ORGANISATION.

- FEB 3: KALMAN FILTER & SUMMARY
- FEB 4: PROJECTS & SUMMARY
- MARCH 10: MSI SCRIPT

USE ALSO VIDEOS FOR PREPARATION

- MARCH 17, 14^{00} EXAM

FLIGHT MODELLING
FEB 2, 4, 20
10-12 (ask Gianni)

Differential Equation Types

Moritz Diehl

PROJECTS:

Overview

- Ordinary Differential Equations (ODE)
- Differential Algebraic Equations (DAE)
- Partial Differential Equations (PDE)
- Delay Differential Equations (DDE)

Comments

Ordinary Differential Equations (ODE)

- General ODE:

$$\dot{x}(t) = f(x(t), u(t), \epsilon(t), p)$$

- states $x(t)$, control inputs $u(t)$, disturbances $\epsilon(t)$, unknown parameters p (constant in time). All variables are vector valued.
- Here, $\dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\partial x}{\partial t}$. Total and partial derivative coincide as x only depends on t.
- for notational simplicity, we usually omit time dependence and write $\dot{x} = f(x, u, \epsilon, p)$
- for even simpler notation, from now on we omit $u(t), \epsilon(t)$ and p in this talk. They should be added again when necessary.
- Standard form of ODE for this talk:

$$\dot{x} = f(x)$$

Comments

Examples (what are their state vectors?)

- Pendulum
- Hot plate with pot
- Continuously Stirred Tank Reactors (CSTR)
- Robot arms
- Moving robots
- Race cars
- Airplanes in free flight

Comments

Comments

Differential Algebraic Equations (DAE)

$e^{z} + x = 0 \iff z = \log(-x)$

- similar to ODE
- besides *differential states* $x \in \mathbb{R}^{n_x}$ there are also *algebraic states* $z \in \mathbb{R}^{n_z}$
- Standard form of DAE ("semi-explicit DAE"):

$\dot{x} = f(x, z^*(x))$

$e^{z} + z + \sin(z) + x = 0$

$\dot{x} = \cos(z)$

$g(x,z)=0$

$$\boxed{\begin{aligned} \dot{x} &= f(x, z) \\ 0 &= g(x, z) \end{aligned}}$$

n_x

n_z $z^*(x)$

$g(x, z^*(x)) = 0$

- The *algebraic equations* $g(x, z) = 0$ implicitly determine z. Here, z and g have the same dimension, i.e. $g(x, z) \in \mathbb{R}^{n_z}$
- for uniqueness and numerical solvability, we usually have to assume that the Jacobian $\frac{\partial g}{\partial z} \in \mathbb{R}^{n_z \times n_z}$ is invertible ("index one")
- Index-one DAE can be solved by dedicated solvers

Comments

Equivalence of DAE with ODE (1)

- Index-one DAE can in theory be differentiated to obtain a standard ODE
- take total time derivative of algebraic equation w.r.t. time t:

$$\underline{g(x,z)=0} \quad \Rightarrow \quad \frac{\mathrm{d}g}{\mathrm{d}t}(x,z)=0$$

- right equation is equivalent to

$$\boxed{\frac{\partial g}{\partial z}}\dot{z}+\frac{\partial g}{\partial x}\dot{x}=0$$

which, because of invertibility of $\frac{\partial g}{\partial z}$ is equivalent to

$$\boxed{\dot{z}=-\left(\frac{\partial g}{\partial z}\right)^{-1}\frac{\partial g}{\partial x}f(x,z)}$$

- this procedure is called "index reduction"

Comments

Equivalence of DAE with ODE (2)

- After index reduction, we obtain an ODE
  (ODE = "DAE of index zero")

$$\boxed{\begin{aligned} \dot{x} &= f(x,z) \\ \dot{z} &= -\left(\frac{\partial g}{\partial z}\right)^{-1} \frac{\partial g}{\partial x} f(x,z) \end{aligned}}$$

- this ODE ensures that $\frac{\mathrm{d}g}{\mathrm{d}t} = 0$, i.e. the value of $g(x(t), z(t))$ remains constant along trajectories: g is an "invariant"
- algebraic equation satisfied for all t if it holds for initial value, i.e.

$$\boxed{g(x(0), z(0)) = 0}$$

,

Comments

More General DAE Formulations

1. fully implicit DAE
2. high index DAE

Comments

1) Fully Implicit DAE

- Fully-implicit DAE described by one large nonlinear equation system

$$\boxed{f(\dot{x}, x, z) = 0}$$

with $f(\dot{x}, x, z) \in \mathbb{R}^{(n_x+n_z)}$ and $\boxed{\frac{\partial f}{\partial(\dot{x},z)}}$ invertible (index-one)

- a special case of this are implicit ODE $f(\dot{x}, x) = 0$ $\boxed{\frac{\partial f}{\partial \dot{x}}}$ invertible
- often appear in mechanical or chemical applications
- Example: conservation equations like thermal energy in a basin of water given by $E(t) = k \cdot m(t) \cdot T(t)$ with heat capacity k, mass $m(t)$, and temperature $T(t)$

$$\dot{E} = k\dot{m}T + km\dot{T} = \cancel{0}$$ NET ENERGY INFLUX

Comments

Fully Implicit DAE Example in MATLAB

- Function $\boxed{\texttt{ode15i}}$ solves fully implicit DAE, all states in one vector $y = (x, z)^\top$. Grammar: $\boxed{f(t, y, \dot{y}) = 0}$
- define implicit DAE:

```
function [ resid ] = mydae( t, y, ydot )
resid=zeros(2,1);
resid(1)=ydot(1)+y(1)+y(2);
resid(2)=y(2)-sin(t);
end
```

ydot(1) = -y1 - sin(t)

y(2) = sin(t)

- create consistent initial values:

```
y0=[10;0];
ydot0=[-10;1];
```

- call solver (on time interval $[0, 10]$):

```
[tout, yout]=ode15i(@mydae,[0, 10], y0,ydot0)
plot(tout,yout)
```

Comments

2) High Index DAE

- high index DAE = DAE of "index n" with $n \geq 2$
- index refers to number of total time derivatives needed to reduce it to index zero (=ODE)
- in practice, reduction to index one DAE is enough, because good DAE solvers exist for index one e.g. MATLAB ode15i

Comments

Overview

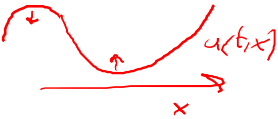
- Ordinary Differential Equations (ODE)
- Differential Algebraic Equations (DAE)
- **Partial Differential Equations (PDE)**
- Delay Differential Equations (DDE)

Partial Differential Equations (PDE)

- typically arise from spatially distributed parameters (PDE = "distributed parameter systems")
- involve partial derivatives of several variables, not only w.r.t. of time t, but also with respect of spatial coordinates x
- often, solution is called $u(t, x)$

  Attention: x and u have totally different meanings here than otherwise!
- easiest example: heat (diffusion) equation in one dimension:

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}$$



with D diffusion constant

Comments

PDE (contd.)

- need to specify "boundary conditions" in space and "initial conditions" at time zero (i.e. $u(x, 0)$)
- initial conditions are given by a profile in space: we have, loosely speaking, infinitely many states!
- can be discretized by e.g. Finite Element Method (FEM), Finite Volumes, Finite Differences
- often, only spatial derivatives are discretized, but time derivatives remain so that ODE solver can be used ("method of lines")

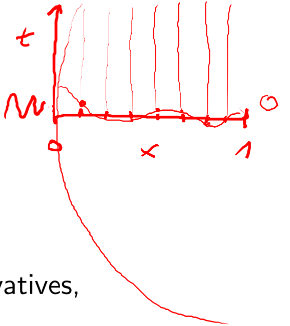
Comments

PDE Examples

- temperature profile in a microchip, a water tank, a wall, the inner part of the earth
- airflow in a computer, around an airplane, in a building, in the atmosphere ("computational fluid dynamics", CFD)
- growth of bacteria in a petri-dish
- chemical concentrations in a tubular reactor
- ...

Comments

Example: Heat Equation

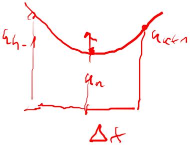


- Regard heat equation

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2}$$

  with $x \in [0,1]$. Boundary conditions:
  $u(0,t) = \sin(t), u(1,t) = 0$.
- apply "method of lines" i.e. keep time derivatives,
- apply finite differences to spatial derivatives
- use grid size $\Delta x = 1/N$, regard $u_k \approx u(k \cdot \Delta x, t)$
- obtain an ODE

$$\dot{u}_k = D\frac{(u_{k+1} - 2u_k + u_{k-1})}{(\Delta x)^2}$$



  for $k = 1, \ldots, N-1$.
- incorporate boundary conditions as

$$u_0 = \sin(t) \quad \text{and} \quad u_N = 0$$

- set initial condition e.g. to $u(x,0) = 0$.
- use stiff ODE solver `ode15s` to simulate the system

Comments

Example in MATLAB (1)

- setup ODE:

```
function [ udot] = mypde(t,u )
N=20; D=0.1; udot=zeros(N,1);
u0=sin(t);
udot(1)=N*N*D*(u0-2*u(1)+u(2));
for k=2:N-1
udot(k)=N*N*D*(u(k-1)-2*u(k)+u(k+1));
end
uN=0;
udot(N)=N*N*D*(uN-2*u(N)+uN);
```

(N-1)

Comments

Example in MATLAB (2)

- specify initial values:
  y0=zeros(20,~~0~~1);
- call ODE solver (on interval $[0, 10]$) and plot results:
  ```
  [tout,yout]=ode15s(@mypde, [0 10], y0)
  figure(1); plot(tout,yout);
  figure(2); surf(tout,linspace(0,1,20),yout')
  ```

$$\boxed{\dot{x} = Ax}$$

Comments

Overview

- Ordinary Differential Equations (ODE)
- Differential Algebraic Equations (DAE)
- Partial Differential Equations (PDE)
- **Delay Differential Equations (DDE)**

Delay Differential Equations (DDE)

- arise because of communication delays or transport phenomena
- general form with delay d:

$$\dot{x}(t) = f(x(t-d))$$

- in order to simulate system, need to know $x(t)$ on a complete interval $t \in [0, d]$. Like for PDE, we have infinitely many initial conditions!
- Can model delay by "pipe flow" PDE on interval $x \in [0, 1]$

$$\frac{\partial u}{\partial t} = -\frac{1}{d}\frac{\partial u}{\partial x}$$

with $u(0, t) = x(t)$ (input into pipe) and $x(t-d) = u(1, t)$ (output of pipe).

Approximating DDE

- Pipe flow can be approximated by spatial discretization, resulting in a sequence of first order delays ("PT1")
- Example with $x \in \mathbb{R}$:

$$\dot{x}(t) = -x(t-d)$$

- introducing N "helper states" $u_1, \dots, u_N$ we get for $k = 1, \dots, N$:

$$\dot{u}_k = -\frac{N}{d}(u_k - u_{k-1})$$

  with $u_0(t) = x(t)$
- last helper state approximates delayed value, i.e. $x(t-d) \approx u_N$
- in practice, often $N = 2$ to 5 approximate real delay sufficiently accurate

Example in MATLAB

- setup ODE:

```
function [ ydot] = mydde(t, y)
d=1; N=20; ydot=zeros(N,1);
for k=2:N
    ydot(k)=-N/d*(y(k)-y(k-1));
end
ydot(1)= - y(N);
end
```

- specify initial values:

```
y0=zeros(20,0); y0(1)=1
```

- call ODE solver (on interval $[0, 10]$) and plot results:

```
[tout,yout]=ode15s(@mypde, [0 10], y0);
figure(1); plot(tout,yout);
figure(2); surf(tout,linspace(0,1,20),yout')
```

Comments

Comments

Comments

Comments

Comments

Comments

Comments