YATIŞKIN –HAL MODEL EŞİTLİKLERİNİN ÇÖZÜMÜ [1-5]

Kaynaklar

1. Luyben, W.L.1990. Process Modeling, Simulation and Control for Chemical Engineers, 2nd ed.,McGraw-Hill, New York.
2. Bequette, B.W. 1998. Process Dynamics, Modeling, Analysis and Simulation, Prentice Hall, New Jersey
3. Thomas E. Marlin, 2000. Designing Procesees and Control Systems for Dynamic Performance, 2nd Edition, McGraw Hill Book Co, Singapore.
4. Matlab 9, The MathWorks, Inc., Apple Hill Drive, Natick, MA.,2009
5. Alpbaz M.,Proses Kontrol, A.Ü.F.F. Döner Sermaye İşletmesi Yayınları, No:121993 Ankara

Yatışkın –Hal Model Eşitliklerinin Çözümü

Bu denklemin çözümündeki kökler değişkenlerin yatışkın hal değerleridir.

N değişken içeren bir sistem için n tane eşitlik göz önünde bulundurulursa,

$$
\begin{aligned}
&f_1(x_1, x_2, \dots x_n) = 0 \\
&f_2(x_1, x_2, \dots x_n) = 0 \\
&\dots\dots\dots \\
&f_n(x_1, x_2, \dots x_n) = 0
\end{aligned}
\qquad (1)
$$

$$f(x) = 0$$

Şeklinde cebirsel denklemler elde edilebilir.
Be denklem setini matris formunda yazmak istersek;

N değişkenli ve n tane doğrusal model eşitliklerinin

genel formunu göstermek istersek;

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & . & . & a_{1n} \\ a_{21} & a_{22} & a_{23} & . & . & a_{2n} \\ . & . & . & . & . & . \\ a_{n1} & a_{n2} & a_{n3} & . & . & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ . \\ . \\ b_n \end{bmatrix} \qquad (2)$$

Bunun matris formu ise ;

$$Ax=b \qquad (3)$$

Bilinmeyen x i çözmek için düzenleme yapılırsa,

$f(x)=Ax-b=0$

Düzenlenirse;

$x= A^{-1} b$

Buradan bilinmeyen x vektörü bulunabilir.

Matlab ortamında çözüm için;

```
>> x = inv(A)*b
```

EXAMPLE 3.1 Linear Absorption Model, Solved Using MATLAB

Consider a 5-stage absorption column (presented in Module 6 in Section V) that has a model of the following form (**x** is a vector of stage liquid-phase compositions and **u** is a vector of column feed compositions):

$$0 = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u}$$

or

$$\mathbf{A}\,\mathbf{x} = -\,\mathbf{B}\,\mathbf{u}$$

the solution for **x** is $\mathbf{x} = -\mathbf{A}^{-1}\mathbf{B}\,\mathbf{u}$

The values of **A**, **B**, and **u** are:

```
a =
   -0.3250    0.1250         0         0         0
    0.2000   -0.3250    0.1250         0         0
         0    0.2000   -0.3250    0.1250         0
         0         0    0.2000   -0.3250    0.1250
         0         0         0    0.2000   -0.3250
b =
    0.2000         0
         0         0
         0         0
         0         0
         0    0.2500
u =
         0
    0.1000
```

The following MATLAB command can be used to solve for **x**

```
» x = -inv(a)*b*u

x =
    0.0076
    0.0198
    0.0392
    0.0704
    0.1202
```

Use of the MATLAB left-division operator (\) yields the same result more efficiently (faster computation time), using the LU decomposition technique:

```
» x = -a\(b*u)
```

EXAMPLE 3.2 A Reactor with Second-Order Kinetics

The dynamic model for an isothermal, constant volume, chemical reactor with a single second-order reaction is:

$$\frac{dC_A}{dt} = \frac{F}{V} C_{Af} - \frac{F}{V} C_A - kC_A^2$$

Find the steady-state concentration for the following inputs and parameters:

$$F/V = 1 \text{ min}^{-1},\ C_{Af} = 1 \text{ gmol/liter},\ k = 1 \text{ liter/(gmol min)}$$

At steady-state, $dC_A/dt = 0$, and substituting the parameter and input values, we find

$$1 - C_{As} - C_{As}^2 = 0$$

where the subscript s is used to denote the steady-state solution. For notational convenience, let $x = C_{As}$, and write the algebraic equation as

$$f(x) = -x^2 - x + 1 = 0$$

We can directly solve this equation using the quadratic formula to find $x = -1.618$ and 0.618 to be the solutions. Obviously a concentration cannot be negative, so the only physically meaningful solution is $x = 0.618$. Although we know the answer using the quadratic formula, our objective is to illustrate the behavior of the direct substitution method.

To use the direct substitution method, we can rewrite the function in two different ways: (i) $x^2 = -x + 1$ and (ii) $x = -x^2 + 1$. We will analyze (i) and leave (ii) as an exercise for the reader (see student exercise 4).

(i) Here we rewrite $f(x)$ to find the following direct substitution arrangement

MATLAB ROUTINES FOR SOLVING FUNCTIONS OF A SINGLE VARIABLE

MATLAB has two routines that can solve for the zeros of a function of a single variable. FZERO is used for a general nonlinear equation, while ROOTS can be used if the nonlinear equation is a polynomial.

3.4.1 FZERO

The first routine that we use for illustration purposes is `fzero`. `fzero` uses a combination of interval halving and false position.

In order to use `fzero`, you must first write a MATLAB m-file to generate the function that is being evaluated. Consider the function $f(x) = x^2 - 2x - 3 = 0$.

The following MATLAB m-file evaluates this function (the m-file is named `fcn1.m`):

```
function y = fcn1(x)
y = x^2 - 2*x - 3;
```

After generating the m-file `fcn1.m`, the user must provide a guess for the solution to the `fzero` routine. The following command gives an initial guess of $x = 0$.

```
y = fzero('fcn1',0)
```

MATLAB returns the answer:

```
y = -1
```

For an initial guess of $x = 2$, the user enters

```
z = fzero('fcn1',2)
```

and MATLAB returns the answer

```
z = 3
```

These results are consistent with those of Example 3.2, where we found that there were two solutions to a similar problem (we could use the quadratic formula to find them). Again, the solution obtained depends on the initial guess.

A third argument allows the user to select a relative tolerance (the default is the machine precision, `eps`). A fourth argument triggers a printing of the iterations.

3.4.2 ROOTS

Since the equation that we were solving was a polynomial equation, we could also use the MATLAB routine `roots` to find the zeros of the polynomial. Consider the polynomial function:

$$x^2 - 2x - 3 = 0$$

The user must create a vector of the coefficients of the polynomial, in descending order.

```
c = [1 -2 -3]'
```

Then the user can type the following command

```
roots(c)
```

and MATLAB returns

```
ans =
     3
    -1
```

Again, these are the two solutions that we expect.

MULTIVARIABLE SYSTEMS

In the previous sections we discussed the solution of a single algebraic equation with a single unknown variable. We covered direct substitution, bisection, reguli falsi, and Newton's method. In this section, we will discuss the reduction of a multivariable problem to a single-variable problem, as well as the multivariable Newton's method.

Consider a system of n nonlinear equations in n unknowns

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

There are some special cases where $n - 1$ variables can be solved in terms of one variable—then a single variable solution technique can be used. This approach is shown in the following example.

XAMPLE 3.3 Reducing a two-variable problem to a single-variable problem

Solve the following system of nonlinear equations.

$$f_1(x_1,x_2) = x_1 - 4x_1^2 - x_1x_2 = 0 \tag{3.15}$$

$$f_2(x_1,x_2) = 2x_2 - x_2^2 + 3x_1x_2 = 0 \tag{3.16}$$

From (3.15) we can solve for x_2 in terms of x_1 to find:

$$x_2 = 1 - 4x_1 \tag{3.17}$$

EXAMPLE 3.3 Reconsidered. Using MATLAB

The m-file used to implement Example 3.3 using `fsolve` is:

```
function f = nle(x)
f(1)= x(1)-4*x(1)*x(1)-x(1)*x(2);
f(2)= 2*x(2)-x(2)*x(2)+3*x(1)*x(2);
```

which is placed in an m-file called `nle.m`

The initial guess is entered

```
x0 = [1  1]';
```

and we obtain the solution by entering

```
x = fsolve('nle',x0)
```

which gives us the expected results

```
x = [0.2500   0.0000]'
```

Computationally faster results will be obtained if the analytical Jacobian is used.

$$\mathbf{J} = \begin{bmatrix} 1 - 8x_1 - x_2 & -x_1 \\ 3\,x_2 & 2 - 2\,x_2 + 3\,x_1 \end{bmatrix}$$

The following function file generates the analytical Jacobian for this problem.

```
function gf = gradnle(x)
gf(1,1)=1-8*x(1)-x(2);
gf(1,2)=-x(1);
gf(2,1)=3*x(2);
gf(2,2)=2-2*x(2)+3*x(1);
```

which we place in an m-file called `gradnle.m`. We can then solve this problem by entering

```
x0 = [1  1]';
options(5)=0;
x = fsolve('nle',x0,options,'gradnle')
```

The options vector can be used to select the Levenberg-Marquardt method by setting

```
options(5)=1;
```

Sistemde Yatışkın-hal Enerji denkliği yazılırsa:

{ Birim zamanda sisteme giren enerji miktarı } - { Birim zamanda sistemden Çıkan enerji miktarı } = 0

$$mc_pT_0^0 + Q - mc_pT_1^0 = 0 \qquad (2)$$

Burada, m :Akışkanın birim zamandaki kütlesini

c_p :Akışkanın ısı kapasitesini,

T_0^0 :Akışkanın tanka giriş sıcaklığını,

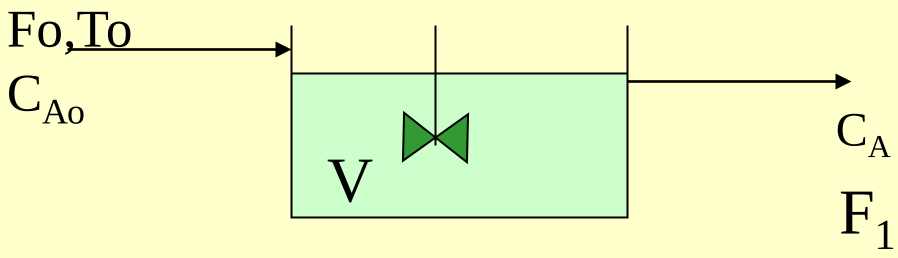
Q : Tanka birim zamanda dışarıdan verilen ısı miktarını,

T_1^0 :Akışkanın tankdan çıkış sıcaklığını göstermektedir.

Örnek: Sürekli Tam Karıştırmalı bir reaktör modeli

Aşağıdaki bir CSTR'de birinci mertebeden bir tersinmez reaksiyon oluşmaktadır.

$$A \xrightarrow{k} B; \quad r_A = -kC_A$$



Şekil 3: Karıştırmalı bir reaktör

Sistemde Yatışkın-hal toplam kütle denkliği:

{Birim zamanda sisteme giren madde miktarı} - {Birim zamanda sistemden Çıkan madde miktarı} = 0

$$F_0\rho_0 - F_1\rho_1 = 0 \qquad (3)$$

Sistemde Yatışkın-hal A bileşeni kütle denkliği:

{Birim zamanda sisteme giren A miktarı} - {Birim zamanda sistemden Çıkan madde miktarı} - {Birim zamanda sistemden kaybolan A miktarı} = 0

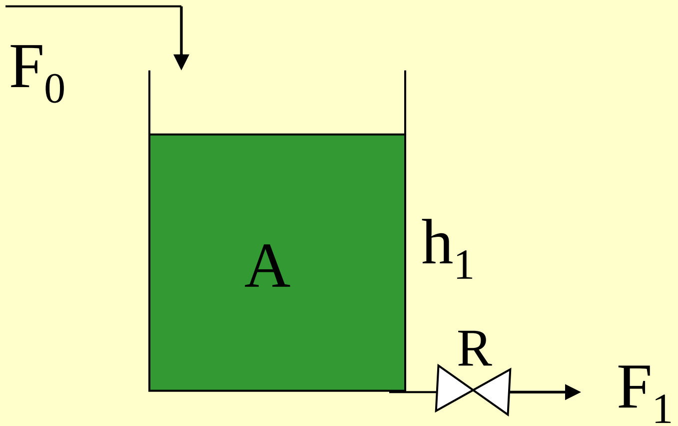
$$FC_{A0}^{0} - FC_{A}^{0} - kC_{A}^{0}V = 0 \qquad (4)$$

Sistemde hacim sabit olduğundan F hacimsel akış hızları eşit alınmıştır

Kimyasal proseslerin yatışkın olmayan hal modelleri

I. Mertebeden Dif.Denklemlerle ifade edilen sistemler

-Sıvı Seviye Sistemi



Şekil 4: Sıvı seviye tankı

Sistemde Yatışkın Olmaya kütle denkliği yazılırsa;

$$\left\{ \begin{array}{l} \text{Birim zamanda sisteme} \\ \text{giren madde miktarı} \end{array} \right\} - \left\{ \begin{array}{l} \text{Birim zamanda sistemden} \\ \text{Çıkan madde miktarı} \end{array} \right\} = \left\{ \begin{array}{l} \text{Birim zamanda} \\ \text{sistemde biriken} \\ \text{madde miktarı} \end{array} \right\}$$

$$F_0\rho_0 - F_1\rho_1 = A\rho\frac{dh_1}{dt} \qquad (5)$$

$F_1 = \frac{h_1}{R}$ (5) denkleminde yerine konup düzenlenirse,

$$AR\frac{dh_1}{dt} + h_1 = RF_0 \qquad (6)$$

$$\tau\frac{dh_1}{dt} + h_1 = RF_0 \qquad (7)$$

τ Zaman sabitidir