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.

Contents

1 Example: Ring Oscillator Dynamics 1

2 Analytic ODE Solution 2

3 Numerical Verification 5

4 Numerical Simulation 6

1 Example: Ring Oscillator Dynamics

Some circuit elements

\[ i_{out} = g(v_{in}) \approx -Gv_{in} \]  (1)

\[ v = \frac{Q}{C} = \frac{1}{C} \int_{-\infty}^{t} i \, dt \]  (2)
Figure 1: An inverting transconductor (inverter) converts 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

$$C \frac{dv_1}{dt} = g(v_3) \approx -G v_3$$

$$C \frac{dv_2}{dt} = g(v_1) \approx -G v_1$$

$$C \frac{dv_3}{dt} = g(v_2) \approx -G v_2$$

$$v_1(0) = v_{10}$$
$$v_2(0) = v_{20}$$
$$v_3(0) = v_{30}$$

2 Analytic ODE Solution

Eigenvalues

Ring oscillator ODE dynamics in matrix notation:

$$\frac{dv}{dt} = Av \quad v(0) = v_0$$

with

$$v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad A = \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \quad v(0) = \begin{pmatrix} v_{10} \\ v_{20} \\ v_{30} \end{pmatrix}$$
where $G/C \equiv 1$ with no loss of generality.

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

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

### Eigenvectors

The corresponding eigenvectors are:

$$\begin{align*}
\lambda_1 &= -1 : \quad \mathbf{x}_1 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\
\lambda_2 &= e^{+j\pi/3} = \frac{1}{2} + j\frac{\sqrt{3}}{2} : \quad \mathbf{x}_2 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ e^{j\pi/3} \\ e^{-j\pi/3} \end{pmatrix} \\
\lambda_3 &= e^{-j\pi/3} = \frac{1}{2} - j\frac{\sqrt{3}}{2} : \quad \mathbf{x}_3 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ e^{-j\pi/3} \\ e^{j\pi/3} \end{pmatrix}
\end{align*} \quad (8)$$

The eigenvectors form a complex orthonormal basis:

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

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

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

\[ v(t) = \sum_{i=1}^{3} c_i x_i e^{\lambda_i t} \]

\[ = c_1 e^{-t} \left( \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right) + \\
\]

\[ c_2 e^{\frac{2 \pi i}{3} t} e^{\frac{2 \pi i}{3}} \left( \begin{array}{c} 1 \\ e^{+2\pi i/3} \\ e^{-2\pi i/3} \end{array} \right) + \\
\]

\[ c_3 e^{\frac{2 \pi i}{3} t} e^{-\frac{2 \pi i}{3}} \left( \begin{array}{c} 1 \\ e^{-2\pi i/3} \\ e^{2\pi i/3} \end{array} \right) \]

(10)

\( v(t) \) 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.

First Eigenmode– Common-mode Decaying Exponential

Second/third Eigenmode– Exponentially Rising Three-phase Oscillations

Initial Conditions

Figure 4: Ring oscillator ODE solution for \( v(0) = (1,1,1)^T \).
The IC constrained solution is obtained by virtue of the orthonormality of the eigenvectors (see also Lecture 1):

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

(11)

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

$$v_1 = \frac{e^{-t}}{3} (v_{10} + v_{20} + v_{30}) + \frac{2 e^{t/2}}{3} (v_{10} \cos(\sqrt{3}/2 t) + v_{20} \cos(\sqrt{3}/2 t + \frac{2\pi}{3}) + v_{30} \cos(\sqrt{3}/2 t - \frac{2\pi}{3}))$$

(12)

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

3 Numerical Verification

Matlab Implementation

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

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

(13)

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

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

(14)
Figure 6: Ring oscillator ODE solution for $v(0) = (1, 0, 0)^T$.

for efficient matlab implementation:

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

Initial Conditions

4 Numerical Simulation

Euler Integration

Euler numerical integration produces approximate solutions to:

$$\frac{dx}{dt} = f(x(t), t)$$ (15)

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

$$\frac{dx}{dt}(t) = \frac{1}{\Delta t} (x(t+\Delta t) - x(t)) + O(\Delta t^2)$$ (16)

leading to the recursion:

$$x(t+\Delta t) \approx x(t) + \Delta t \cdot f(x(t), t).$$ (17)

Matlab Euler example (ring oscillator):

```matlab
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
```
Euler Integration

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:

\[
\frac{dx}{dt}(t + \frac{\Delta t}{2}) = \frac{1}{\Delta t} (x(t + \Delta t) - x(t)) + O(\Delta t^3)
\]  

leading to a recursion:

\[
x(t + \Delta t) \approx x(t) + \Delta t f(x(t + \Delta t/2), t + \Delta t/2)
\]

\[
\approx x(t) + \frac{\Delta t}{2} (f(x(t), t) + f(x(t + \Delta t), t + \Delta t)).
\]  

Matlab CN example (ring oscillator):

```matlab
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
```

Crank-Nicolson Integration
Higher-Order Integration

Crank-Nicolson is implicit in that the solution at time step $t + \Delta t$ is recursive in the state variable $x(t + \Delta t)$, rather than forward e.g., given in previous values of the state variable $x(t)$. 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):

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

Explicit Forward Second-Order Integration

Matlab Built-in ode23 Solver

Further Reading

Bibliography

References

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


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