Modeling and Simulation

Partial Differential Equations

John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

02686 Scientific Computing for Differential Equations
Lecture 04A

Open-loop simulation

Nonlinear and linear systems

- Nonlinear system

$$
\begin{aligned}
x(t_0) &= x_0 \\
\dot{x}(t) &= f(t, x(t), u(t), p) \qquad && t \in [t_0,\, t_f] \\
y(t) &= g(t, x(t), p) && t \in [t_0,\, t_f]
\end{aligned}
$$

- Linear system

$$
\begin{aligned}
x(t_0) &= x_0 \\
\dot{x}(t) &= A(p)x(t) + B(p)u(t) \qquad && t \in [t_0,\, t_f] \\
y(t) &= C(p)x(t) && t \in [t_0,\, t_f]
\end{aligned}
$$

- Time discretization. Equidistant time step, Δt. N time steps.

$$
\begin{aligned}
&t_0 < t_1 < t_2 < \ldots < t_k < t_{k+1} < ... < t_{N-1} < t_N = t_f \\
&t_{k+1} = t_k + \Delta t, \qquad k = 0, 1, \ldots, N-1
\end{aligned}
$$

- Piecewise constant input (zero-order-hold, ZOH)

$$
u(t) = u_k \qquad t_k \leq t < t_{k+1} \quad k = 0, 1, \ldots, N-1
$$

- Computational task
  Given t_0, x_0, $\{u_k\}_{k=0}^{N-1}$, p.
  Compute: $\{x_k = x(t_k), y_k = y(t_k)\}_{k=0}^{N}$.

Nonlinear system - explicit and implicit Euler methods

- Nonlinear system

$$\begin{aligned} x(t_0) &= x_0 \\ \dot{x}(t) &= f(t, x(t), u(t), p) && t \in [t_0,\, t_f] \\ y(t) &= g(t, x(t), p) && t \in [t_0,\, t_f] \end{aligned}$$

  Piecewise constant input (zero-order-hold, ZOH)

$$u(t) = u_k \qquad t_k \leq t < t_{k+1} \qquad k = 0, 1, \ldots, N-1$$

- Explicit Euler

$$\begin{aligned} x_0 &= \bar{x}_0 & t_0 &= \bar{t}_0 \\ x_{k+1} &= x_k + f(t_k, x_k, u_k, p)\Delta t & t_{k+1} &= t_k + \Delta t & k &= 0, 1, \ldots, N-1 \\ y_k &= g(t_k, x_k, p) & & & k &= 0, 1, \ldots, N \end{aligned}$$

- Implicit Euler

$$\begin{aligned} x_0 &= \bar{x}_0 & t_0 &= \bar{t}_0 \\ R_k(x_{k+1}) &= x_{k+1} - f(t_{k+1}, x_{k+1}, u_k, p)\Delta t - x_k = 0 & t_{k+1} &= t_k + \Delta t \\ y_k &= g(t_k, x_k, p) \end{aligned}$$

Nonlinear system

- Nonlinear system

$$
\begin{aligned}
x(t_0) &= \bar{x}_0 \\
\dot{x}(t) &= f(t, x(t), u(t), p) \\
y(t) &= g(t, x(t), p)
\end{aligned}
$$

  Piecewise constant input (zero-order-hold, ZOH)

$$
u(t) = u_k \qquad t_k \leq t < t_{k+1}
$$

- Explicit Euler.
  Given x_0, t_0, $\{u_k\}_{k=0}^{N-1}$, p; Δt, δt, N, N_δ. Compute $\{x_k, y_k\}_{k=0}^{N}$.
  - Initial condition

$$
x_0 = \bar{x}_0 \qquad t_0 = \bar{t}_0
$$

  - Integration from t_k to $t_{k+1} = t_k + \Delta t = t_k + N_\delta \delta t$, $x_{k+1} = \Phi(t_k, x_k, u_k, p)$, $k = 0, 1, \ldots, N$:

$$
\begin{aligned}
x_{k,0} &= x_k & t_{k,0} &= t_k \\
x_{k,i+1} &= x_{k,i} + f(t_{k,i}, x_{k,i}, u_k, p)\delta t & t_{k,i+1} &= t_{k,i} + \delta t \\
x_{k+1} &= x_{k,N_\delta} & t_{k+1} &= t_{k,N_\delta}
\end{aligned}
$$

  - Output

$$
y_k = g(t_k, x_k, p)
$$

Linear system

- Nonlinear system

$$
\begin{aligned}
x(t_0) &= \bar{x}_0 \\
\dot{x}(t) &= A_c(p)x(t) + B_c(p)u(t) && t \in [t_0,\, t_f] \\
y(t) &= C(p)x(t) && t \in [t_0,\, t_f]
\end{aligned}
$$

Piecewise constant input (zero-order-hold, ZOH)

$$
u(t) = u_k \qquad t_k \le t < t_{k+1}
$$

- Compute discrete time matrices

$$
\begin{aligned}
&\text{Explicit Euler :} && A = I + A_c(p)\Delta t && B = B_c(p)\Delta t \\
&\text{Implicit Euler :} && A = (I - A_c(p)\Delta t)^{-1} && B = (I - A_c(p)\Delta t)^{-1} B_c(p)\Delta t
\end{aligned}
$$

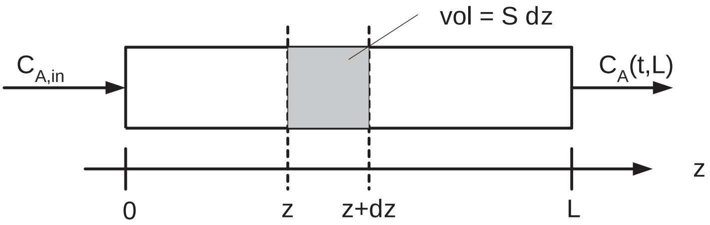
$$
\text{Exact (expm) :} \qquad \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c(p) & B_c(p) \\ 0 & 0 \end{bmatrix} \Delta t \right)
$$

- Simulate in discrete time

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k
\end{aligned}
$$

Advection-Diffusion-Reaction PDE

Flow in a Pipe - Advection



Flux for advective flow: $N_A = vC_A$

Mass balance

$$Accumulated = Influx - Outflux$$

$$Accumulated = [C_A(t + \Delta t, z) - C_A(t, z)]\, S\Delta z$$
$$Influx = N_A(t, z)S\Delta t$$
$$Outflux = N_A(t, z + \Delta z)S\Delta t$$

Advection PDE

$$\frac{\partial C_A}{\partial t} = -\frac{\partial N_A}{\partial z} = -v\frac{\partial C_A}{\partial z}$$

The model

- Equidistant spatial discretization

$$z_j = \left(j - \frac{1}{2}\right)\Delta z \qquad \Delta z = \frac{L}{N_z}$$

- Spatial discretization of partial differential equation (method of lines)

$$\frac{dC_{A,j}(t)}{dt} = -\frac{N_{A,j+1/2}(t) - N_{A,j-1/2}(t)}{\Delta z} \qquad j = 1, 2, \ldots, N_z$$

- Fluxes

$$\begin{aligned} N_{A,j+1/2}(t) &= vC_{A,in}(t) && j = 0 \\ N_{A,j+1/2}(t) &= vC_{A,j}(t) && j = 1, 2, \ldots, N_z - 1 \\ N_{A,j+1/2}(t) &= vC_{A,j}(t) && j = N_z \end{aligned}$$

can be represented as

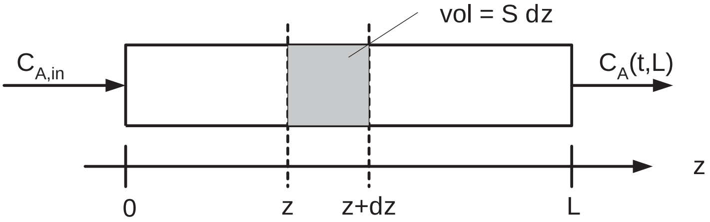
$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), p) \qquad x(t_0) = x_0 \\ y(t) &= g(x(t), p) \end{aligned}$$

with $x = [C_{A,1};\, C_{A,2};\, \ldots, C_{A,N_z}]$, $u = C_{A,in}$, $p = v$, $y = C_{A,out} = C_{A,N_z}$

Implementation - Matlab

```
function xdot = PipeAdvection(t,x,u,p)

% unpack states, inputs, and parameters
cA   = x;
cAin = u;

n  = p.Nz;
dz = p.dz;

v  = p.v;

% advection at finite volume interfaces
NadvA           = zeros(n+1,1);
NadvA(1,1)      = v*cAin;
NadvA(2:n+1,1) = v*cA(1:n,1);

% flux = advection
NA = NadvA;

% Differential Equations (mass balances at finite volumes)
cAdot = (NA(2:n+1,1)-NA(1:n,1))/(-dz);

% pack states
xdot = cAdot;
```

Flow in a Pipe - Diffusion



Flux for diffusive flow: $N_A = J_A$

Diffusion by Fick's law: $J_A = -D_A \dfrac{\partial C_A}{\partial z}$

Mass balance

$$Accumulated = Influx - Outflux$$

$$\begin{aligned} Accumulated &= \left[C_A(t+\Delta t, z) - C_A(t,z)\right] S\Delta z \\ Influx &= N_A(t,z)S\Delta t \\ Outflux &= N_A(t,z+\Delta z)S\Delta t \end{aligned}$$

Diffusion PDE

$$\frac{\partial C_A}{\partial t} = -\frac{\partial N_A}{\partial z} = D_A \frac{\partial^2 C_A}{\partial z^2}$$

The model

- Equidistant spatial discretization

$$z_j = \left(j - \frac{1}{2}\right)\Delta z \qquad \Delta z = \frac{L}{N_z}$$

- Spatial discretization of partial differential equation (method of lines)

$$\frac{dC_{A,j}(t)}{dt} = -\frac{N_{A,j+1/2}(t) - N_{A,j-1/2}(t)}{\Delta z} \qquad j = 1, 2, \ldots, N_z$$

- Fluxes

$$\begin{aligned} N_{A,j+1/2}(t) &= 0 && j = 0 \\ N_{A,j+1/2}(t) &= -D_A \frac{C_{A,j+1}(t) - C_{A,j}(t)}{\Delta z} && j = 1, 2, \ldots, N_z - 1 \\ N_{A,j+1/2}(t) &= 0 && j = N_z \end{aligned}$$

can be represented as

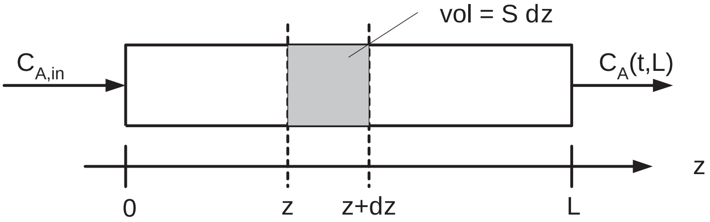
$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), p) \qquad x(t_0) = x_0 \\ y(t) &= g(x(t), p) \end{aligned}$$

with $x = [C_{A,1};\, C_{A,2};\, \ldots, C_{A,N_z}]$, $u = C_{A,in}$, $p = D_A$, $y = C_{A,out} = C_{A,N_z}$

Implementation - Matlab

```
function xdot = PipeDiffusion(t,x,u,p)

% unpack states, inputs, and parameters
cA   = x;
cAin = u; % NOT used in this model but kept for generality

n  = p.Nz;
dz = p.dz;

DA = p.DA;

% diffusion at finite volume interfaces
JA          = zeros(n+1,1);
JA(2:n,1)   = (-DA/dz)*(cA(2:n,1)-cA(1:n-1,1));

% flux = diffusion
NA = JA;

% Differential Equations (mass balances at finite volumes)
cAdot = (NA(2:n+1,1)-NA(1:n,1))/(-dz);

% pack states
xdot = cAdot;
```

Flow in a Pipe - Advection and Diffusion



Flux for advective and diffusive flow: $N_A = vC_A + J_A$

Diffusion by Fick's law: $J_A = -D_A \dfrac{\partial C_A}{\partial z}$

Mass balance

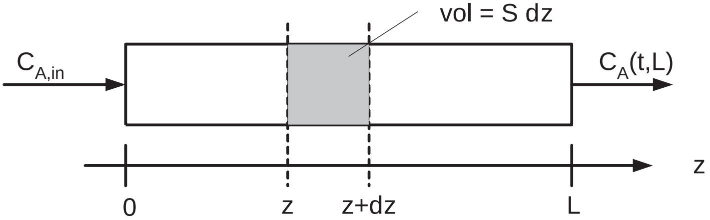
$$Accumulated = Influx - Outflux$$

$$\begin{aligned} Accumulated &= \left[C_A(t+\Delta t, z) - C_A(t,z)\right] S\Delta z \\ Influx &= N_A(t,z) S\Delta t \\ Outflux &= N_A(t, z+\Delta z) S\Delta t \end{aligned}$$

Advection-Diffusion PDE

$$\frac{\partial C_A}{\partial t} = -\frac{\partial N_A}{\partial z} = -v\frac{\partial C_A}{\partial z} + D_A \frac{\partial^2 C_A}{\partial z^2}$$

Flow and Chemical Reaction in a Pipe



Chemical Reaction: $A \rightarrow P \qquad r = kC_A \qquad R_A = -r$

Flux for convective and diffusive flow: $N_A = vC_A + J_A \qquad J_A = -D_A \dfrac{\partial C_A}{\partial z}$

$$\begin{aligned}
Accumulated &= \left[C_A(t+\Delta t, z) - C_A(t,z)\right] S\Delta z \\
Influx &= N_A(t,z)S\Delta t \\
Outflux &= N_A(t, z+\Delta z)S\Delta t \\
Generated &= R_A S\Delta z \Delta t
\end{aligned}$$

- Model (mass balance)

$$\frac{\partial C_A(t,z)}{\partial t} = -\frac{\partial N_A(t,z)}{\partial z} + R_A(t,z)$$

- Boundary conditions

$$\begin{aligned} z = 0: \qquad & N_A(t,0) = vC_{A,in} \\ z = L: \qquad & N_A(t,L) = vC_A(t,L) \end{aligned}$$

- Initial condition

$$t = 0: \qquad C_A(0,z) = C_{A0}(z)$$

- Flux

$$N_A(t,z) = \overbrace{vC_A(t,z)}^{\text{advection}} \overbrace{-D_A \frac{\partial C_A(t,z)}{\partial z}}^{\text{diffusion}}$$

- Stoichiometry and kinetics

$$A \to P \qquad r = kC_A$$

- Production rates

$$R_A = -r$$

- Equidistant spatial discretization

$$z_j = \left(j - \frac{1}{2}\right) \Delta z \qquad \Delta z = \frac{L}{N_z}$$

- Spatial discretization of partial differential equation (method of lines)

$$\frac{dC_{A,j}(t)}{dt} = -\frac{N_{A,j+1/2}(t) - N_{A,j-1/2}(t)}{\Delta z} + R_{A,j}(t) \qquad j = 1, 2, \ldots, N_z$$

- Fluxes

$$\begin{aligned}
N_{A,j+1/2}(t) &= vC_{A,in}(t) && j = 0 \\
N_{A,j+1/2}(t) &= vC_{A,j}(t) - D_A \frac{C_{A,j+1}(t) - C_{A,j}(t)}{\Delta z} && j = 1, 2, \ldots, N_z - 1 \\
N_{A,j+1/2}(t) &= vC_{A,j}(t) && j = N_z
\end{aligned}$$

- Reaction rates

$$r_j(t) = kC_{A,j}(t) \qquad j = 1, 2, \ldots, N_z$$

- Production rates

$$R_{A,j}(t) = -r_j(t) \qquad j = 1, 2, \ldots, N_z$$

The model

- Equidistant spatial discretization

$$z_j = \left(j - \frac{1}{2}\right)\Delta z \qquad \Delta z = \frac{L}{N_z}$$

- Spatial discretization of partial differential equation (method of lines)

$$\frac{dC_{A,j}(t)}{dt} = -\frac{N_{A,j+1/2}(t) - N_{A,j-1/2}(t)}{\Delta z} + R_{A,j}(t) \qquad j = 1, 2, \ldots, N_z$$

- Fluxes

$$\begin{aligned}
N_{A,j+1/2}(t) &= vC_{A,in}(t) && j = 0 \\
N_{A,j+1/2}(t) &= vC_{A,j}(t) - D_A \frac{C_{A,j+1}(t) - C_{A,j}(t)}{\Delta z} && j = 1, 2, \ldots, N_z - 1 \\
N_{A,j+1/2}(t) &= vC_{A,j}(t) && j = N_z
\end{aligned}$$

- Reaction rates

$$r_j(t) = kC_{A,j}(t) \qquad j = 1, 2, \ldots, N_z$$

- Production rates

$$R_{A,j}(t) = -r_j(t) \qquad j = 1, 2, \ldots, N_z$$

can be represented as

$$\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), p) \qquad x(t_0) = x_0 \\
y(t) &= g(x(t), p)
\end{aligned}$$

with $x = [C_{A,1};\, C_{A,2};\, \ldots; C_{A,N_z}]$, $u = C_{A,in}$, $p = [v; D_A; k]$, $y = C_{A,out} = C_{A,N_z}$

Implementation - Matlab

```matlab
function xdot = PipeAdvectionDiffusionReaction1(t,x,u,p)
cA   = x;
cAin = u;

n  = p.Nz;
dz = p.dz;

v  = p.v;
DA = p.DA;
k  = p.k;

% convection at finite volume interfaces
NconvA          = zeros(n+1,1);
NconvA(1,1)     = v*cAin;
NconvA(2:n+1,1) = v*cA(1:n,1);

% diffusion at finite volume interfaces
JA          = zeros(n+1,1);
JA(2:n,1)   = (-DA/dz)*(cA(2:n,1)-cA(1:n-1,1));

% flux = convection + diffusion
NA = NconvA + JA;

% reaction and production rates in finite volumes
r  = k*cA;
RA = -r;

% Differential Equations (mass balances at finite volumes)
cAdot = (NA(2:n+1,1)-NA(1:n,1))/(-dz) + RA;
xdot = cAdot;
```

Linear system representation

$$\frac{dC_{A,1}}{dt} = \left(\frac{v}{\Delta z}\right) C_{A,in} + \left(-\frac{v + \frac{D_A}{\Delta z}}{\Delta z} - k\right) C_{A,1} + \left(\frac{\frac{D_A}{\Delta z}}{\Delta z}\right) C_{A,2}$$

$$\frac{dC_{A,j}}{dt} = \left(\frac{v + \frac{D_A}{\Delta z}}{\Delta z}\right) C_{A,j-1} + \left(-\frac{v + 2\frac{D_A}{\Delta z}}{\Delta z} - k\right) C_{A,j} + \left(\frac{\frac{D_A}{\Delta z}}{\Delta z}\right) C_{A,j+1} \qquad j = 2, \ldots, N_z - 1$$

$$\frac{dC_{A,N_z}}{dt} = \left(\frac{v + \frac{D_A}{\Delta z}}{\Delta z}\right) C_{A,N_z-1} + \left(-\frac{v + \frac{D_A}{\Delta z}}{\Delta z} - k\right) C_{A,N_z}$$

$$\dot{x}(t) = A_c x(t) + B_c u(t)$$
$$y(t) = C_c x(t)$$

$N_z = 5$: $x = [C_{A,1};\, C_{A,2};\, C_{A,3};\, C_{A,4};\, C_{A,5}]$, $u = C_{A,in}$, $y = C_{A,out} = C_{A,5}$

$$A_c = \begin{bmatrix} -\tilde{\alpha} & \gamma & 0 & 0 & 0 \\ \beta & -\alpha & \gamma & 0 & 0 \\ 0 & \beta & -\alpha & \gamma & 0 \\ 0 & 0 & \beta & -\alpha & \gamma \\ 0 & 0 & 0 & \beta & -\tilde{\alpha} \end{bmatrix} \qquad B_c = \begin{bmatrix} \delta \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad C_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

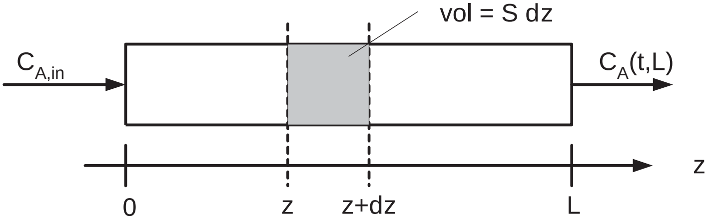
$$\alpha = \beta + \gamma + k \quad \tilde{\alpha} = \beta + k \quad \beta = \frac{v + \frac{D_A}{\Delta z}}{\Delta z} \quad \gamma = \frac{\frac{D_A}{\Delta z}}{\Delta z} \quad \delta = \frac{v}{\Delta z}$$

ZOH-discretization

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k$$

$$\begin{bmatrix} A & B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} T_s\right) \qquad C = C_c$$

Flow and Chemical Reaction in a Pipe



Chemical Reaction: $2A \to P \qquad r = kC_A^2 \qquad R_A = -2r$

Flux for convective and diffusive flow: $N_A = vC_A + J_A \qquad J_A = -D_A \dfrac{\partial C_A}{\partial z}$

$$
\begin{aligned}
Accumulated &= [C_A(t+\Delta t, z) - C_A(t,z)]\, S\Delta z \\
Influx &= N_A(t,z)S\Delta t \\
Outflux &= N_A(t, z+\Delta z)S\Delta t \\
Generated &= R_A S\Delta z\Delta t
\end{aligned}
$$

- Model (mass balance)

$$\frac{\partial C_A(t,z)}{\partial t} = -\frac{\partial N_A(t,z)}{\partial z} + R_A(t,z)$$

- Boundary conditions

$$\begin{aligned} z = 0: &\qquad N_A(t,0) = vC_{A,in} \\ z = L: &\qquad N_A(t,L) = vC_A(t,L) \end{aligned}$$

- Initial condition

$$t = 0: \qquad C_A(0,z) = C_{A0}(z)$$

- Flux

$$N_A(t,z) = \overbrace{vC_A(t,z)}^{\text{advection}} \overbrace{-D_A \frac{\partial C_A(t,z)}{\partial z}}^{\text{diffusion}}$$

- Stoichiometry and kinetics

$$2A \rightarrow P \qquad r = kC_A^2$$

- Production rates

$$R_A = -2r$$

- Equidistant spatial discretization

$$z_j = \left(j - \frac{1}{2}\right)\Delta z \qquad \Delta z = \frac{L}{N_z}$$

- Spatial discretization of partial differential equation (method of lines)

$$\frac{dC_{A,j}(t)}{dt} = -\frac{N_{A,j+1/2}(t) - N_{A,j-1/2}(t)}{\Delta z} + R_{A,j}(t) \qquad j = 1, 2, \ldots, N_z$$

- Fluxes

$$\begin{aligned}
N_{A,j+1/2}(t) &= vC_{A,in}(t) && j = 0 \\
N_{A,j+1/2}(t) &= vC_{A,j}(t) - D_A \frac{C_{A,j+1}(t) - C_{A,j}(t)}{\Delta z} && j = 1, 2, \ldots, N_z - 1 \\
N_{A,j+1/2}(t) &= vC_{A,j}(t) && j = N_z
\end{aligned}$$

- Reaction rates

$$r_j(t) = k\left(C_{A,j}(t)\right)^2 \qquad j = 1, 2, \ldots, N_z$$

- Production rates

$$R_{A,j}(t) = -2r_j(t) \qquad j = 1, 2, \ldots, N_z$$

The model

- Equidistant spatial discretization

$$z_j = \left(j - \frac{1}{2}\right)\Delta z \qquad \Delta z = \frac{L}{N_z}$$

- Spatial discretization of partial differential equation (method of lines)

$$\frac{dC_{A,j}(t)}{dt} = -\frac{N_{A,j+1/2}(t) - N_{A,j-1/2}(t)}{\Delta z} + R_{A,j}(t) \qquad j = 1, 2, \ldots, N_z$$

- Fluxes

$$\begin{aligned}
N_{A,j+1/2}(t) &= vC_{A,in}(t) && j = 0\\
N_{A,j+1/2}(t) &= vC_{A,j}(t) - D_A \frac{C_{A,j+1}(t) - C_{A,j}(t)}{\Delta z} && j = 1, 2, \ldots, N_z - 1\\
N_{A,j+1/2}(t) &= vC_{A,j}(t) && j = N_z
\end{aligned}$$

- Reaction rates

$$r_j(t) = k\left(C_{A,j}(t)\right)^2 \qquad j = 1, 2, \ldots, N_z$$

- Production rates

$$R_{A,j}(t) = -2r_j(t) \qquad j = 1, 2, \ldots, N_z$$

can be represented as

$$\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), p) \qquad x(t_0) = x_0\\
y(t) &= g(x(t), p)
\end{aligned}$$

with $x = [C_{A,1};\, C_{A,2};\, \ldots; C_{A,N_z}]$, $u = C_{A,in}$, $p = [v; D_A; k]$, $y = C_{A,out} = C_{A,N_z}$

Implementation - Matlab

```matlab
function xdot = PipeAdvectionDiffusionReaction2(t,x,u,p)
cA   = x;
cAin = u;

n  = p.Nz;
dz = p.dz;

v  = p.v;
DA = p.DA;
k  = p.k;

% convection at finite volume interfaces
NconvA          = zeros(n+1,1);
NconvA(1,1)     = v*cAin;
NconvA(2:n+1,1) = v*cA(1:n,1);

% diffusion at finite volume interfaces
JA          = zeros(n+1,1);
JA(2:n,1)   = (-DA/dz)*(cA(2:n,1)-cA(1:n-1,1));

% flux = convection + diffusion
NA = NconvA + JA;

% reaction and production rates in finite volumes
r  = k*(cA.*cA);
RA = -2.0*r;

% Differential Equations (mass balances at finite volumes)
cAdot = (NA(2:n+1,1)-NA(1:n,1))/(-dz) + RA;
xdot = cAdot;
```