



Esercitazione 2

Esercizi con Equazioni Differenziali

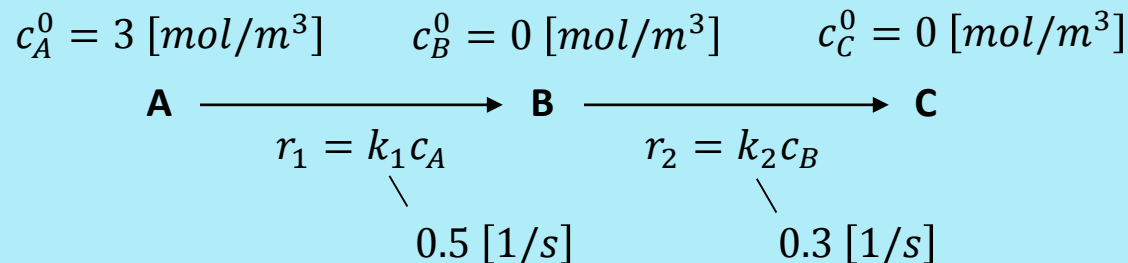
Corso di Strumentazione e Controllo di Impianti Chimici

Prof. Davide Manca

Tutor: Giuseppe Pesenti



Esempio: Esercizio 1.2 – Reazioni in serie all'interno di un reattore batch



$$\frac{dc_A}{dt} = -k_1 c_A$$

$$\frac{dc_B}{dt} = k_1 c_A - k_2 c_B$$

$$\frac{dc_C}{dt} = k_2 c_B$$

$\bar{t} ? \quad c_B(t = \bar{t}) = \max$

E' necessario risolvere un sistema di equazioni differenziali



Risolvere sistemi di equazioni differenziali:

- **Integratori ode suite**

L'obiettivo è integrare un sistema di equazioni che sono funzione di una variabile indipendente (t).

Si definisce un'**unica funzione contenente tutte le equazioni differenziali** del sistema:

```
function vettoreoutput = sistemaeq(t, vettoreinput)
...
vettoreoutput = ...
```

- riceve come input **t** e un **unico vettore** contenente le altre variabili
- ha come output un **unico vettore** con tutte le equazioni differenziali
- il vettore output della funzione deve essere un **vettore colonna**

Per utilizzare un integratore ode (**ode45**, ode23, ode113, **ode15s**, **ode23s**, ...):

```
>> [tode, yode] = ode45(handlesistemaeq, rangeintegrazionet, valoriiniziali)
```

Memorizza i risultati in un vettore per t e una matrice per le altre variabili lungo t



Esempio: Esercizio 1.2 – Reazioni in serie all'interno di un reattore batch

```
k1=0.5; % [1/s]
k2=0.3; % [1/s]
y0=[3 0 0]; % A, B, C [mol/m^3]
tspan=[0 25]; % [s]
handle=@(t,y) sistdiff(t,y,k1,k2);
[t,y]=ode45(handle,tspan,y0);
```

```
function vetdiff = sistdiff(~, vetconc, k1, k2)

vetdiff=zeros(3,1);
vetdiff(1)=-k1*vetconc(1);
vetdiff(2)=k1*vetconc(1)-k2*vetconc(2);
vetdiff(3)=k2*vetconc(2);
```

funzione equivalente, più efficiente: minore allocazione di memoria

```
function vetdiff = sistdiff(tempo, vetconc, k1, k2)

concA=vetconc(1);
concB=vetconc(2);
concC=vetconc(3);

r1=k1*concA;
r2=k2*concB;

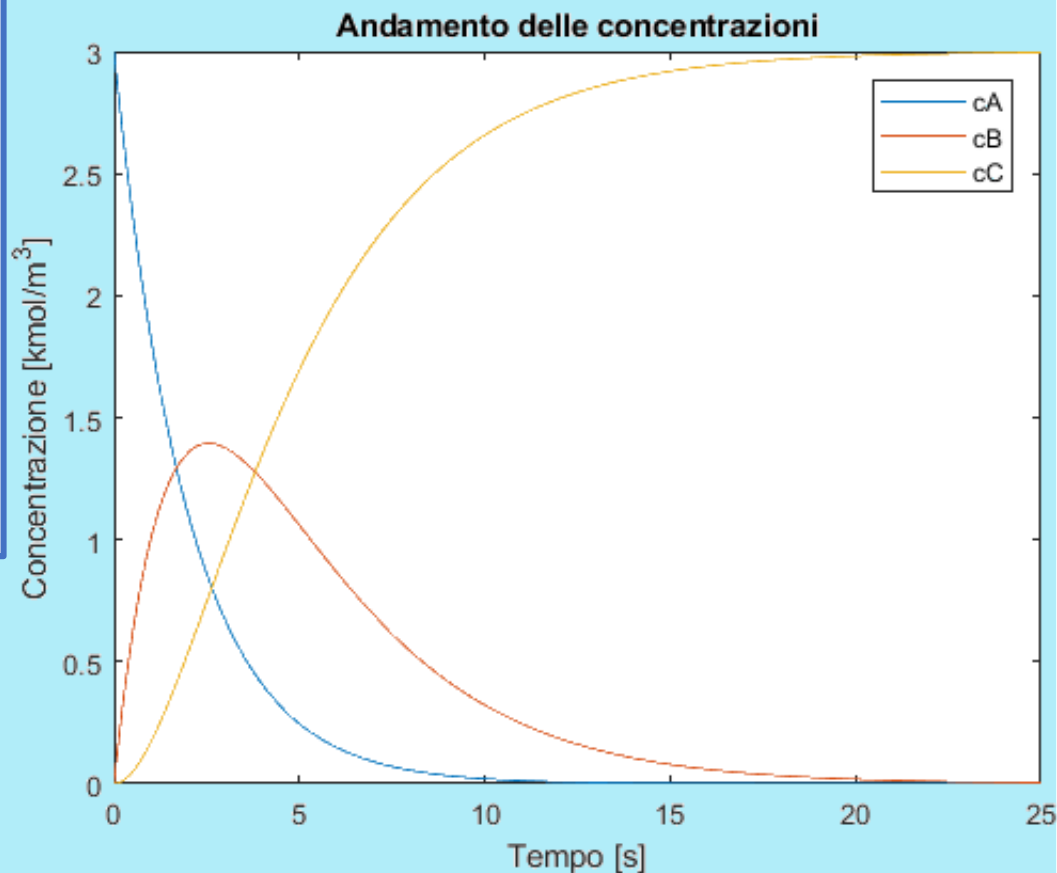
vetdiff=zeros(3,1); % l'integratore ode
                     % richiede un vettore colonna
vetdiff(1)=-r1;
vetdiff(2)=r1-r2;
vetdiff(3)=r2;
```





Esempio: Esercizio 1.2 – Reazioni in serie all'interno di un reattore batch

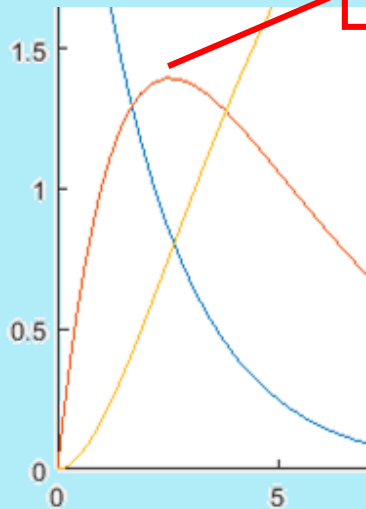
```
k1=0.5; % [1/s]
k2=0.3; % [1/s]
y0=[3 0 0]; % A, B, C [mol/m^3]
tspan=[0 25]; % [s]
handle=@(t,y) sistdiff(t,y,k1,k2);
[t,y]=ode45(handle,tspan,y0);
plot(t,y)
xlabel('Tempo [s]')
ylabel('Concentrazione [kmol/m^3]')
title('Andamento delle concentrazioni')
legend('cA','cB','cC')
```





Esempio: Esercizio 1.2 – Reazioni in serie all'interno di un reattore batch

$$\bar{t} ? \quad c_B(t = \bar{t}) = \max$$



```
cB=y(:,2);  
maxindex=find(cB==max(cB));  
maxcB=cB(maxindex);  
tmax=t(maxindex);  
disp(tmax)  
disp(maxcB)
```

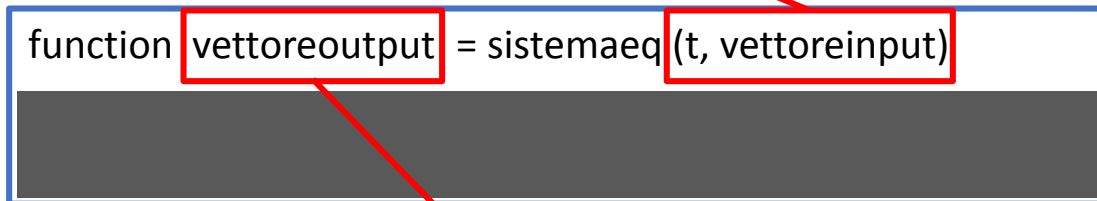
→ 2.4462
1.3931

```
disp(['cB massima di ',num2str(maxcB), ' mol/m^3 raggiunta dopo ',num2str(tmax),' s'])
```

→ cB massima di 1.3931 mol/m³ raggiunta dopo 2.4462 s

Partendo dai valori iniziali di t e delle altre variabili:

- l'integratore stabilisce la lunghezza del passo di integrazione corrente $h=\Delta t$
- a seconda del metodo, l'integratore effettua **una o più chiamate** alla funzione con input diversi
 - chiama la funzione fornendo come **input** i **valori** di t e del vettore di variabili



```
function vettoreoutput = sistemaeq(t, vettoreinput)
```

- la funzione restituisce in **output** il vettore di **valori** delle variabili
- l'integratore vede la funzione come una **black-box**
 - **può conoscere soltanto** gli **output** per gli **input** di ogni chiamata effettuata
- l'integratore arriva a **stimare il valore** delle variabili ad un passo successivo **$t+h$**



L'integratore gestisce l'**integrazione lungo il range specificato**.

A ogni step n :

- decide la lunghezza del passo di integrazione h
- stima la soluzione allo step successivo $n+1$
- **calcola l'incertezza** della soluzione trovata

$$\left. \begin{array}{l} t_n, y(t_n) \longrightarrow t_{n+h}, y(t_{n+h}) \\ t_n, y(t_n) \longrightarrow t_{n+h/2}, y(t_{n+h/2}) \longrightarrow t_{n+h}, y(t_{n+h}) \end{array} \right\} \Delta y = y_{n+1}^{one\ step} - y_{n+1}^{two\ half-steps} \approx \text{stima errore}$$

- **confronta** la stima dell'errore locale con la **tolleranza** impostata (default: 10^{-3} relativa, 10^{-6} assoluta)
 - se la tolleranza **non** è rispettata: **riduce il passo h** e riparte per cercare una nuova soluzione
 - se la tolleranza è rispettata: la **soluzione** trovata è **valida** per il presente step
 - se l'errore era estremamente ridotto, **aumenta h** per il prossimo step

Il comando **odeset** mostra le **impostazioni predefinite** degli integratori ode

>> odeset

AbsTol: [positive scalar or vector {1e-6}]

RelTol: [positive scalar {1e-3}]

NormControl: [on | {off}]

NonNegative: [vector of integers]

OutputFcn: [function_handle]

OutputSel: [vector of integers]

Refine: [positive integer]

Stats: [on | {off}]

InitialStep: [positive scalar]

MaxStep: [positive scalar]

BDF: [on | {off}]

MaxOrder: [1 | 2 | 3 | 4 | {5}]

Jacobian: [matrix | function_handle]

JPattern: [sparse matrix]

Vectorized: [on | {off}]

Mass: [matrix | function_handle]

MStateDependence: [none | {weak} | strong]

MvPattern: [sparse matrix]

MassSingular: [yes | no | {maybe}]

InitialSlope: [vector]

Events: [function_handle]

Per **creare un set di opzioni**:

>> opzioniprova=odeset();

>> disp(opzioniprova)

AbsTol: []

BDF: []

Events: []

InitialStep: []

Jacobian: []

JConstant: []

JPattern: []

Mass: []

MassSingular: []

MaxOrder: []

MaxStep: []

NonNegative: []

NormControl: []

OutputFcn: []

OutputSel: []

Refine: []

RelTol: []

Stats: []

Vectorized: []

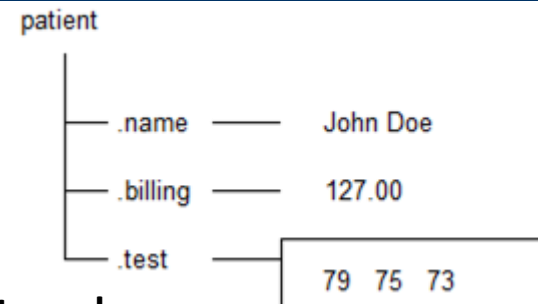
MStateDependence: []

MvPattern: []

InitialSlope: []

- Crea una **struttura** contenente come *campi* le impostazioni disponibili
- Alle impostazioni **non sono assegnati valori**

Una struttura è un **tipo di dati** che raggruppa i dati in elementi.



Per accedere a un elemento e assegnare un valore: **struttura.elemento=valore**

Esempio: nella struttura opzioni della suite ode, assegnare all'elemento AbsTol il valore 1e-12:

```
>> opzioniprova2=odeset;
```

```
>> opzioniprova2.AbsTol=1e-12;
```

```
>> disp(opzioniprova2)
```

```
>> opzioniprova3=odeset('AbsTol', 1e-12)
```

AbsTol: 1.0000e-12

BDF: []

Events: []

ecc

In alternativa, gli elementi possono essere assegnati già durante la creazione della struttura:

```
>> opzioniprova=odeset('elemento1',valore1,'elemento2',valore2)
```

La struttura con il set di opzioni può essere fornita all'integratore ode con la sintassi:

```
>> [tout, yout] = ode45(handlefunz, intervallo, valoriiniz, opzioniprova )
```

L'integratore ode utilizza le opzioni fornite

- La soluzione di ogni step deve rispettare le tolleranze **AbsTol** e **RelTol**.
- Se un elemento è **senza valore**, utilizza il valore di **default** (mostrato dal comando odeset)

Esempio: risolvere il sistema di equazioni differenziali dell'esercizio precedente 1.2 con tolleranza relativa 1e-10 e tolleranza assoluta 1e-12

```
k1=0.5; % [1/s]
k2=0.3; % [1/s]
y0=[3; 0; 0]; % A, B, C [mol/m^3]
tspan=[0 25]; % [s]
handle=@(t,y) sistdiff(t,y,k1,k2);
[t,y]=ode45(handle,tspan,y0);
```

1.3931 mol/m³ dopo 2.4462 s
i vettori t e y hanno **89 elementi**

```
k1=0.5; % [1/s]
k2=0.3; % [1/s]
y0=[3; 0; 0]; % A, B, C [mol/m^3]
tspan=[0 25]; % [s]
handle=@(t,y) sistdiff(t,y,k1,k2);
opzioniode=odeset('RelTol',1e-10,'AbsTol',1e-12);
[t,y]=ode45(handle,tspan,y0,opzioniode);
```

1.3943 mol/m³ dopo 2.5519 s
i vettori t e y hanno **1357 elementi**



Biomassa di batteri nel volume di un reattore.

B indica la concentrazione di batteri, assunta uniforme, in kmol/m^3

$$\begin{aligned} B(t) \\ B(t + \Delta t) \end{aligned}$$

$$\frac{dB(t)}{dt} = r(t)$$

$$B(t + \Delta t) - B(t) \approx r \cdot \Delta t$$

$$r(t) = k(\cdot) \cdot B(t)$$

r rappresenta il tasso di crescita della popolazione di batteri

$$r(t) = k(S) \cdot B(t)$$

I batteri per crescere e moltiplicarsi consumano un **substrato** S , con concentrazione in kmol/m^3

$k(S)$?

$$k(S) = k' \cdot S$$

$$k(S) = \frac{k_1 \cdot S}{k_2 + S} \quad \text{Eq. di Michaelis-Menten}$$

$$S \text{ piccolo: } k(S) = k' \cdot S$$

$$S \text{ piccolo: } k(S) \sim \frac{k_1}{k_2} S = k' \cdot S$$

$$S \text{ grande: } k(S) \rightarrow \infty$$

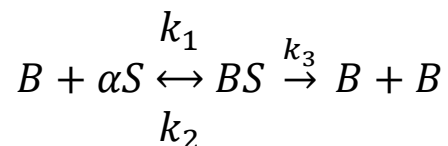
$$S \text{ grande: } k(S) \sim \frac{k_1 \cdot S}{S} = k_1$$

$$r = k' \cdot S \cdot B$$

$$r = \frac{k_1 \cdot S}{k_2 + S} \cdot B$$



Riproduzione batteri:



Rate di crescita:

$$r = k_3 \cdot c_{BS}$$

Bilancio su BS:

$$\frac{dc_{BS}}{dt} = k_1 c_B c_S - k_2 c_{BS} - k_3 c_{BS}$$

Ipotesi quasi steady-state per BS

$$\frac{dc_{BS}}{dt} \sim 0$$

$$c_{BS} = \frac{k_1 c_B c_S}{k_2 + k_3} = \frac{c_B c_S}{K_M}$$

In ogni istante:

$$c_B^{tot} = c_B + c_{BS}$$

$$K_M = \frac{k_2 + k_3}{k_1}$$

$$c_{BS} = \frac{(c_B^{tot} - c_{BS}) c_S}{K_M}$$

$$c_{BS} = \frac{c_B^{tot} \cdot c_S}{K_M + c_S}$$

$$\frac{dc_B^{tot}}{dt} = r = k_3 \cdot \frac{c_B^{tot} \cdot c_S}{K_M + c_S}$$

$$\frac{dc_S^{tot}}{dt} = -\alpha \cdot r = -\alpha \cdot k_3 \cdot \frac{c_B^{tot} \cdot c_S}{K_M + c_S}$$

$$\frac{dc_B^{tot}}{dt} = r = k_3 \cdot \frac{c_B^{tot} \cdot c_S}{K_M + c_S}$$

$$\frac{dc_S^{tot}}{dt} = -\alpha \cdot r = -\alpha \cdot k_3 \cdot \frac{c_B^{tot} \cdot c_S}{K_M + c_S}$$

$$\frac{dB}{dt} = k_1 \cdot \frac{B \cdot S}{k_2 + S}$$

$$\frac{dS}{dt} = -k_3 \cdot k_1 \cdot \frac{B \cdot S}{k_2 + S}$$

$$k_1 = 0.5 \text{ h}^{-1}$$

$$k_2 = 10^{-7} \text{ kmol/m}^3$$

$$k_3 = 0.6$$

$$B(t=0) = 1 \text{ kmol/m}^3$$

$$S(t=0) = 1 \text{ kmol/m}^3$$

Esercizio 2.1 Determinare l'evoluzione di B e S nel temp.

Un'equazione differenziale è detta **stiff** (rigida, difficile) se i metodi di risoluzione sono numericamente instabili a meno di utilizzare step di integrazione estremamente piccoli.

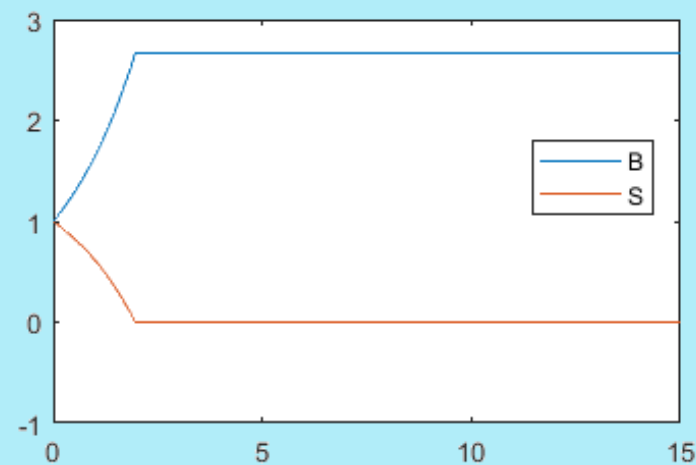
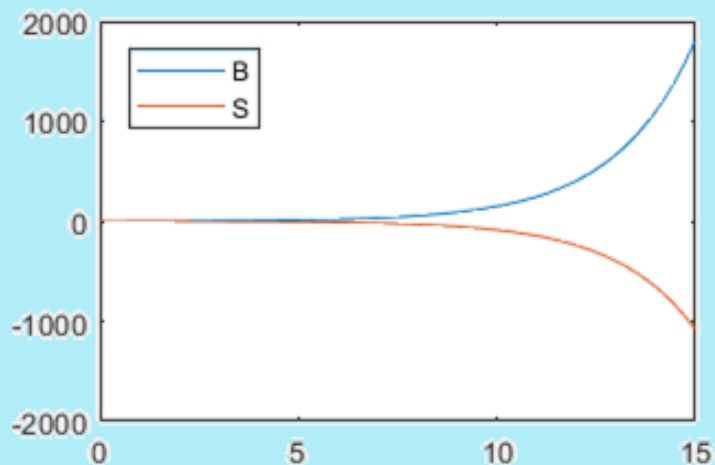
- **ode45**, **ode23**, **ode113**
- **ode15s**, **ode23s**

non adatte a problemi stiff
adatte a problemi **stiff**



Utilizzando **ode45**:

```
>> [tout,yout]=ode45(@crescitabatteri,ranget,y0);  
>> [tout,yout]=ode45(@crescitabatteri,ranget,y0,odeset('AbsTol',1e-6,'RelTol',1e-3));
```



Trovare la soluzione corretta con ode45 è in teoria possibile:

```
>> [tout,yout]=ode45(@crescitabatteri,ranget,y0,odeset('AbsTol',1e-12,'RelTol',1e-10));
```

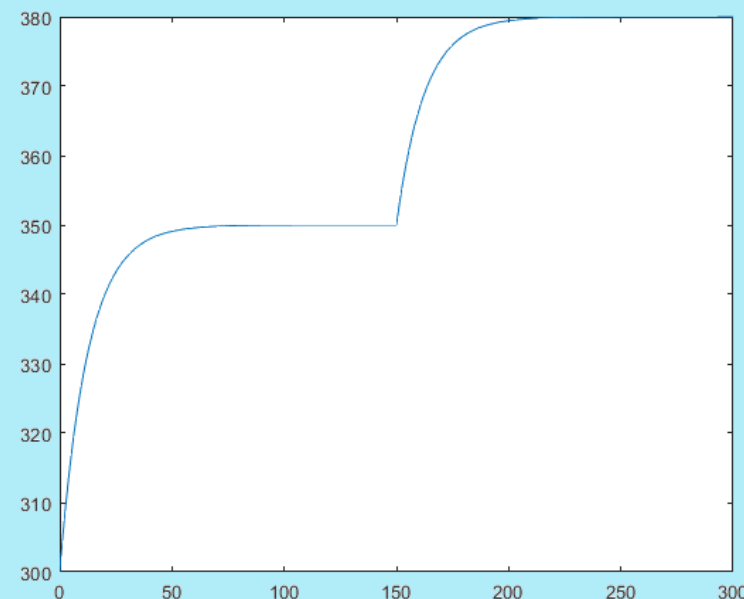
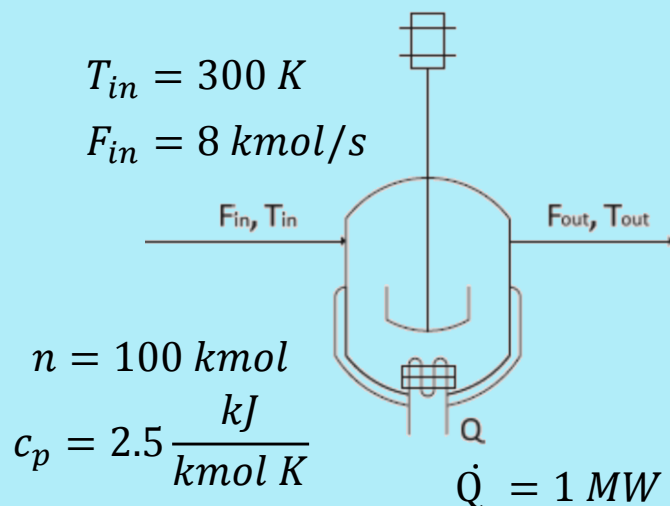
Ma ode45 è inefficiente per risolvere problema stiff.

Utilizzando **ode15s**:

```
>> [tout,yout]=ode15s(@crescitabatteri,ranget,y0,odeset('AbsTol',1e-12,'RelTol',1e-10));
```



Esercizio 2.2 – Variazione della temperatura in un serbatoio



$$\begin{cases} \frac{dn}{dt} = F_{in} - F_{out} \\ \frac{dE}{dt} \approx \frac{dH}{dt} = \dot{H}_{in} - \dot{H}_{out} + \dot{Q} \end{cases}$$

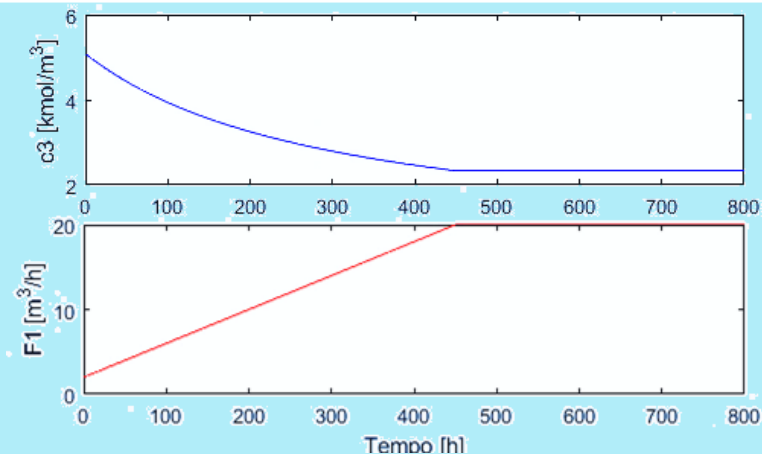
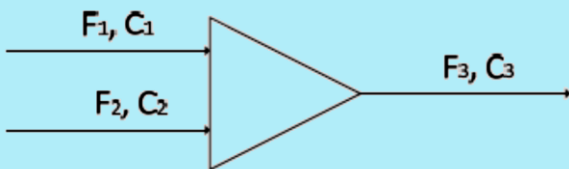
$$\frac{dn}{dt} = 0 \quad F_{in} = F_{out} = F$$

$$\frac{nc_P dT}{dt} = F_{in} c_P T_{in} - F_{out} c_P T + \dot{Q} = F c_P (T_{in} - T) + \dot{Q}$$

Studiare l'effetto di un disturbo a gradino di $+30^\circ\text{C}$ su T_{in}



Esercizio 2.3 – Miscelazione di due correnti



$$\frac{dn_{tot}}{dt} = F_1 + F_2 - F_3 = 0 \quad \text{Il miscelatore non si riempie né si svuota}$$

$$F_3 = F_1 + F_2$$

$$\frac{dn_i}{dt} = \frac{Vdc_i^3}{dt} = F_1c_i^1 + F_2c_i^2 - F_3c_i^3$$

$$V = 0.5 \text{ m}^3$$

$$c_i^1 = 0.5 \text{ kmol/m}^3$$

$$c_i^2 = 6 \text{ kmol/m}^3$$

$$c_i^3(t=0) = \frac{F_1(t=0)c_i^1 + F_2c_i^2}{F_3(t=0)}$$

$$F_2 = 10 \text{ m}^3/\text{h}$$

$$F_1(t=0) = 2 \text{ m}^3/\text{h}$$

$$F_1(t) = F_1(t=0) + 0.04 \cdot t \text{ m}^3/\text{h}$$

$$F_{1,max} = 20 \text{ m}^3/\text{h}$$

$$F_1(t) = \min(2 + 0.04 \cdot t, 20) \text{ m}^3/\text{h}$$



Esercizio 2.4 – Reazione indesiderata in un serbatoio

$$\begin{cases} \frac{dz}{dt} = \frac{\Psi}{B} (1 - z)^n \cdot h(\theta) \\ \frac{d\theta}{dt} = \Psi (1 - z)^n \cdot h(\theta) - \theta \end{cases}$$

$$C.I. \quad \begin{cases} z(0) = 0 \\ \theta(0) = 1 \end{cases}$$

$$h(\theta) = \exp\left(\frac{\theta}{1 + \varepsilon\theta}\right)$$

$$n = 1 \quad B = 20 \quad \varepsilon = 0.05$$

$$\Psi = 0.35 \div 0.65$$

Studiare la dinamica di z e θ al variare di Ψ

