

Soluzione Ese 07 – Gravity-flow tank

Strumentazione e Controllo di Impianti Chimici

Tutor: Giuseppe Pesenti

Codice base senza variabili GLOBAL

```
Dserb=12; % m
Hserb=7; % m
Dforo=1.3; % m
Hforo=0.5; % m
Cd=0.616; % -
Fin_0=8.5; % m^3/s
h_0=4.72; % m
g=9.8; % m/s^2
```

Dati del problema

```
Aserb=pi/4*Dserb^2; % m^2
Aforo=pi/4*Dforo^2; % m^2
Ac=Aforo*Cd; % m^2
```

Calcolo delle aree

```
hSS=1/(2*g)*(Fin_0/(Cd*Aforo))^2+Hforo; % m
```

Calcolo del valore di altezza
del liquido a stazionario

```
opz=odeset('RelTol',1e-8);
handleeqdiff=@(t,h)gravityflowtank(t,h,Fin_0,Ac,Aserb,g,Hforo);
[tout, hout]=ode45(handleeqdiff,[0 60*60],h_0,opz);
```

Function handle e
integrazione con ode45

```
figure(1)
plot(tout,hout,'b')
hold on
plot(tout,ones(size(hout))*hSS,'k')
```

Grafici dell'andamento nel
tempo e del valore a stazionario

```
function dhdt=gravityflowtank(t,h,Fin,Ac,Aserb,g,Hforo)
```

```
if t>=10*60 && t<20*60
    Fin=0.65*Fin;
elseif t>=40*60 && t<50*60
    Fin=0.5*Fin;
end
```

```
vc=sqrt(2*g*(h-Hforo)); % m/s
Fout=Ac*vc; % m^3/s
```

```
dhdt=(Fin-Fout)/Aserb; % m/s
```

```
end
```

Funzione contenente
l'equazione differenziale

Codice base con variabili GLOBAL

```
clear all
close all

global Fin_0 Ac Aserb g Hforo

Dserb=12; % m
Hserb=7; % m
Dforo=1.3; % m
Hforo=0.5; % m
Cd=0.616; % -
Fin_0=8.5; % m^3/s
h_0=4.72; % m
g=9.8; % m/s^2

Aserb=pi/4*Dserb^2; % m^2
Aforo=pi/4*Dforo^2; % m^2
Ac=Aforo*Cd; % m^2

hSS=1/(2*g)*(Fin_0/(Cd*Aforo))^2+Hforo; % m

opz=odeset('RelTol',1e-8);
handleeqdiff=@(t,h)gravityflowtank(t,h);
[tout, hout]=ode45(handleeqdiff,[0 60*60],h_0,opz);

figure(1)
plot(tout,hout,'b')
hold on
plot(tout,ones(size(hout))*hSS,'k')

function dhdt=gravityflowtank(t,h)

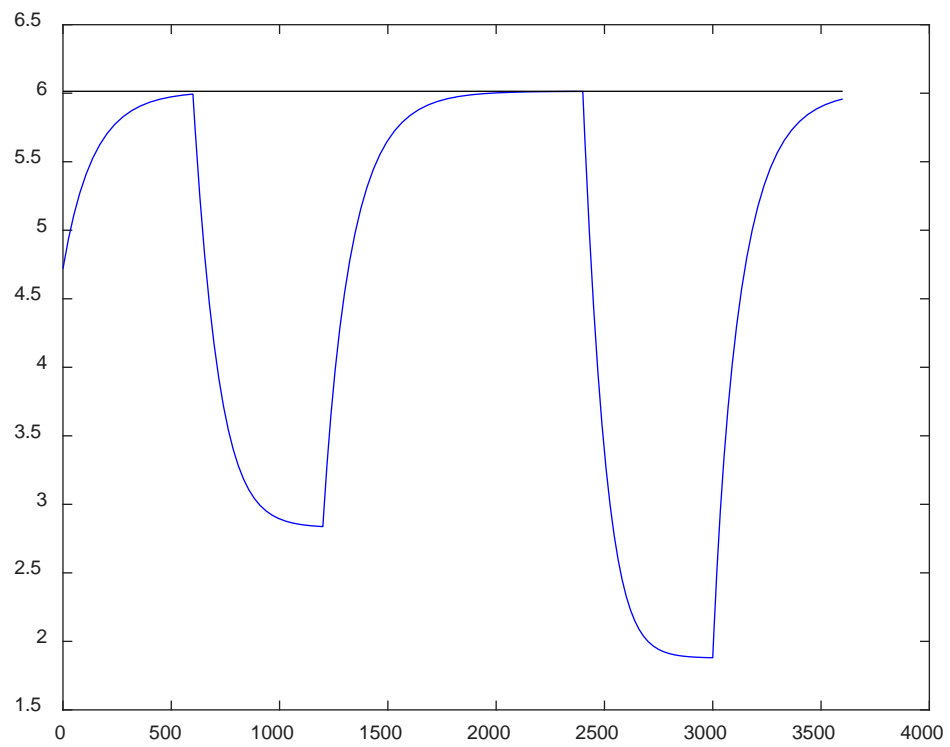
global Fin_0 Ac Aserb g Hforo

if t>=10*60 && t<20*60
    Fin=0.65*Fin_0;
elseif t>=40*60 && t<50*60
    Fin=0.5*Fin_0;
else
    Fin=Fin_0;
end

vc=sqrt(2*g*(h-Hforo)); % m/s
Fout=Ac*vc; % m^3/s

dhdt=(Fin-Fout)/Aserb; % m/s

end
```



Calcolare e diagrammare le portate a ogni step riuscito Codice con OutputFcn e variabili GLOBAL

```
clear all
close all

global Finvett Foutvett Fin_0 Hforo g Ac Aserb

Dserb=12; % m
Hserb=7; % m
Dforo=1.3; % m
Hforo=0.5; % m
Cd=0.616; % -
Fin_0=8.5; % m^3/s
h_0=4.72; % m
g=9.8; % m/s^2

Aserb=pi/4*Dserb^2; % m^2
Aforo=pi/4*Dforo^2; % m^2
Ac=Aforo*Cd; % m^2

hSS=1/(2*g)*(Fin_0/(Cd*Aforo))^2+Hforo; % m

handleout=@(t,h,flag)salvaportate(t,h,flag);
opz=odeset('RelTol',1e-8,'OutputFcn',handleout,'Refine',1);
handleeqdiff=@(t,h)gravityflowtank(t,h);
[tout, hout]=ode45(handleeqdiff,[0 60*60],h_0,opz);

figure(1)
plot(tout,hout,'b')
hold on
plot(tout,ones(size(hout))*hSS,'k')

function dhdt=gravityflowtank(t,h)
global Fin_0 Ac Aserb g Hforo

Fin=Fin_0;
if t>=10*60 && t<20*60
    Fin=0.65*Fin_0;
elseif t>=40*60 && t<50*60
    Fin=0.5*Fin_0;
end

vc=sqrt(2*g*(h-Hforo)); % m/s
Fout=Ac*vc; % m^3/s

dhdt=(Fin-Fout)/Aserb; % m/s

end
```

Function handle
dell'output function

Se Refine > 1, o se è
fornito un vettore di
valori per t e non solo gli
estremi di integrazione,
l'OutputFcn è chiamata
con più valori.

```

function status=salvaporate(t,h,flag)
global Finvett Foutvett Fin_0 Hforo g Ac

status=0;

if flag=="
    Fin=Fin_0;
    if t>=10*60 && t<20*60
        Fin=0.65*Fin_0;
    elseif t>=40*60 && t<50*60
        Fin=0.5*Fin_0;
    end
    vc=sqrt(2*g*(h-Hforo)); % m/s
    Fout=Ac*vc; % m^3/s

    figure(3)
    hold on
    plot(t,Fin,'b.')
    title('Fin')

    figure(4)
    hold on
    plot(t,Fout,'b.')
    title('Fout')

end

end

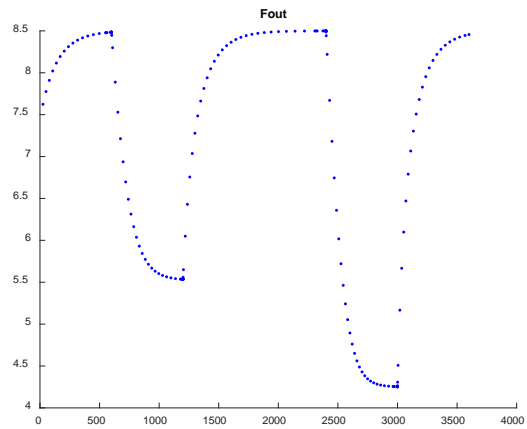
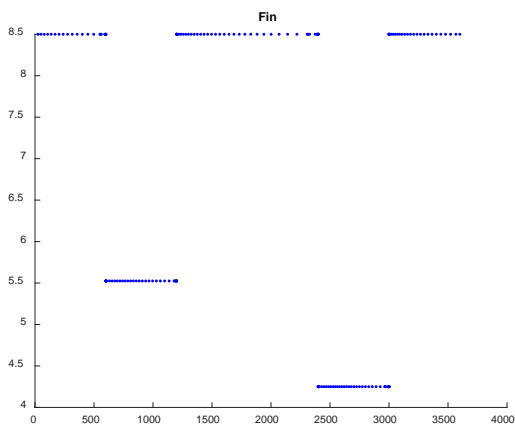
```

Funzione utilizzata come
output function

Se l'output della funzione è 1, il
risolutore ode ferma l'integrazione

Per il confronto tra stringhe si può anche
utilizzare: `if strcmp(flag,"")`

Plot "in tempo reale"



Calcolare, memorizzare e diagrammare le portate a ogni step riuscito Codice con OutputFcn e variabili GLOBAL

```
clear all
close all

global Finvett Foutvett Fin_0 Hforo g Ac Aserb

Dserb=12; % m
Hserb=7; % m
Dforo=1.3; % m
Hforo=0.5; % m
Cd=0.616; % -
Fin_0=8.5; % m^3/s
h_0=4.72; % m
g=9.8; % m/s^2

Aserb=pi/4*Dserb^2; % m^2
Aforo=pi/4*Dforo^2; % m^2
Ac=Aforo*Cd; % m^2

hSS=1/(2*g)*(Fin_0/(Cd*Aforo))^2+Hforo; % m

handleout=@(t,h,flag)salvaportate(t,h,flag);
opz=odeset('RelTol',1e-8,'OutputFcn',handleout,'Refine',1);
handleeqdiff=@(t,h)gravityflowtank(t,h);
[tout, hout]=ode45(handleeqdiff,[0 60*60],h_0,opz);

figure(1)
plot(tout,hout,'b')
hold on
plot(tout,ones(size(hout))*hSS,'k')

figure(2)
plot(tout(2:end),Finvett)
hold on
plot(tout(2:end),Foutvett)
legend('Fin','Fout')

function dhdt=gravityflowtank(t,h)
global Fin_0 Ac Aserb g Hforo

Fin=Fin_0;
if t>=10*60 && t<20*60
    Fin=0.65*Fin_0;
elseif t>=40*60 && t<50*60
    Fin=0.5*Fin_0;
end

vc=sqrt(2*g*(h-Hforo)); % m/s
Fout=Ac*vc; % m^3/s

dhdt=(Fin-Fout)/Aserb; % m/s

end
```

```

function status=salvaporate(t,h,flag)
global Finvett Foutvett Fin_0 Hforo g Ac

status=0;

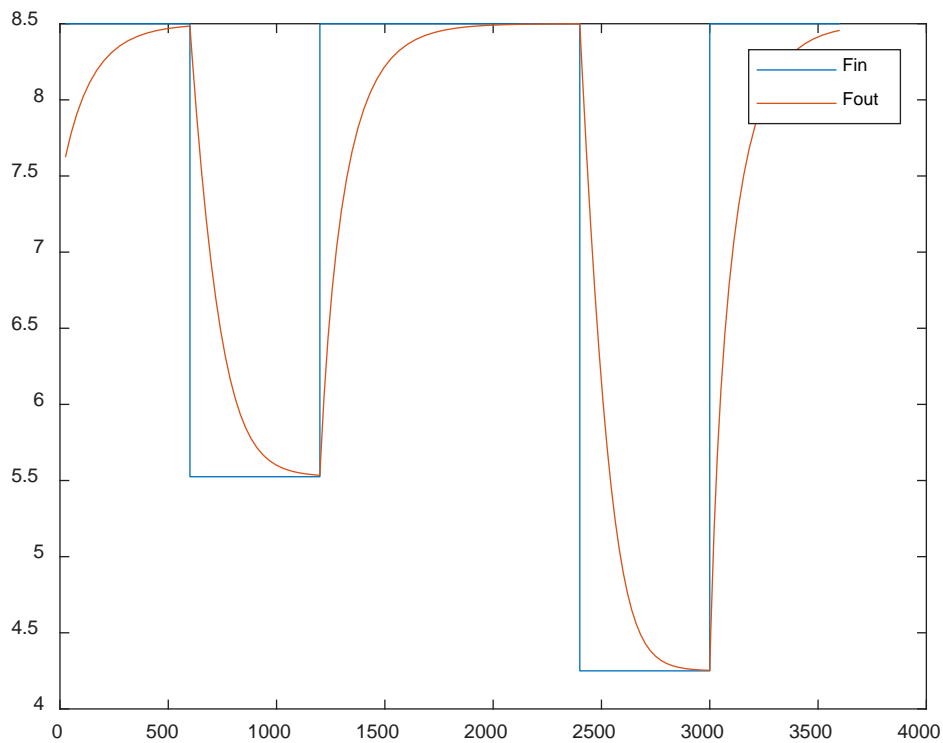
if flag=="init"
    Finvett=[];
    Foutvett=[];
elseif flag==" "
    Fin=Fin_0;
    if t>=10*60 && t<20*60
        Fin=0.65*Fin_0;
    elseif t>=40*60 && t<50*60
        Fin=0.5*Fin_0;
    end
    vc=sqrt(2*g*(h-Hforo)); % m/s
    Fout=Ac*vc; % m^3/s

    Finvett=[Finvett;Fin];
    Foutvett=[Foutvett;Fout];

end

end

```



Calcolare, memorizzare e diagrammare le portate a ogni step riuscito Codice con OutputFcn e variabili PERSISTENT

```
clear all
close all

Dserb=12; % m
Hserb=7; % m
Dforo=1.3; % m
Hforo=0.5; % m
Cd=0.616; % -
Fin_0=8.5; % m^3/s
h_0=4.72; % m
g=9.8; % m/s^2

Aserb=pi/4*Dserb^2; % m^2
Aforo=pi/4*Dforo^2; % m^2
Ac=Aforo*Cd; % m^2

hSS=1/(2*g)*(Fin_0/(Cd*Aforo))^2+Hforo; % m

handleout=@(t,h,flag)salvaportate(t,h,flag,Fin_0,Hforo,g,Ac);
opz=odeset('RelTol',1e-8,'OutputFcn',handleout,'Refine',1);
handleeqdiff=@(t,h)gravityflowtank(t,h,Fin_0,Ac,Aserb,g,Hforo);
[tout, hout]=ode45(handleeqdiff,[0 60*60],h_0,opz);

figure(1)
plot(tout,hout,'b')
hold on
plot(tout,ones(size(hout))*hSS,'k')
xlabel('t [s]')
ylabel('h [m]')

figure(2)
plot(tout(2:end),Fin_save)
hold on
plot(tout(2:end),Fout_save)
legend('Fin','Fout')
xlabel('t [s]')
ylabel('F [m^3/s]')

function dhdt=gravityflowtank(t,h,Fin,Ac,Aserb,g,Hforo)

if t>=10*60 && t<20*60
    Fin=0.65*Fin;
elseif t>=40*60 && t<50*60
    Fin=0.5*Fin;
end

vc=sqrt(2*g*(h-Hforo)); % m/s
Fout=Ac*vc; % m^3/s

dhdt=(Fin-Fout)/Aserb; % m/s

end
```



```

function status=salvaportate(t,h,flag,Fin,Hforo,g,Ac)
persistent Finvett Foutvett

status=0;

if flag=="init"
    Finvett=[];
    Foutvett=[];
elseif flag=="
    if t>=10*60 && t<20*60
        Fin=0.65*Fin;
    elseif t>=40*60 && t<50*60
        Fin=0.5*Fin;
    end
    vc=sqrt(2*g*(h-Hforo)); % m/s
    Fout=Ac*vc; % m^3/send

    Finvett=[Finvett;Fin];
    Foutvett=[Foutvett;Fout];
elseif flag=="done"
    assignin('base','Fin_save',Finvett);
    assignin('base','Fout_save',Foutvett);
    clear Finvett Foutvett
end

end

```

