```
clear all;
```

Generalized Inverse Problem for parameters estimation

Initial Problem

$$\partial_t\Psi - \nabla.(D\nabla\Psi) = G(\Psi,\alpha) + q$$

$$d = M(\Psi) + \epsilon$$

Generalized inverse resolution

$$\partial_t\Psi - \nabla.(D\nabla\Psi) = G(\Psi,\alpha) + < C_q, \overline{\lambda} >$$

$$\Psi(t_0) = \Phi_0 + < C_a, \overline{\lambda}(t_0) >$$

$$D\nabla\Psi \cdot \overrightarrow{n} = 0$$

$$\partial_t\overline{\lambda} - \nabla \cdot (D\nabla\overline{\lambda}) = -\partial_\Psi G^T \cdot \overline{\lambda} - M^T(\delta)W_\epsilon(d - M(\Psi))$$

$$\overline{\lambda}(t_k) = 0$$

$$D\nabla\overline{\lambda} \cdot \overrightarrow{n} = 0$$

$$\alpha = \alpha_0 + C_\alpha \int_{t_0}^{t_k} \partial_\alpha G^T \cdot \overline{\lambda} dt$$

Resolution

Parameters

```
global t0 tf delta_t N P x h diff_u diff_c;
t0=0e0; tf=7e0; delta_t=1e-2;   N=100;  h=1/(N-1);  P = round((tf-t0)/delta_t+1);
x = linspace(0,1,N)';
diff_u=1e-4; diff_c = 1e-3;
```

Initial condition

```
a = 1/2;b = 7/10;c = (a+b)/2;
u_0 = zeros(N,1);
c_0 = zeros(N,1);
for i=1:N
    if (abs(x(i)-c)<(b-a)/2)
        y = (x(i)-c)/((b-a)/2);
        u_0(i) = exp(1-1/(1-y^2));
    end
    c_0(i) = 0.5*u_0(i);%1e0-0.5*u_0(i);
end
L1_N = zeros(N,1);
L2_N = zeros(N,1);
Coefs_0 = log([0.1;0.05;0.0391;0.06;0.1]);
```

```
Coefs = Coefs_0;
```

Measurements

```
Measures = measurements(u_0,c_0,log([0.2,0.1,0.1,0.03,0.08]));
```

Covariances matrix and their inverse

```
C_eps = sparse(diag(1e-1*ones(2*N,1))); %Covariance Matrix of $C_epsilon$
W_eps = inv(C_eps);
C_alpha = sparse(diag((1e1)^2*ones(5,1)));%sparse(diag([2e0,2e0,6e-1,3e-1,2e-0]));
C_a = sparse(diag(1e0*h*ones(2*N,1)));
```

Resolution of the equation in Ψ

```
index = 1;
Psi1 = zeros(N,P);Psi2 = zeros(N,P);
u_prev = u_0;
c_prev = c_0;
Psi1(:,1) = u_0;Psi2(:,1) = c_0;
CrossProduct = zeros(N,1);
while (index*delta_t<=tf)
    u_t = calc_Psi_1(Coefs, CrossProduct,u_prev);
    c_t = calc_Psi_2(Coefs, CrossProduct,c_prev,u_t);
    Psi1(:,index+1)=u_t;
    Psi2(:,index+1)=c_t;
    u_prev = u_t;
    c_prev = c_t;
    index = index + 1;
end
```

Resolution of the equation in $\overline{\lambda}$

```
index = P-1;
Lambda1 = zeros(N,P);Lambda2 = zeros(N,P);
L1_next = L1_N; L2_next = L2_N;
Lambda1(:,P) = L1_N;Lambda2(:,P) = L2_N;
Corr2 = zeros(N,1);
while(index*delta_t>=t0)
    if (mod(index+1,10)==0)
        Corr = W_eps*(Measures(:,index+1)-M(Psi1(:,index+1),Psi2(:,index+1)));
        Corr1 = Corr(1:N,1);
        Corr2 = Corr(N+1:2*N,1);
    else
        Corr1 = zeros(N,1);
        Corr2 = zeros(N,1);
    end
    L2_t = calc_lambda_2(Coefs, Psi1(:,index+1), L2_next, Corr2);
    L1_t = calc_lambda_1(Coefs, Psi1(:,index+1), Psi2(:,index+1),L1_next, L2_t, Corr1);
    Lambda1(:,index+1) = L1_t;
    Lambda2(:,index+1) = L2_t;
    L1_next = L1_t; L2_next = L2_t;
    index = index - 1;
end
```

Calibration of parameters

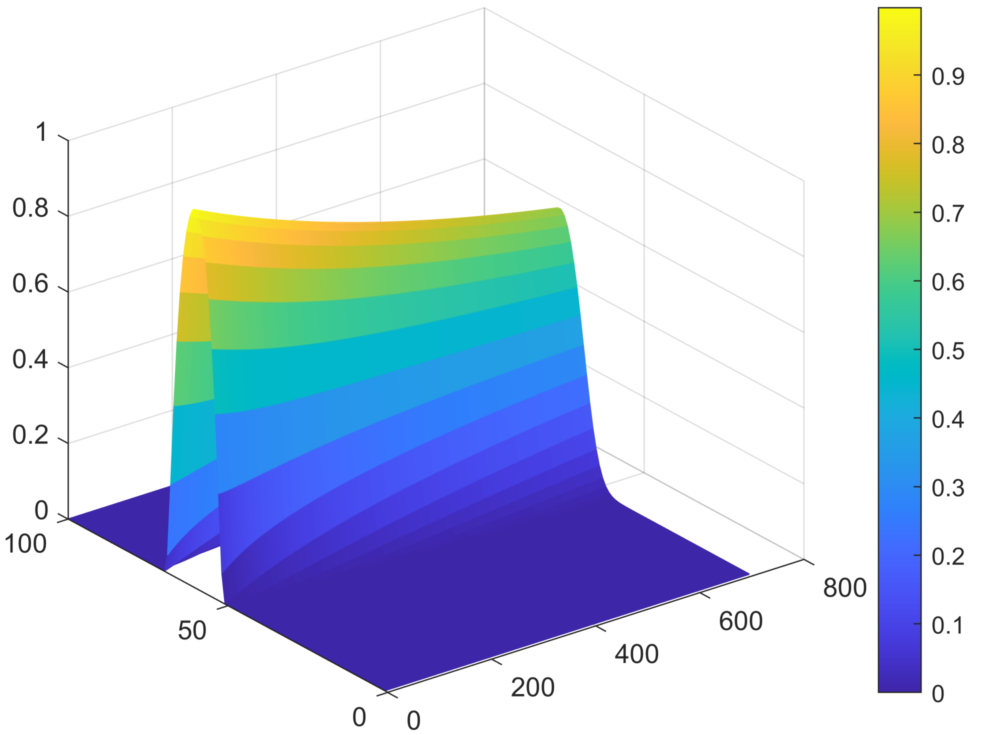
```
%Int_approx = calc_int_param(Lambda1, Lambda2, Psi1, Psi2, Coefs);
%Coefs = Coefs_0 + C_alpha*Int_approx;
Coefs_prev = Coefs;
cond2 = 1e0;counter_here = 0;
while (cond2>=1e-6)  %Coefs = Coefs_0 + C_alpha*Int_approx;
    Int_approx = calc_int_param(Lambda1, Lambda2, Psi1, Psi2, Coefs_prev);
    Coefs_next = Coefs_prev - 0.1*(Coefs_prev-Coefs_0 -C_alpha*Int_approx);
    cond2 = norm(Coefs_next-Coefs_prev)/norm(Coefs_prev);
    Coefs_prev = Coefs_next;
    counter_here = counter_here + 1 ;
end
Coefs = Coefs_next;
```

Plots

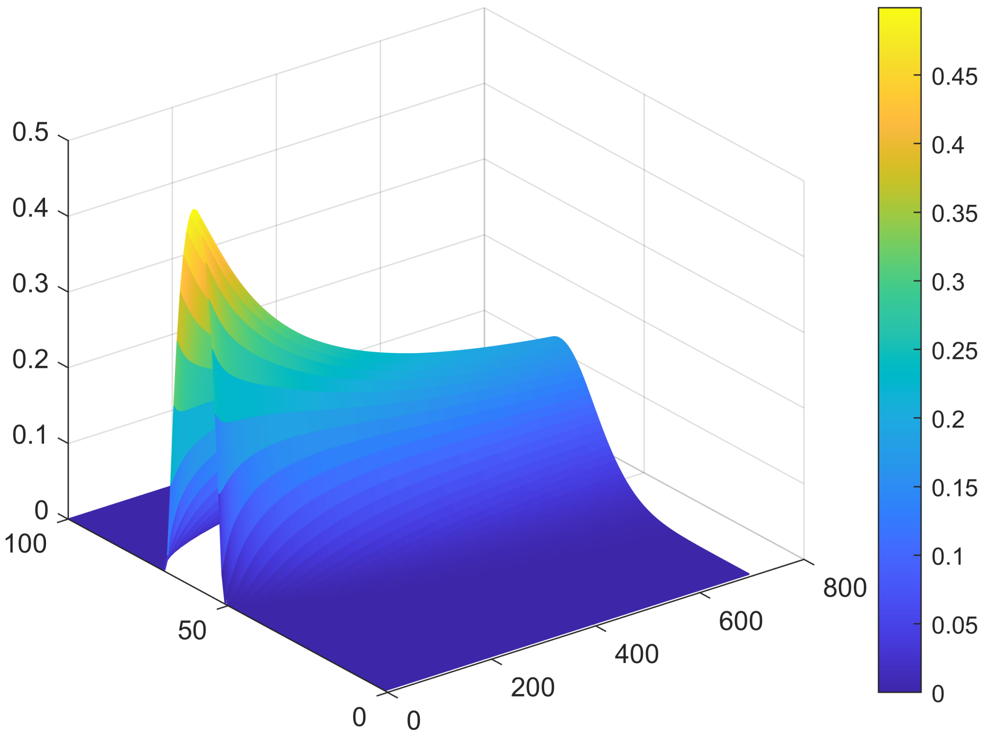
```
% figure;mesh(Psi1);colorbar;
% figure;mesh(Psi2);colorbar;
% figure;mesh(Lambda1);colorbar;
% figure;mesh(Lambda2);colorbar;
```

Iteration in the functions

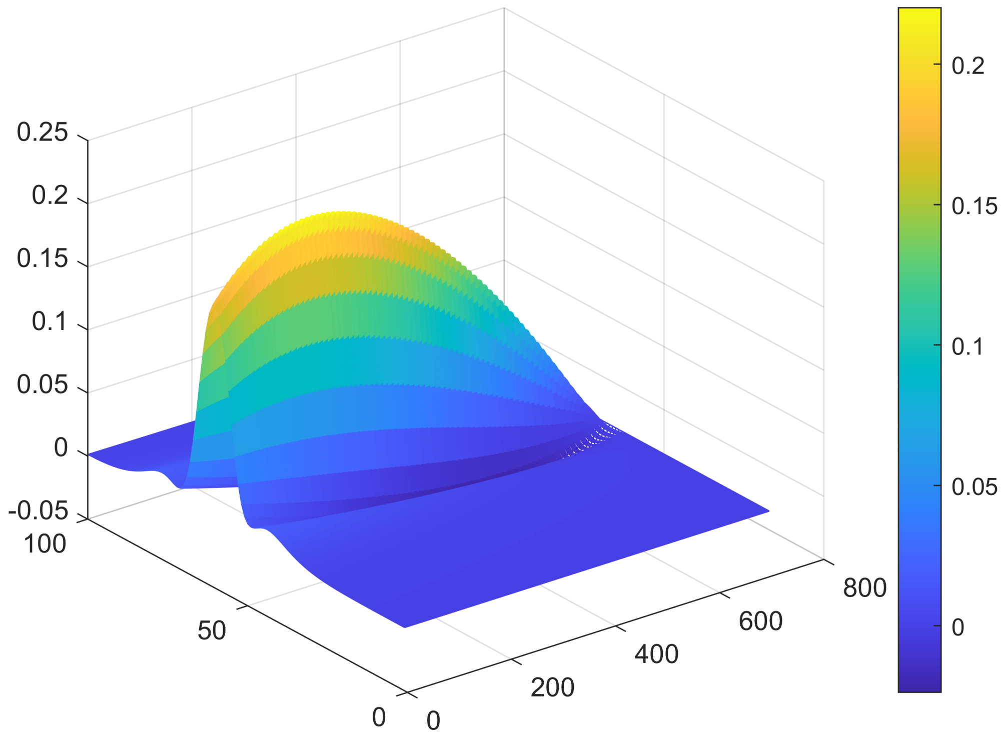
```
iter = 1; itermax = 150;
Coefs_all = zeros(5,itermax+1);
Coefs_all(:,1) = Coefs;
cond = norm(Coefs_all(:,1));
bar = waitbar(0,'Lets get started');
while (iter<=itermax && cond>=1e-3)
    waitbar(iter/itermax,bar,'Progression');
    %Psi
    index = 1;
    Psi1 = zeros(N,P);Psi2 = zeros(N,P);
    Err_cond_ini = C_a*[Lambda1(:,1);Lambda2(:,1)];
    u_prev = u_0 + (Err_cond_ini(1:N,1)); % à compléter
    c_prev = c_0 + (Err_cond_ini(N+1:2*N,1)); % à compléter
    Psi1(:,1) = u_prev;Psi2(:,1) = c_prev;
    CrossProduct = Int_model_error(Lambda1,Lambda2);
    while (index*delta_t<=tf)
        u_t = calc_Psi_1(Coefs, CrossProduct(1:N,index),u_prev);
        c_t = calc_Psi_2(Coefs, CrossProduct(N+1:2*N,index),c_prev,u_t);
        Psi1(:,index+1)=u_t;
        Psi2(:,index+1)=c_t;
        u_prev = u_t;
        c_prev = c_t;
        index = index + 1;
    end
    %Lambda
    index = P-1;
    Lambda1 = zeros(N,P);Lambda2 = zeros(N,P);
    L1_next = L1_N; L2_next = L2_N;
    Lambda1(:,P) = L1_N;Lambda2(:,P) = L2_N;
    Corr2 = zeros(N,1);
    while(index*delta_t>=t0)
        if (mod(index+1,10)==0)
            Corr = W_eps*(Measures(:,index+1)-M(Psi1(:,index+1),Psi2(:,index+1)));
            Corr1 = Corr(1:N,1);
            Corr2 = Corr(N+1:2*N,1);
        else
            Corr1 = zeros(N,1);
            Corr2 = zeros(N,1);
        end
        L2_t = calc_lambda_2(Coefs, Psi1(:,index+1), L2_next, Corr2);
        L1_t = calc_lambda_1(Coefs, Psi1(:,index+1), Psi2(:,index+1),L1_next, L2_t, Corr1);
        Lambda1(:,index+1) = L1_t;
        Lambda2(:,index+1) = L2_t;
        L1_next = L1_t; L2_next = L2_t;
        index = index - 1;
    end

    %Alpha
    Coefs_prev = Coefs;
    cond2 = 1e0;counter_here = 0;
    while (cond2>=1e-6)  %Coefs = Coefs_0 + C_alpha*Int_approx;
        Int_approx = calc_int_param(Lambda1, Lambda2, Psi1, Psi2, Coefs_prev);
        Coefs_next = Coefs_prev - 0.1*(Coefs_prev-Coefs_0 -C_alpha*Int_approx);
        cond2 = norm(Coefs_next-Coefs_prev)/norm(Coefs_prev);
        Coefs_prev = Coefs_next;
        counter_here = counter_here + 1 ;
    end
    Coefs = Coefs_next;
    disp(counter_here);
    %Coefs = Coefs_all(:,iter) -0.01*(Coefs_all(:,iter)-Coefs_0 -C_alpha*Int_approx);

    iter = iter+1;
    Coefs_all(:,iter) = Coefs;
    cond = norm(Coefs_all(:,iter)-Coefs_all(:,iter-1))/norm(Coefs_all(:,iter-1));
end
close(bar);
figure;mesh(Psi1);colorbar;
```



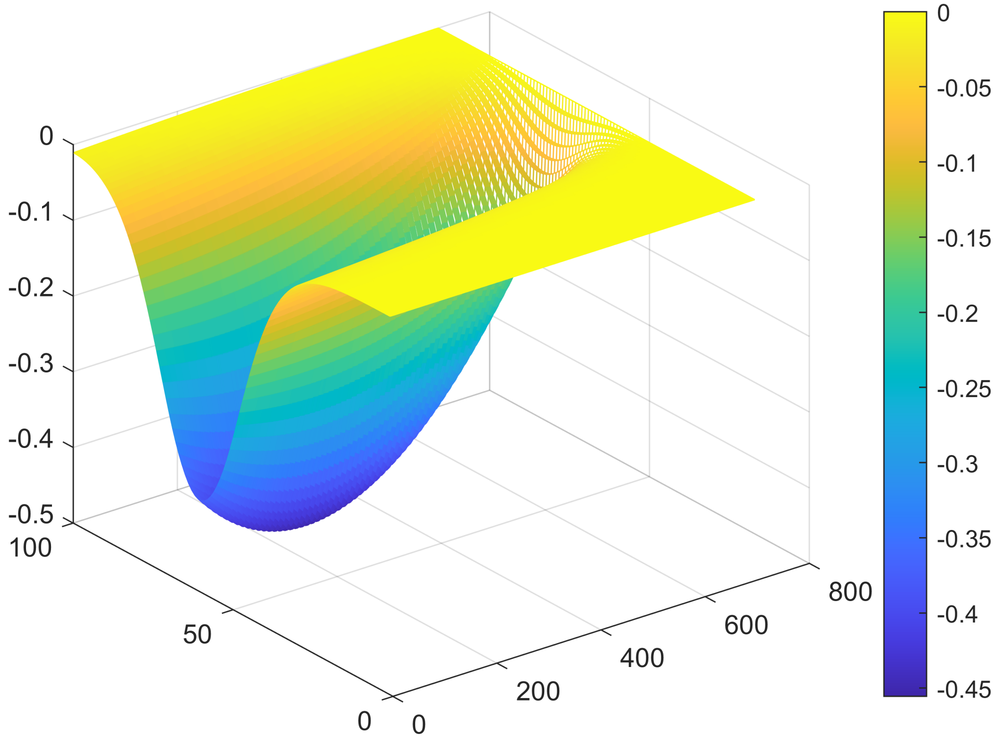
```
figure;mesh(Psi2);colorbar;
```



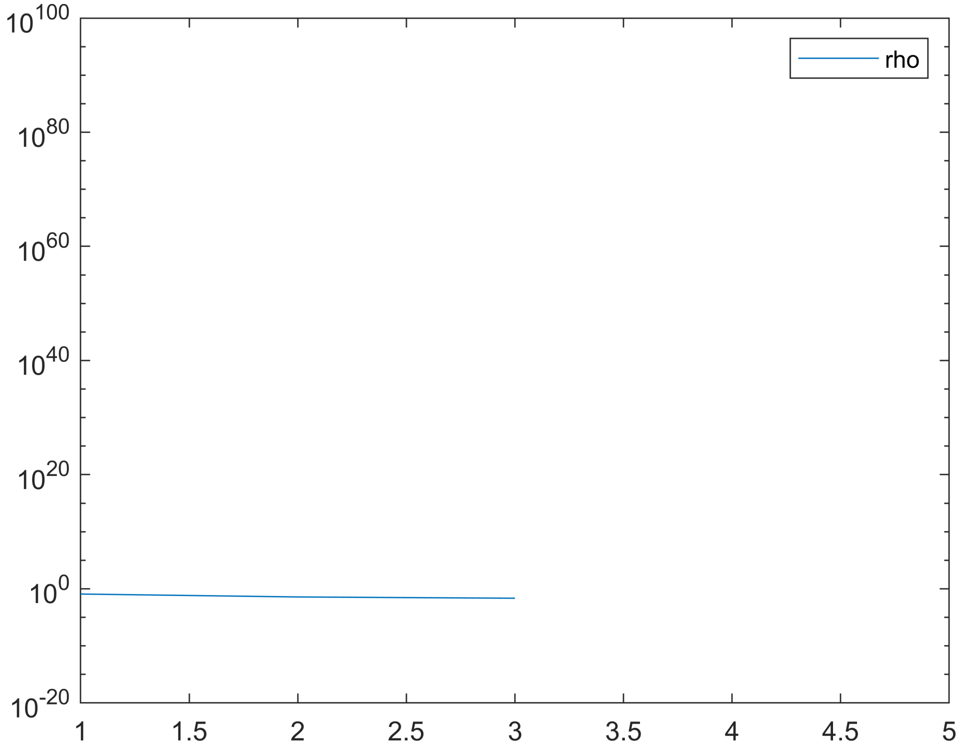
```
figure;mesh(Lambda1);colorbar;
```



```
figure;mesh(Lambda2);colorbar;
```



```
figure;semilogy(exp(Coefs_all(:,1:iter)'));legend('rho','delta','alpha','beta','gamma');
```


10^{100}
10^{80}
10^{60}
10^{40}
10^{20}
10^{0}
10^{-20}
rho