Test Problems and Linear ODEs

Lecture 01 - Exercise Slides

John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

02686 Scientific Computing for Differential Equations

Learning Objectives

1. Implement and solve initial value problems using Matlab
2. Model simple problems
3. Discuss and analyze the behavior of nonlinear systems
4. Discuss nonlinear phenomena

Outline

Test Problems

- Van der Pol Problem
- Prey-Predator Problem
- Lorentz Problem
- Robertson Chemical Reaction
- Brusselator
- Hovorka Model (Type 1 Diabetes)

Van der Pol Problem

Van der Pol Problem

Van der Pol problem

$$y''(t) = \mu(1 - y(t)^2)y'(t) - y(t)$$

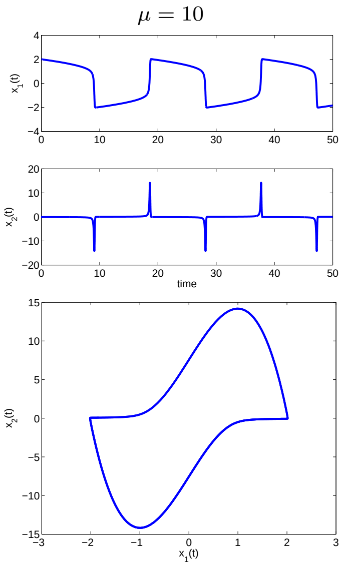
as system of first order differential equations

$$\dot{x}_1(t) = x_2(t)$$
$$\dot{x}_2(t) = \mu(1 - x_1(t)^2)x_2(t) - x_1(t)$$

with

$$y(t) = x_1(t)$$

This problem is stiff depending on the value of μ



Matlab Implementation: Non-stiff problem (μ small) - explicit solver - ode45

Model

```
function xdot = VanDerPol(t,x,mu)
% VANDERPOL  Implementation of the Van der Pol model
%
% Syntax: xdot = VanDerPol(t,x,mu)

xdot=zeros(2,1);
xdot(1) = x(2);
xdot(2) = mu*(1-x(1)*x(1))*x(2)-x(1);
```

Driver

```
mu = 10;
x0 = [2.0; 0.0];
options = odeset('RelTol',1.0e-6,'AbsTol',1.0e-6);
[T,X]=ode45(@VanDerPol,[0 5*mu],x0,options,mu);
```

Jacobian

Consider the initial value problem

$$\dot{x}(t) = f(t, x(t)) \quad x(t_0) = x_0$$

The Jacobian is

$$J(t, x(t)) = \frac{\partial f}{\partial x}(t, x(t))$$

Example: Van der Pol

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} x_2 \\ \mu(1 - x_1^2)x_2 - x_1 \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2\mu x_1 x_2 - 1 & \mu(1 - x_1^2) \end{bmatrix}$$

Matlab Implementation - Jacobian

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2\mu x_1 x_2 - 1 & \mu(1 - x_1^2) \end{bmatrix}$$

Matlab Implementation

```
function Jac = JacVanDerPol(t,x,mu)
% JACVANDERPOL  Jacobian for the Van der Pol Equation
%
% Syntax: Jac = JacVanDerPol(t,x,mu)

Jac = zeros(2,2);
Jac(2,1) = -2*mu*x(1)*x(2)-1.0;
Jac(1,2) = 1.0;
Jac(2,2) = mu*(1-x(1)*x(1));
```

Matlab Implementation: Stiff problem (μ large) - implicit solver - ode15s

Model

```
function xdot = VanDerPol(t,x,mu)
% VANDERPOL  Implementation of the Van der Pol model
%
% Syntax: xdot = VanDerPol(t,x,mu)

xdot=zeros(2,1);
xdot(1) = x(2);
xdot(2) = mu*(1-x(1)*x(1))*x(2)-x(1);
```

Driver

```
mu = 10;
x0 = [2.0; 0.0];
options = odeset('Jacobian',@JacVanDerPol,'RelTol',1.0e-6,'AbsTol',1.0e-6);
[T,X]=ode15s(@VanDerPol,[0 5*mu],x0,options,mu);
```

Prey-Predator Problem

Prey-Predator Problem

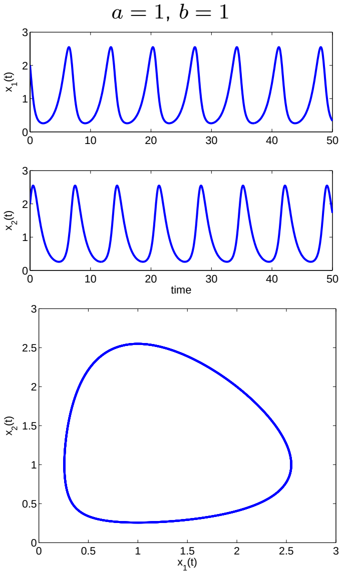
The Prey-Predator problem

$$\dot{x}_1(t) = a(1 - x_2(t))x_1(t)$$
$$\dot{x}_2(t) = -b(1 - x_1(t))x_2(t)$$

has applications in

- Biology (dynamics of preys and predators)
- Biotechnology (cell dynamics - Lotka-Volterra)

The solution is an example of a limit cycle



Matlab Implementation - explicit solver (non-stiff system)

Model

```
function xdot = PreyPredator(t,x,a,b)
% PREYPREDATOR  The Prey-Predator Model
%
% Syntax: xdot = PreyPredator(t,x,a,b)
xdot = zeros(2,1);
xdot(1) = a*(1-x(2))*x(1);
xdot(2) = -b*(1-x(1))*x(2);
```

Driver

```
a = 1;
b = 1;
x0 = [2; 2];
options = odeset('RelTol',1.0e-6,'AbsTol',1.0e-6);
[T,X]=ode45(@PreyPredator,[0 50],x0,options,a,b);
```

Matlab Implementation - implicit solver (stiff system)

Model

```
function xdot = PreyPredator(t,x,a,b)
% PREYPREDATOR  The Prey-Predator Model
%
% Syntax: xdot = PreyPredator(t,x,a,b)
xdot = zeros(2,1);
xdot(1) = a*(1-x(2))*x(1);
xdot(2) = -b*(1-x(1))*x(2);
```

Jacobian

```
function Jac = JacPreyPredator(t,x,a,b)

Jac = zeros(2,2);
Jac(1,1) = a*(1-x(2));
Jac(2,1) = b*x(2);
Jac(1,2) = -a*x(1);
Jac(2,2) = -b*(1-x(1));
```

Driver

```
a = 1;
b = 1;
x0 = [2; 2];
options = odeset('Jacobian',@JacPreyPredator,'RelTol',1.0e-6,'AbsTol',1.0e-6);
[T,X]=ode15s(@PreyPredator,[0 50],x0,options,a,b);
```

Lorentz Problem

The Lorentz Problem - The Lorentz attractor

$$\dot{x}_1(t) = \sigma(x_2(t) - x_1(t))$$
$$\dot{x}_2(t) = x_1(t)(\rho - x_3(t)) - x_2(t)$$
$$\dot{x}_3(t) = x_1(t)x_2(t) - \beta x_3(t)$$

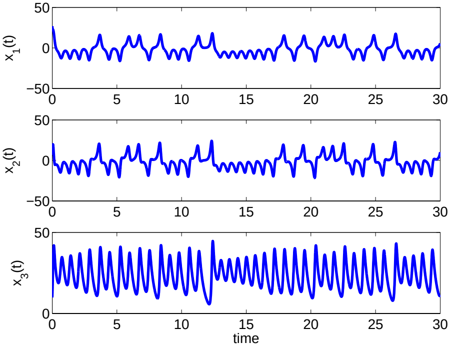
σ: Prandtl number

ρ: Rayleigh number

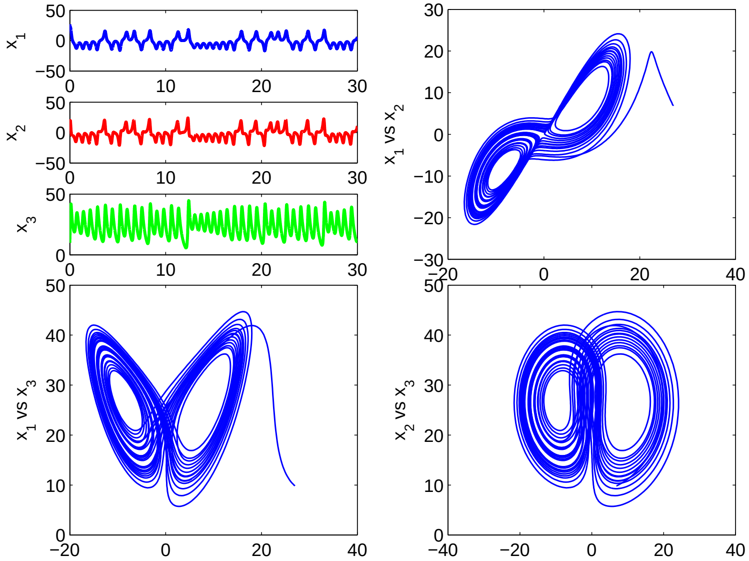
$$\sigma = 10 \quad \rho = 28 \quad \beta = 8/3$$

$$\eta = \sqrt{\rho(\beta - 1)}$$

$$x_c = \begin{bmatrix} \rho - 1 \\ \eta \\ \eta \end{bmatrix} \qquad x(0) = x_c + \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}$$



The Lorentz Problem - The Lorentz attractor



Robertson's Chemical Reaction Problem

Robertson's Chemical Reaction Problem

The chemical reactions

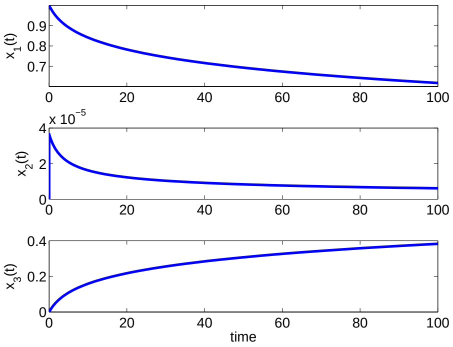
$$\begin{aligned} A &\to B & r_1 &= \alpha x_1 \\ B + B &\to C + B & r_2 &= \gamma x_2^2 \\ B + C &\to A + C & r_3 &= \beta x_2 x_3 \end{aligned}$$

can be represented by

$$\begin{aligned} \dot{x}_1 &= -\alpha x_1 + \beta x_2 x_3 & x_1(0) &= 1 \\ \dot{x}_2 &= \alpha x_1 - \beta x_2 x_3 - \gamma x_2^2 & x_2(0) &= 0 \\ \dot{x}_3 &= \gamma x_2^2 & x_3(0) &= 0 \end{aligned}$$

with parameters

$$\alpha = 0.04 \quad \beta = 1.0 \cdot 10^4 \quad \gamma = 3.0 \cdot 10^7$$

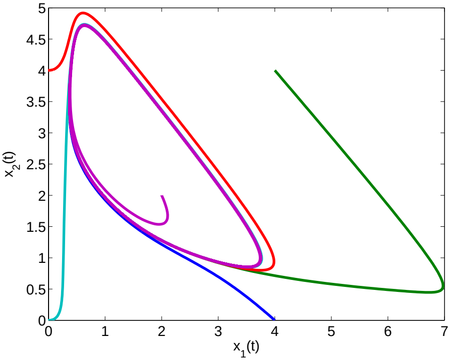


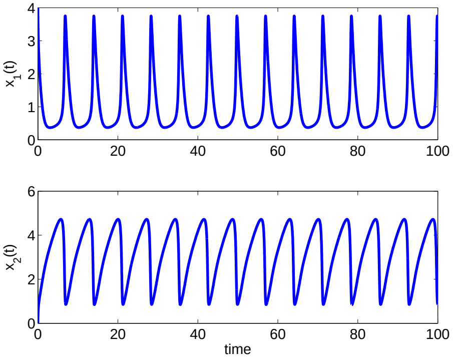
The Brusselator Problem

The Brusselator Problem

$$\dot{x}_1 = A + x_1^2 x_2 - (B+1)x_1 \qquad A = 1$$

$$\dot{x}_2 = Bx_1 - x_1^2 x_2 \qquad B = 3$$

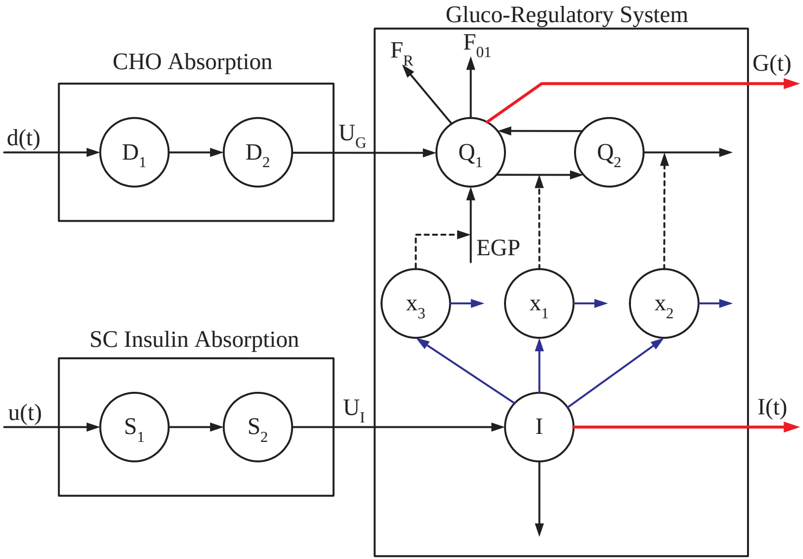




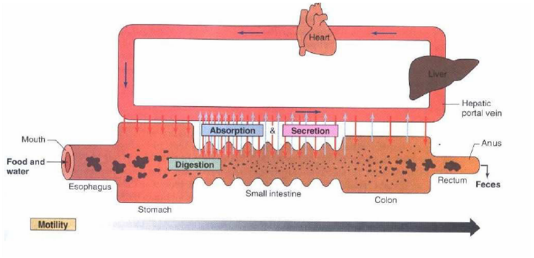
The Hovorka Model

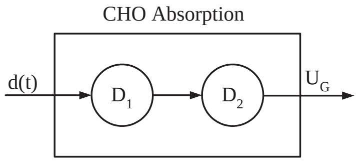
Type 1 Diabetes

The Hovorka Model for in vivo Glucose-Insulin Dynamics



Meal Model = CHO Absorption

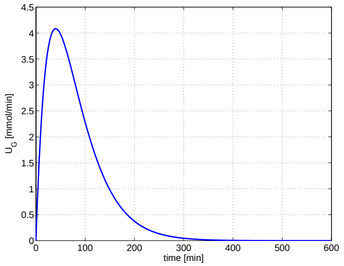




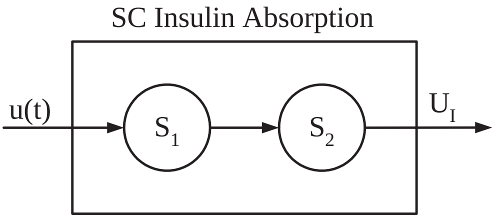
$$\frac{dD_1}{dt}(t) = A_G \frac{1000}{M_{wG}} d(t) - \frac{1}{\tau_D} D_1(t)$$

$$\frac{dD_2}{dt}(t) = \frac{1}{\tau_D} D_1(t) - \frac{1}{\tau_D} D_2(t)$$

$$U_G = \frac{1}{\tau_D} D_2(t)$$

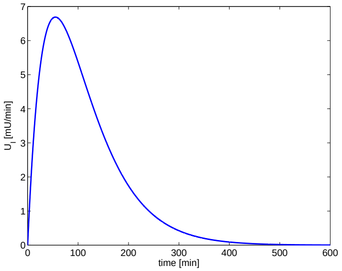


SC Insulin Absorption Model



$$\frac{dS_1}{dt}(t) = u(t) - \frac{1}{\tau_S} S_1(t)$$
$$\frac{dS_2}{dt}(t) = \frac{1}{\tau_S} S_1(t) - \frac{1}{\tau_S} S_2(t)$$

$$U_I(t) = \frac{1}{\tau_S} S_2(t)$$



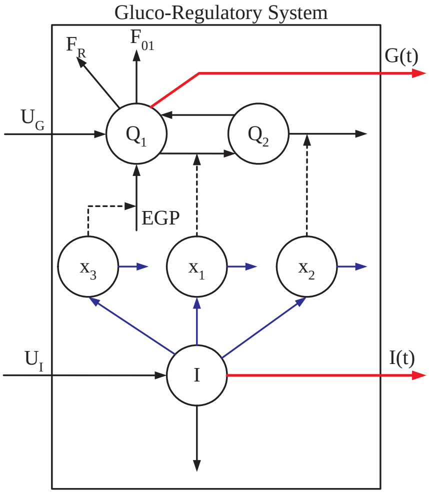
Glucose Subsystem

Plasma glucose

$$\frac{dQ_1}{dt}(t) = U_G(t) - F_{01,c}(t) - F_R(t)$$
$$- x_1(t)Q_1(t)$$
$$+ k_{12}Q_2(t)$$
$$+ EGP_0(1 - x_3(t))$$

Glucose in muscle and adipose tissue

$$\frac{dQ_2}{dt}(t) = x_1(t)Q_1(t)$$
$$- k_{12}Q_2(t)$$
$$- x_2(t)Q_2(t)$$



Glucose Consumption

Plasma Glucose Concentration

$$G(t) = \frac{Q_1(t)}{V_G}$$

Insulin independent glucose consumption (CNS)

$$F_{01,c}(t) = \begin{cases} F_{01} & G(t) \geq 4.5 \text{ mmol/L} \\ F_{01}G(t)/4.5 & \text{otherwise} \end{cases}$$

Renal Excretion

$$F_R(t) = \begin{cases} 0.003(G(t) - 9)V_G & G(t) \geq 9 \text{ mmol/L} \\ 0 & \text{otherwise} \end{cases}$$

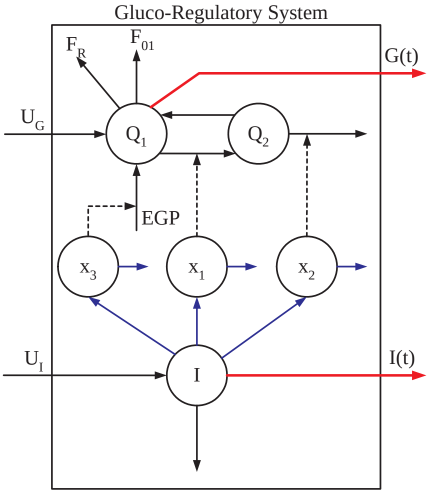
Insulin Sub System

Plasma insulin concentration

$$\frac{dI}{dt}(t) = \frac{U_I(t)}{V_I} - k_e I(t)$$

Insulin action

$$\frac{dx_1}{dt}(t) = -k_{a1}x_1(t) + k_{b1}I(t)$$
$$\frac{dx_2}{dt}(t) = -k_{a2}x_2(t) + k_{b2}I(t)$$
$$\frac{dx_3}{dt}(t) = -k_{a3}x_3(t) + k_{b3}I(t)$$



Parameters: $BW = 70$ kg, $M_{wG} = 180.1577$ g/mol

	Symbol	Value	Unit
Transfer rate	k_{12}	0.066	1/min
Deactivation rate	k_{a1}	0.006	1/min
Deactivation rate	k_{a2}	0.06	1/min
Deactivation rate	k_{a3}	0.03	1/min
Insulin elimination rate	k_e	0.138	1/min
CHO absorption constant	τ_D	40	min
Insulin absorption constant	τ_S	55	min
CHO utilization	A_G	0.8	-
Transport insulin sensitivity	$S_{I,1} = \frac{k_{b1}}{k_{a1}}$	$51.2 \cdot 10^{-4}$	L/mU
Disposal insulin sensitivity	$S_{I,2} = \frac{k_{b2}}{k_{a2}}$	$8.2 \cdot 10^{-4}$	L/mU
EGP insulin sensitivity	$S_{I,3} = \frac{k_{b3}}{k_{a3}}$	$520 \cdot 10^{-4}$	L/mU
Insulin distribution volume	$\frac{V_I}{BW}$	0.12	L/kg
Glucose distribution volume	$\frac{V_G}{BW}$	0.16	L/kg
Liver glucose production	$\frac{EGP_0}{BW}$	0.0161	$\frac{\text{mmol}}{\text{min}}$/kg
CNS glucose consumption	$\frac{F_{01}}{BW}$	0.0097	$\frac{\text{mmol}}{\text{min}}$/kg

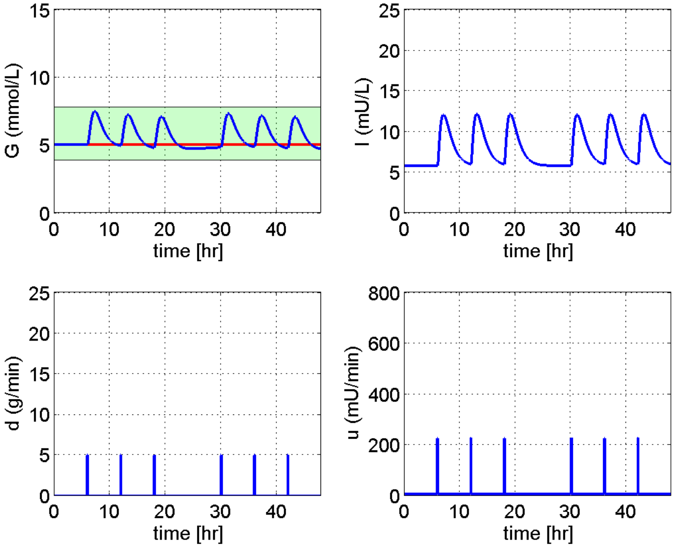
Simulation Scenario

- Constant basal rate of insulin, 6.68 mU/min (from pump or emulating long acting insulin)
- Sampling time, $T_s = 5.0$ min
- Varying meal sizes and corresponding insulin boluses

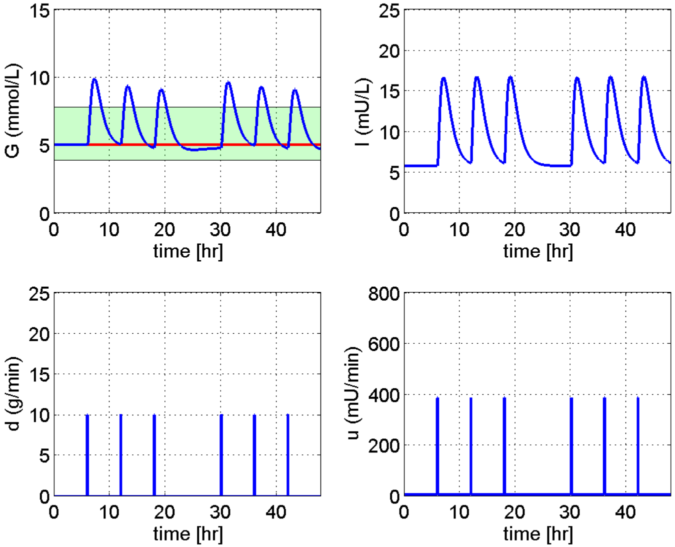
Matlab Implementation

```
function [Tx,G,I,X]=HovorkaModelSimulation(T,x0,U,D,par)
% HOVORKAMODELSIMULATION   Simulation using the Hovorka model
%
% Syntax: [Tx,G,I,X]=HovorkaModelSimulation(T,x0,U,D,par)

options = odeset('RelTol',1e-6,'AbsTol',1e-6);
nx = length(x0);
N = length(T);
Tx(1) = T(1);
X = x0';
for k=1:N-1
    x = X(end,:)';
    [Tk,Xk]=ode45(@HovorkaModel,[T(k) T(k+1)],x,options,U(:,k),D(:,k),par);
    X  = [X; Xk];
    Tx = [Tx; Tk];
end

G = X(:,5)/par.VG;
I = X(:,7);
```

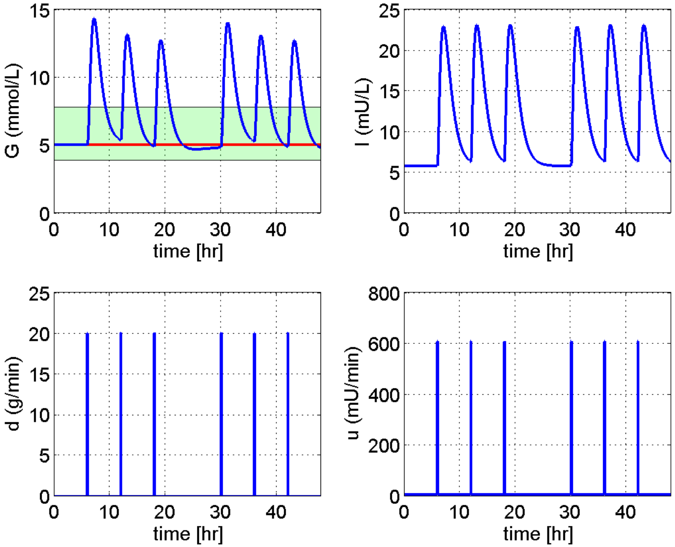
25 g CHO meals, insulin bolus 1100 mU, Ts = 5.0 min



50 g CHO meals, insulin bolus 1900 mU, Ts = 5.0 min



100 g CHO meals, insulin bolus 3000 mU, Ts = 5.0 min



Linear Systems

Linear System - Real λ

Consider the linear system

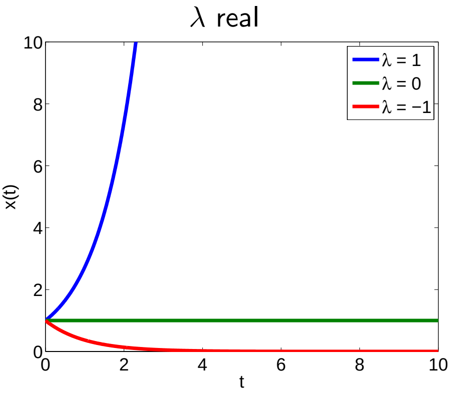
$$\frac{dx}{dt}(t) = \lambda x(t) \qquad x(0) = x_0$$
$$t \in [0, \infty[, \quad x \in \mathbb{R}, \quad \lambda \in \mathbb{R}$$

which can also be written as

$$\dot{x}(t) = \lambda x(t) \qquad x(0) = x_0$$

The solution is

$$x(t) = \exp(\lambda t)\, x_0$$



Linear System - Complex λ

Consider the linear system

$$\frac{dx}{dt}(t) = \lambda x(t) \qquad x(0) = x_0$$
$$t \in [0, \infty[, \quad x \in \mathbb{C}, \quad \lambda \in \mathbb{C}$$

which can also be written as

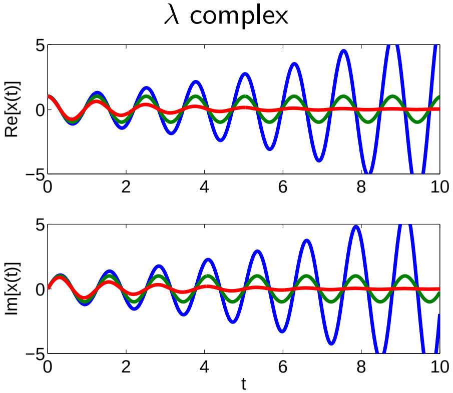
$$\dot{x}(t) = \lambda x(t) \qquad x(0) = x_0$$

The solution is

$$x(t) = \exp(\lambda t)\, x_0$$

Let $\lambda = a + ib$ then

$$x(t) = \exp(at)(\cos(bt) + i\sin(bt))\, x_0$$

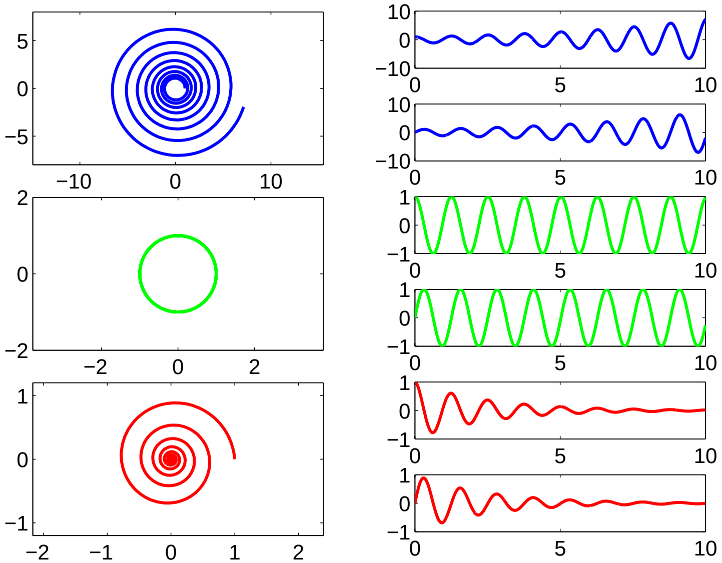


$\lambda = 0.2 + 5i \qquad \text{Re}(\lambda) > 0$

$\lambda = 0.0 + 5i \qquad \text{Re}(\lambda) = 0$

$\lambda = -0.4 + 5i \qquad \text{Re}(\lambda) < 0$

Linear System - Complex λ



Two Dimensional Linear System - Real Eigenvalues

The system of linear ODEs

$$\dot{x}(t) = Ax(t) \quad x(0) = x_0$$
$$x \in \mathbb{R}^2 \qquad A \in \mathbb{R}^{2\times 2}$$

has the solution

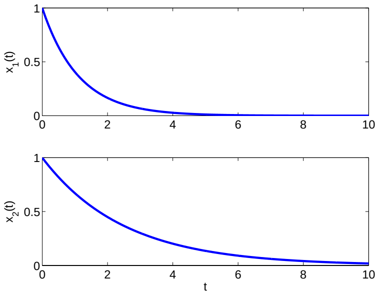
$$x(t) = \exp(At)x_0$$

Example

$$A = \begin{bmatrix} -0.9 & 0 \\ 0 & -0.4 \end{bmatrix} \quad x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

corresponds to

$$\dot{x}_1(t) = -0.9x_1(t) \qquad x_1(0) = 1$$
$$\dot{x}_2(t) = -0.4x_2(t) \qquad x_2(0) = 1$$



Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -0.9 \\ -0.4 \end{bmatrix}$$
$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2D LTI System - Stable Node

System

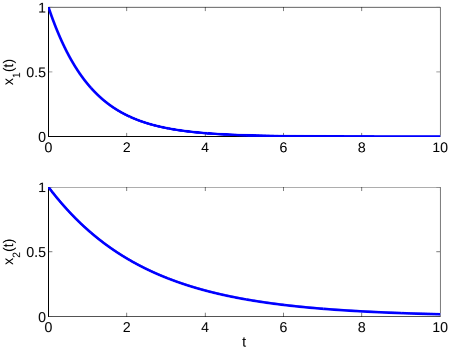
$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

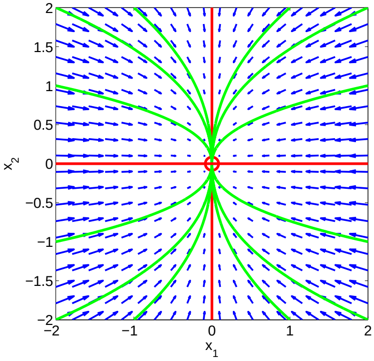
$$A = \begin{bmatrix} -0.9 & 0 \\ 0 & -0.4 \end{bmatrix}$$

Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -0.9 \\ -0.4 \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





2D LTI System - Unstable Node

System

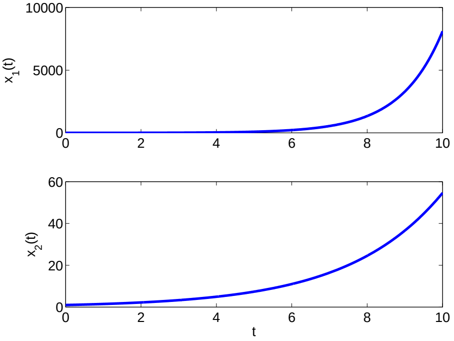
$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

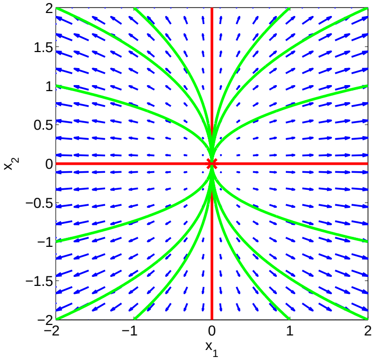
$$A = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.4 \end{bmatrix}$$

Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.4 \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





2D LTI System - Saddle Point

System

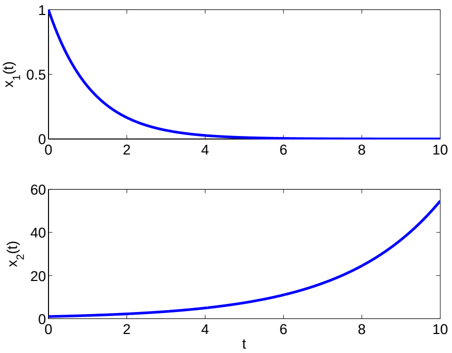
$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

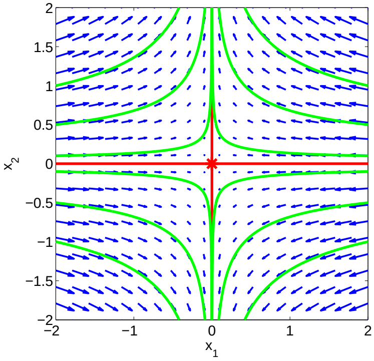
$$A = \begin{bmatrix} -0.9 & 0 \\ 0 & 0.4 \end{bmatrix}$$

Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -0.9 \\ 0.4 \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





2D LTI System - Stable Spiral

System

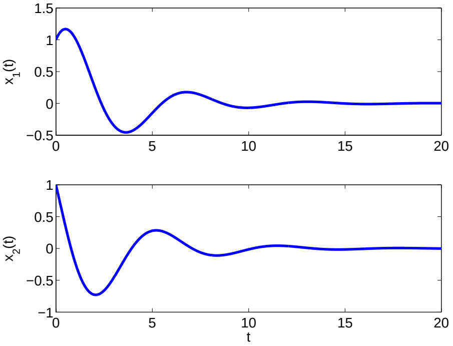
$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

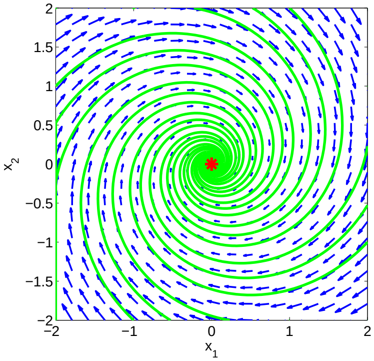
$$A = \begin{bmatrix} -0.3 & 1 \\ -1 & -0.3 \end{bmatrix}$$

Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -0.3 + i \\ -0.3 - i \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 + i\frac{1}{\sqrt{2}} & 0 - i\frac{1}{\sqrt{2}} \end{bmatrix}$$





2D LTI System - Unstable Spiral

System

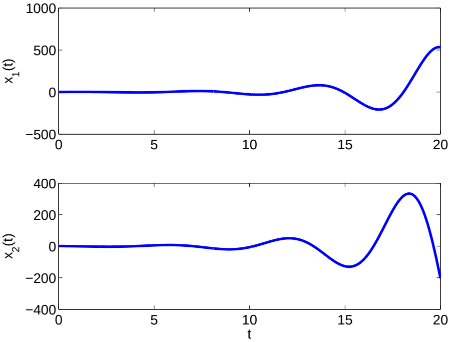
$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

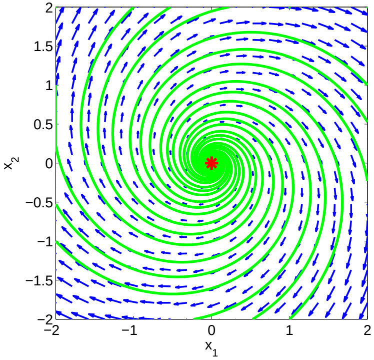
$$A = \begin{bmatrix} 0.3 & 1 \\ -1 & 0.3 \end{bmatrix}$$

Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0.3 + i \\ 0.3 - i \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 + i\frac{1}{\sqrt{2}} & 0 - i\frac{1}{\sqrt{2}} \end{bmatrix}$$





2D LTI System - Limit Cycle

System

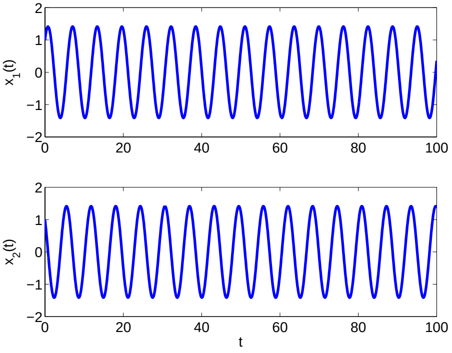
$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

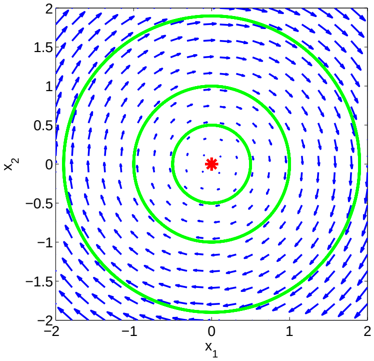
$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Eigenvalues and eigenvectors

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 + i \\ 0 - i \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 + i\frac{1}{\sqrt{2}} & 0 - i\frac{1}{\sqrt{2}} \end{bmatrix}$$





Similar Transformation

Consider the LTI system

$$\dot{x}(t) = Ax(t) \qquad x(0) = x_0$$

and the similar transformation

$$y = Tx$$

with T being non-singular such that

$$x = T^{-1}y$$

Consider the sequence of operations:

1. $\dot{x}(t) = Ax(t)$
2. $T\dot{x}(t) = TAx(t)$
3. $\frac{d}{dt}\left[Tx\right](t) = TAT^{-1}y(t)$
4. $\dot{y}(t) = \left[TAT^{-1}\right]y(t)$

Then

$$\dot{y}(t) = \bar{A}y(t) \qquad y(0) = y_0$$

with

$$\bar{A} = TAT^{-1} \qquad y_0 = Tx_0$$

Learning Objectives

1. Implement and solve initial value problems using Matlab
2. Model simple problems
3. Discuss and analyze the behavior of nonlinear systems
4. Discuss nonlinear phenomena

Questions and Comments

John Bagterp Jørgensen
jbjo@dtu.dk

Department of Applied Mathematics and Computer Science
Technical University of Denmark

Exercises

Simulation of nonlinear models:

1. Implement and simulate the Van der Pol Problem
2. Implement and simulate the Prey-Predator Problem
3. Implement and simulate the Lorentz Problem
4. Implement and simulate the Brusselator Problem
5. Implement and simulate the Hovorka Model
   (Extra exercise - a bit harder than the others)

Test equation:

1. Write the test equation as an IVP
2. What is the analytical solution of the test equation
3. Solve the test equation numerically