Differential Equations

Equations that combine variables and functions of variables with derivatives of the variables as well.

Classical Methods:

Catalog of solution methods for various specific types of differential equations.

State Variable Techniques:

Linear, constant coefficient differential equations.

Systems of such equations.

Linear Approximations to non-linear systems.

First-order Linear, Homogeneous differential equation with constant coefficient

Form : $\dot{y} = \lambda y$, or: $\dot{y} - \lambda y = 0$: " • " indicates time derivative.

'Guess' at solution : $y(t) = ce^{\lambda t}$

$$\text{Try out guess : } \begin{cases} \dot{y}(t) = \lambda ce^{\lambda t} \\ \dot{y} - \lambda y = \lambda ce^{\lambda t} - \lambda y \\ = \lambda ce^{\lambda t} - \lambda ce^{\lambda t} = 0 \end{cases}$$

Constant is determined by initial condition: $y = y(0)$ at $t = 0$.

$$y(0) = ce^{\lambda \cdot 0} = c = y(0)$$

$$y(t) = y(0)e^{\lambda t}$$

First-order Linear Homogeneous Time Varying Coefficient

$$\dot{y} + a(t)y = 0$$

Solution:

$$\frac{dt}{y} = -a(t)dt \Rightarrow \ln y = -\int_{t_0}^{t} a(s)ds + c$$

$$y(t) = c\exp\left\{-\int_{t_0}^{t} a(s)ds\right\}$$

Example:

Compute future amount of principle plus continuously compounded timevarying interest at rate $r(t)$ on an initial amount $A(t_0)$.

$$\dot{A}(t) = r(t)A(t)$$

$$A(t) = A(t_0)\exp\left\{\int_{t_0}^{t} r(s)ds\right\}$$

First-order Linear Time-varying Nonhomogeneous with

Time-varying Coefficient

$$\dot{y} + a(t)y + b(t) = 0$$

Integrating Factor

Define:

$$\mu(t) = \exp\left\{\int^t a(s)ds\right\}$$

Therefore:

$$\frac{d\mu}{dt} = a(t)\exp\left\{\int^t a(s)ds\right\} = a(t)\mu(t) = \dot{\mu}$$

Now note that:

$$\frac{d}{dt}(\mu y) = \dot{\mu}y + \mu\dot{y} = \mu a y + \mu\dot{y}$$

However:

$$\mu(\dot{y} + a(t)y + b(t)) = 0$$
$$\mu\dot{y} + \mu a y = -\mu b(t)$$
$$\frac{d}{dt}(\mu y) = -\mu(t)b(t)$$

Integrate:

$$\mu y = \int^t \frac{d}{dt}(\mu y) = -\int^t \mu(s)b(s)ds + c$$
$$y(t) = \frac{-\int^t \mu(s)b(s)ds + c}{\mu(t)}$$

where:

$$\mu(t) = \exp\left\{\int^t a(s)ds\right\}$$

Application

Compute future amount of principle plus continuously compounded time varying interest at rate $r(t)$ on an initial amount $A(t_0)$ plus a deposit flow $s(t), t \geq t_0$.

$$\dot{A}(t) = r(t)A(t) + s(t)$$
$$A(t) = \frac{A(t_0) + \int_{t_0}^t \exp\{-R(\tau)\}s(\tau)d\tau}{\exp\{-R(t)\}}$$
$$R(t) \equiv \int_{t_0}^t r(\tau)d\tau$$

Second-order, Linear, Homogeneous, with Constant Coefficients

Form : $\ddot{y} + a_1\dot{y} + a_0 y = 0$: " • •" indicates 2^{nd} time derivative.

Characteristic Equation : $\lambda^2 + a_1\lambda + a_0 = 0 \Rightarrow roots : \lambda_1, \lambda_2$

"Guessed Solution" : $y(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$

Check :

$$\begin{aligned}
y(t) &= c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} \\
\dot{y}(t) &= c_1\lambda_1 e^{\lambda_1 t} + c_2\lambda_2 e^{\lambda_2 t} \\
\ddot{y}(t) &= c_1\lambda_1^2 e^{\lambda_1 t} + c_2\lambda_2^2 e^{\lambda_2 t} \\
\ddot{y} + a_1\dot{y} + a_0 y &= (\lambda_1^2 + a_1\lambda_1 + a_0)c_1 e^{\lambda_1 t} + (\lambda_2^2 + a_1\lambda_2 + a_0)c_2 e^{\lambda_2 t} = 0
\end{aligned}$$

Two initial (or one initial, one final) value conditions needed for c_1, c_2

State Variables Approach

Consider the vector - matrix differential equation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$$

$\mathbf{x}$: $n \times 1$ vector of state variables

$\mathbf{A}$: $n \times n$ matrix of constants.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Eigenvalues and Eigenvectors

Consider a square matrix, **A**

Eigenvalues, λ_i are solutions to:

$$|\mathbf{A} - \lambda\mathbf{I}| = \left| \begin{bmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{bmatrix} \right| = 0$$

Eigenvalues turn out to be solutions to the extended characteristic equation.

Some useful properties of Eigenvalues:

$\lambda_1\lambda_2\cdots\lambda_n = |\mathbf{A}|$ and $\lambda_1 + \lambda_2 + \ldots + \lambda_n = trace(\mathbf{A})$

Matlab command eig(A) gives eigenvalues of **A**

Eigenvectors

$$\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i, or: (\mathbf{A} - \lambda_i\mathbf{I})\mathbf{v}_i = \mathbf{0}$$

$\mathbf{v}_i$'s, **0** are column vectors of length n.

Definitions:

M: Modal matrix; columns are eigenvectors

$$\mathbf{M} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}$$

$\boldsymbol{\Lambda}$: Diagonal matrix of eigenvalues

$$\boldsymbol{\Lambda} \equiv \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

Matlab command [M,lambda]=eig(A):

'M': modal matrix of **A**

'lambda': Diagonal matrix of eigenvalues of **A**

Diagonalization

Form matrices with each column reproducing a term in the definition of Eigenvectors:

$$\mathbf{AM} = \mathbf{M}\boldsymbol{\Lambda}$$

If **M** is non-singular, $\mathbf{M}^{-1}$ exists

M is non-singular when λ's are unique (not repeated).

Premultiply by $\mathbf{M}^{-1}$:

$$\boldsymbol{\Lambda} = \mathbf{M}^{-1}\mathbf{AM}$$

Matlab Calculations

Matlab command [M,lambda]=eig(A):

'M': modal matrix of **A**

'lambda': Diagonal matrix of eigenvalues of **A**

Application of Diagonalization to Linear Systems

Consider the system: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$

Define a transformed variable: $\mathbf{y} \equiv \mathbf{M}^{-1}\mathbf{x}$

Obviously: $\mathbf{x} = \mathbf{M}\mathbf{y}$

Therefore: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} = \mathbf{M}\dot{\mathbf{y}}$

Rearrange: $\dot{\mathbf{y}} = \mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{A}\mathbf{M}\mathbf{y}$

$$\dot{\mathbf{y}} = \Lambda\mathbf{y}$$

How does this help? These are just unrelated first-order linear differential equations.

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \vdots \\ \dot{y}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 y_1 \\ \lambda_2 y_2 \\ \vdots \\ \lambda_n y_n \end{bmatrix}$$

Solution:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} y_1(0)e^{\lambda_1 t} \\ y_2(0)e^{\lambda_2 t} \\ \vdots \\ y_n(0)e^{\lambda_n t} \end{bmatrix}$$

Matlab notation:

```
t=sym('t')

z=exp(diag(lambda)*t)

y0=[...]

y=y0.*z
```

One problem: We generally know $\mathbf{x}(0)$ not $\mathbf{y}(0)$

Another problem: we are interested in $\mathbf{x}(t)$ not $\mathbf{y}(t)$

Retransformation of variables

$$\mathbf{x}(t) = \mathbf{My}(t)$$

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \mathbf{M} \times \begin{bmatrix} y_1(0)e^{\lambda_1 t} \\ y_2(0)e^{\lambda_2 t} \\ \vdots \\ y_n(0)e^{\lambda_n t} \end{bmatrix}$$

Matlab routine to solve this equation

```
A=[..]: Square matrix
[M,lambda]=eig(A)
t=sym('t')
z=exp(diag(lambda)*t)
x0=[..]: column vector
y0=inv(M)*x0
y=y0.*z
x=M*y
```

Also works with complex roots.

More elaborate methods required for repeated roots.

A Note on Complex Eigenvalues

Complex Eigenvalues come in conjugate pairs; i.e., $\lambda + di.\lambda - di$.

Eigenvectors also come in conjugate pairs

Example: $\begin{bmatrix} 1 \\ a + bi \end{bmatrix}, \begin{bmatrix} 1 \\ a - bi \end{bmatrix}$

Consider a 2×2 case

Solution for $x(t)$ in such an example:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = c_1 \begin{bmatrix} 1 \\ a+bi \end{bmatrix} e^{(\lambda+di)t} + c_2 \begin{bmatrix} 1 \\ a-bi \end{bmatrix} e^{(\lambda-di)t}$$

Expand:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = e^{\lambda t} \begin{bmatrix} c_1[\cos(dt) + i\sin(dt)] + c_2[\cos(-dt) + i\sin(-dt)] \\ c_1(a+bi)[\cos(dt) + i\sin(dt)] + c_2(a-bi)[\cos(-dt) + i\sin(-dt)] \end{bmatrix}$$

Recall that: $\cos(-dt) = \cos(dt)$ and $\sin(-dt) = -\sin(dt)$

Rearrange to obtain:

$$= e^{\lambda t} \begin{bmatrix} (c_1 + c_2)\cos(dt) + (c_1 - c_2)i\sin(dt) \\ [a(c_1 + c_2) + b(c_1 - c_2)i]\cos(dt) + [a(c_1 - c_2)i - b(c_1 + c_2)\sin(dt)] \end{bmatrix}$$

Now define the new constants, $k_1 \equiv (c_1 + c_2)$ and $k_2 \equiv (c_1 - c_2)i$

Therefore:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = e^{\lambda t} \begin{bmatrix} k_1\cos(dt) + k_2\sin(dt) \\ (ak_1 + bk_2)\cos(dt) + (ak_2 - bk_1)\sin(dt) \end{bmatrix}$$

Evaluate at $t = 0$ and we find that:

$x_1(0) = k_1$ and $x_2(0) = (ak_1 + bk_2)$

Solve for k_1 and k_2 and plug in to get final solution.

We are therefore assured that time functions are real, never complex.

Numerical solutions (i.e., Matlab) may obscure this fact.

Diagonalization procedure automatically handles these cases.

Higher-order ordinary differential equations

Consider $h_0\dfrac{d^n y}{dt^n} + h_1\dfrac{d^{n-1}y}{dt^{n-1}} + h_2\dfrac{d^{n-2}y}{dt^{n-2}} + \ldots + h_{t-1}\dfrac{dy}{dt} + h_n y(t) = 0$

Need to define: $x_1 = y, x_2 = \dfrac{dy}{dt}, x_3 = \dfrac{d^2y}{dt^2}, \ldots$

Example: Second-order equations as First-order System

$$\ddot{y} + a\dot{y} + by = 0$$

Define $x_1 = \dot{y}$ and $x_2 = y$. Therefore $\ddot{y} = \dot{x}_1$, $\dot{y} = x_1$, and $y = x_2$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -a & -b \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Now solve as outlined above.

Nonhomogeneous Linear Differential equations with Constant Forcing Term

Consider the system:

$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{b}$: That is;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

b_i's are constant.

Simply define: $\mathbf{z} = \mathbf{x} - \mathbf{A}^{-1}\mathbf{b}$

Therefore: $\dot{\mathbf{z}} = \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{A}(\mathbf{z} + \mathbf{A}^{-1}\mathbf{b}) - \mathbf{b}$

Rearrange: $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z}$; solve in the usual way.

Note that: $\mathbf{z}(0) = \mathbf{x}(0) - \mathbf{A}^{-1}\mathbf{b}$

Finally: $\mathbf{x}(t) = \mathbf{z}(t) + \mathbf{A}^{-1}\mathbf{b}$

Linear Approximations to Non-linear Systems

Consider: $\dot{\mathbf{x}} = \mathbf{\Omega}(\mathbf{x})$

$\mathbf{\Omega}$ is a vector function of the vector $\mathbf{x}$

Consider the neighborhood of a constant solution $\bar{\mathbf{x}}$ such that $\mathbf{\Omega}(\bar{\mathbf{x}}) = \mathbf{0}$

Linear Approximation: $\dot{\mathbf{x}} \approx \mathbf{\Omega}(\bar{\mathbf{x}}) + \frac{\partial \mathbf{\Omega}}{\partial \mathbf{x}}(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$

$\frac{\partial \mathbf{\Omega}}{\partial \mathbf{x}}$ is the matrix of partial derivatives of the vector, $\mathbf{\Omega}$, with respect to the elements of $\mathbf{x}$

$\frac{\partial \boldsymbol{\Omega}}{\partial \mathbf{x}}$ is also called the Jacobian of $\boldsymbol{\Omega}$

$$\frac{\partial \boldsymbol{\Omega}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \Omega_1}{\partial x_1} & \frac{\partial \Omega_1}{\partial x_2} & \cdots & \frac{\partial \Omega_1}{\partial x_n} \\ \frac{\partial \Omega_2}{\partial x_1} & \frac{\partial \Omega_2}{\partial x_2} & \cdots & \frac{\partial \Omega_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \Omega_n}{\partial x_1} & \frac{\partial \Omega_n}{\partial x_2} & \cdots & \frac{\partial \Omega_n}{\partial x_n} \end{bmatrix}$$

Therefore: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{b}$

where $\mathbf{A} \equiv \frac{\partial \boldsymbol{\Omega}}{\partial \mathbf{x}}(\bar{\mathbf{x}})$ and $\mathbf{b} \equiv \frac{\partial \boldsymbol{\Omega}}{\partial \mathbf{x}}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{x}}$

Matlab example:

```
syms x1 x2
y1=sym('x1^2+x2')
y2=sym('log(x1)-x2')
z=jacobian([y1,y2],[x1,x2])
Returns:
z =
[ 2*x1, 1]
[ 1/x1, -1]
```

Stability of Linear Systems

Stability requires $\lim_{t \to \infty} \mathbf{x}(t) = \mathbf{0}$ for all initial conditions.

For linear systems, we require all $\lambda_i < 0$

For complex roots, we require that the real part be negative.

Saddle Paths

When some roots are non-negative, we may have stable saddle paths.

There may be initial conditions for which $\lim_{t \to \infty} \mathbf{x}(t) = \mathbf{0}$

Saddle path analysis useful in applications of the Maximum Principle

Consider the transformed variable, $\mathbf{y}(t)$

Recall:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} y_1(0)e^{\lambda_1 t} \\ y_2(0)e^{\lambda_2 t} \\ \vdots \\ y_n(0)e^{\lambda_n t} \end{bmatrix}$$

If $y_j(0) = 0 \; \forall \lambda_j \geq 0$, then all of the $y(t)$ converge to zero. (or stay equal to zero.)

We first identify the $\lambda_j \geq 0$, and set $y_j(0) = 0$, by construction.

We may then recover sets of initial conditions for $\mathbf{x}(0) = \mathbf{My}(0)$ that imply saddle path stability.

A two-dimensional example

$$\dot{\mathbf{x}} = \mathbf{Ax} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mathbf{x}.$$

Solution can also be written as:

$$\mathbf{x}(t) = \mathbf{My}(t) = y_1(0)\mathbf{v}_1 e^{\lambda_1 t} + y_2(0)\mathbf{v}_2 e^{\lambda_2 t}$$
$$= y_1(0)\begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} e^{\lambda_1 t} + y_2(0)\begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} e^{\lambda_2 t}$$

$y_1(0)$ and $y_2(0)$ are constants to be determined by initial conditions.

v_1 and v_2 are the eigenvectors

λ_1 and λ_2 are the roots of the characteristic equation.

Now suppose that $\lambda_1 > 0$

Saddle path requires $y_1(0) = 0$

However: $x(0) = My(0) \Rightarrow \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} 0 \\ y_2(0) \end{bmatrix} = \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} y_2(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$

Typically, one of $x_1(0)$ and $x_2(0)$ is a 'state' variable.

The other of $x_1(0)$ and $x_2(0)$ is a 'jump' variable.

For example, suppose that $x_1(0) = \bar{x}_1$

$v_{21} y_2(0) = \bar{x}_1 \Rightarrow y_2(0) = \frac{\bar{x}_1}{v_{21}}$ and $x_2(0) = v_{22} y_2(0) = \frac{v_{22}}{v_{21}} \bar{x}_1$

Finally, $\mathbf{x}(t) = y_1(0)\begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} e^{\lambda_1 t} + y_2(0)\begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} e^{\lambda_2 t}$

$$= \begin{bmatrix} \bar{x}_1 \\ \frac{v_{22}}{v_{21}}\bar{x}_1 \end{bmatrix} e^{\lambda_2 t}$$

We say that motion is only along the stable eigenvector; in this case $\mathbf{v}_2$.