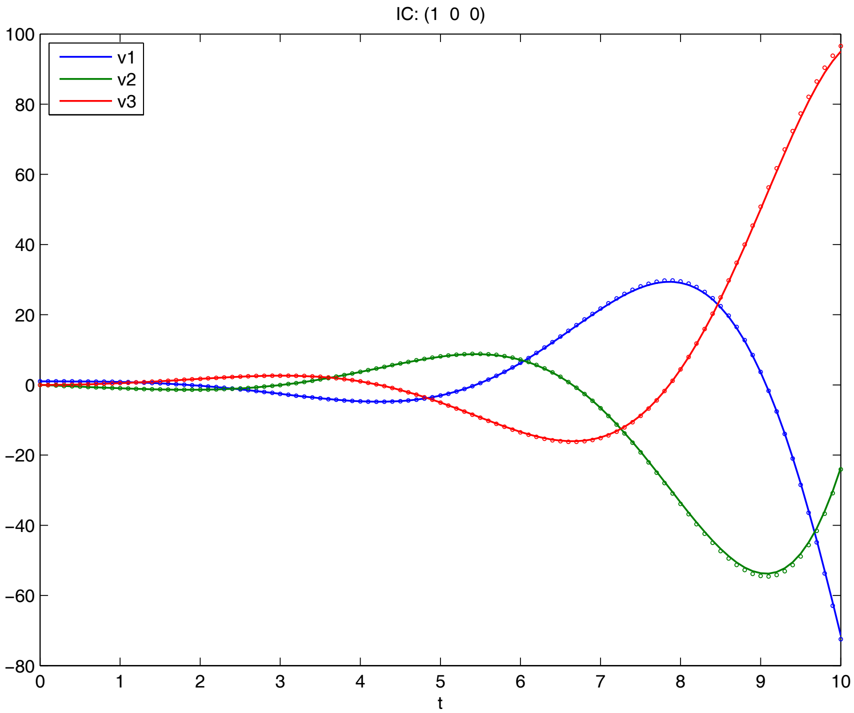


Figure 9: Ring oscillator ODE simulation using an explicit version of Crank-Nicolson integration using a forward series expansion up to second order.

[2] Wikipedia, *Conjugate Transpose,* `http://en.wikipedia.org/wiki/Conjugate_transpose`.

[3] Wikipedia, *Numerical Ordinary Differential Equations,* `http://en.wikipedia.org/wiki/Numerical_ordinary_differential_equations`.

[4] Wikipedia, *Runge Kutta Methods,* `http://en.wikipedia.org/wiki/Runge-Kutta_methods`.

4.21

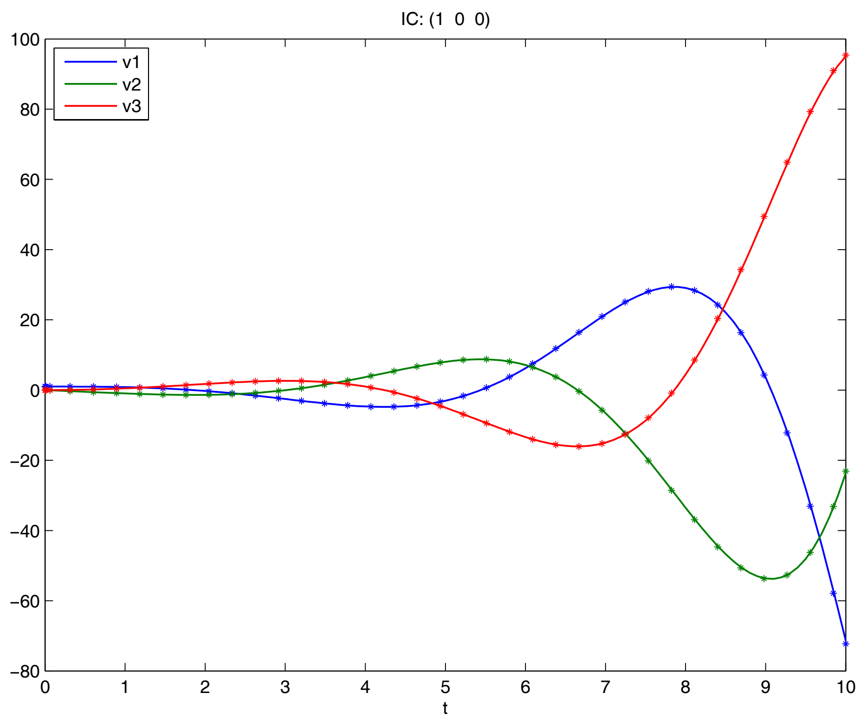


Figure 10: Ring oscillator ODE simulation using the Matlab built-in ode23 numeric ODE solver.