Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research

University Hassiba Benbouali of Chlef
Faculty of Technology
Department of Process Engineering

Level: Master 2 – Pharmaceutical Process Engineering

Process Optimization and Modeling Course

By: Dr. Malika OUAGUED

Academic Year: 2024/2025

Foreword

This manuscript presents a course on optimization and process modeling, designed for second-year Master's students in Pharmaceutical Process Engineering. It follows the official program of the Process Optimization and Modeling module and aims to provide students with a structured and practical approach to these fundamental concepts.

The course is divided into two main chapters, each of which contains different examples involving reflection or simple calculations based on bibliographical data.. The first chapter focuses on process modeling and simulation, introducing commonly used models in process engineering and demonstrating their implementation using Excel and MATLAB. The second chapter is dedicated to process optimization, presenting various techniques and methodologies for solving optimization problems, both with and without constraints.

This course material has been developed based on extensive teaching experience and a thorough study of academic and technical resources, some of which are not explicitly cited in the bibliography. It is designed to offer a clear and comprehensive understanding of process modeling and optimization, providing students with both theoretical foundations and practical applications.

We hope that this manuscript will serve as a valuable learning resource, offering a concise yet effective alternative to traditional textbooks. May it assist students in mastering these essential concepts and applying them successfully in their academic and professional careers.

Table of Contents

Chapitre I:

Modeling and Simulation

CHAPTER I. MODELING AND SIMULATION

I.1. Introduction to the modeling of physical phenomena

This chapter provides knowledge for process modeling and simulation. It also revisits fundamental laws of physics and chemistry to establish models in process engineering.

I.1.1. Definitions

1. System

A system refers to a physical structure considered in isolation, such as a crystallizer, a chemical reactor, or a workshop.

Its inputs are the parameters that influence its behavior, which may or may not be controlled. Its outputs are the parameters resulting from the inputs.

Example: Let us consider an adiabatic stirred reactor operating in steady-state, where a chemical reaction **A→B** occurs.

- The system's input variables are feed flow rate, concentration of A, reactor temperature, and reactor volume.
- The output variables are the conversion rate of A to B, or the concentrations of each component and the exit temperature.

2. Modeling

Modeling is the process of creating a simplified and abstract representation of a system, phenomenon, or real-world situation using numbers and mathematical equations.

This representation, called a model, can take different forms and be created using various techniques.

The purpose of modeling is to better understand, analyze, predict, or simulate the behavior of the real system or phenomenon being studied.

The model allows for experiments, simulations, or calculations that would be difficult, costly, or impossible to perform on the actual system.

I.1.2. Modeling in Process Engineering

- Process modeling is a tool used for the design, development, and optimization of industrial processes, such as chemical plant operations, environmental systems, complex manufacturing operations, and other similar technical processes.

- It involves representing a chemical process using a mathematical model based on experimental data collected from the actual system, which is then solved to obtain insights into process performance.

1. Simulation

Numerical models are created using computers and specialized software.

These models simulate the system's or phenomenon's behavior by solving the mathematical equations that describe it.

- Simulation involves conducting experiments on the model to infer system performance, predict behavior, test compliance with standards, etc.

2. Underlying Phenomena

The structure of a process model depends on the phenomena involved in the process. These phenomena can include:

- Mass and heat transfer within a single phase or between phases;
- Mixing phenomena in equipment;
- Kinetic phenomena related to chemical or biological reactions;
-Thermodynamic phenomena governing physicochemical equilibria and the physical properties of materials.

3. Objectives:

In process engineering, modeling serves two major purposes:

- Knowledge acquisition and consolidation, and
- Process control and supervision.

I.1.3. Steps in Modeling

A model must be developed following a structured approach comprising several steps:

Identification of the primary objective of modeling;

- Characterization of the material system under study: thermodynamic behavior, kinetics,…
- Identification of key phenomena and limiting factors through carefully chosen experiments;
- Selection of a theoretical basis from available theories: formulation of model hypotheses;

- Formulation of equations;
- Selection of a numerical solution method;
- Parameter estimation;
- Model validation using real-world data (experimental strategy methodologies should be employed).

I.2. Methods for Solving Systems of Algebraic Equations

I.2.1. Solving RK and RKS Equations of State Using Matlab/Excel

In process engineering, the RK (Redlich-Kwong) and RKS (Redlich-Kwong-Soave) equations of state are thermodynamic models used to predict the behavior of real gases, particularly in non-ideal conditions. These models extend upon the ideal gas law by incorporating intermolecular forces and the finite volume of molecules.

1. Redlich-Kwong (RK) Equation of State

The RK equation of state is one of the simplest modifications of the van der Waals equation, developed in 1949. It is expressed as:

$$P = \frac{RT}{\vartheta - b} - \frac{a}{\vartheta(\vartheta + b)}$$

With:

$$a = 0.42748\left(\frac{R^2Tc^2}{Pc}\right).\alpha \quad , \quad b = 0.08664\left(\frac{RTc}{Pc}\right) \quad , \quad Tr = \frac{T}{Tc}, \quad \alpha = \frac{1}{Tr^{0.5}}$$

Where:

P: Pressure;

T: Temperature;

v: Molar volume;

R: Universal gas constant, R=0.082 L·atm/mol·K.

a: Parameter related to intermolecular forces

b: Parameter related to the finite size of molecules

Tr: Reduced temperature

Tc, Pc: Critical temperature and pressure.

2. Redlich-Kwong-Soave (RKS) Equation of State

The RKS equation, developed by Soave in 1972, is a modification of the RK equation, introducing a temperature-dependent alpha (α) function to improve accuracy for predicting real gas behavior, particularly for non-polar and slightly polar substances. It is expressed as:

$$P = \frac{RT}{\vartheta - b} - \frac{a}{\vartheta(\vartheta + b)}$$

With:

$a = 0.42748 \left(\frac{R^2 Tc^2}{Pc}\right).\alpha$, $b = 0.08664 \left(\frac{RTc}{Pc}\right)$, $Tr = \frac{T}{Tc}$

$\alpha = [1 + m(1 - Tr^{0.5})]^2$, $m = 0.480 + 1.57\omega + 0.176\omega^2$

Where:

m: Empirical parameter related to the acentric factor (ω)

3. Cubic Form of RK and RKS Equations of State

The Redlich-Kwong (RK) and Redlich-Kwong-Soave (RKS) equations of state can be written as cubic equations in terms of molar volume (v) for computational purposes. Below are their forms:

$$\boldsymbol{\vartheta^3(P) - \vartheta^2(RT) + \vartheta(a - Pb^2 - RTb) - ab = 0}$$

Notes:

- These cubic equations allow for up to three real roots,

- Physical Significance of Roots, for vapor-liquid equilibrium (VLE):

  - The largest root typically represents the gas phase,
  - The smallest root represents the liquid phase.
  - Intermediate roots are generally not physically meaningful.

- These cubic forms are critical for solving real-gas behavior using numerical techniques in software such as MATLAB or Excel.

- These equations can be expressed in terms of the compressibility factor (Z), where Z=Pv/RT , to simplify the computation.

4. RK and RKS Equations of State for Mixtures

When dealing with mixtures, the Redlich-Kwong (RK) and Redlich-Kwong-Soave (RKS) equations of state must account for interactions between different components. This is achieved by defining mixture-specific parameters a and b using mixing rules:

$$a_{mel} = \left(\sum_{i=1}^{Ncomp} y_i a_i^{0.5}\right)^2 \quad \text{and} \quad b_{mel} = \left(\sum_{i=1}^{Ncomp} y_i b_i\right)$$

Where: y_i, is the mole fraction of the vapor-phase components.

5. Applications of RK and RKS in Process Engineering

RK: Used in basic calculations, moderate conditions, and as a starting point for more complex models.

RKS: Widely used in process simulation software for phase equilibria, thermodynamic property estimations, and equipment design, including distillation and refrigeration.

By providing a balance of simplicity and accuracy, these equations play a critical role in designing and optimizing industrial processes.

6. Algorithm for Calculating Molar Volume Using RK and RKS Equations of State

The molar volume (v) is calculated by solving the cubic form of the Redlich-Kwong (RK) or Redlich-Kwong-Soave (RKS) equations of state.

Steps for Solving v:

1. *Input Required Data:*
   - Pressure (P). Temperature (T)
   - Critical Properties (Tc , Pc): Critical temperature and pressure of the substance.
   - Acentric Factor (ω): For RKS only.
   - Gas Constant (R): Universal gas constant.

2. *Calculate RK/RKS Parameters*
   - For RK : Tr, α, a, b
   - For RKS: Tr, m, α, a, b

3. *Solve the Cubic Equation for molar volume.*

Tools and Software for Implementation

- *MATLAB*: Built-in solvers like roots() in MATLAB;
- *Excel:* Numerical libraries in Excel (e.g., Solver).
- *Python:* Use libraries like numpy or scipy.optimize.
- *Process Simulation Software*: Aspen Plus, HYSYS, or equivalent for built-in RK/RKS solvers.

Example 1:

Calculate the molar volume of n-butane at 500 K and 18 atm using the RK equation of state. (Pc=37.96atm ; Tc=425.03K)

Solution

1. Using Excel Table:

Molar volume of nbutane (RK)

Data	
P(atm)	18
T(K)	500
Tc(K)	425,03
Pc(atm)	37,96
R(l.atm/mol.K)	0,082
omega	
Calcul of RK parameters	
Tr	1,17638755
alpha	0,92198698
a	12,6119216
b	0,07954734
Variable (v)	2,03958396
Eq (RK)	8,3308E-07

2. Using Matlab

```
%Calcul of molar volum of n-butane (RK)
 %input data
P=18 %atm
T=500 %K
Tc=425.03  %k
Pc=37.96  %atm
R=0.082  %l.atm/k.mol
% Calcul of RK parameters
Tr=T/Tc
alpha=1/Tr^0.5
a=0.42748*R^2*Tc^2*alpha/Pc
b=0.08664*R*Tc/Pc
% RK equation
eq=[P -R*T (a-P*b^2-R*T*b) -a*b]
v=roots(eq)
v =
   2.0396 + 0.0000i
```

Example 2:

Consider the following gas mixture: CO (100 kmol/hr), H_2 (200 kmol/hr), and CH_3OH (100 kmol/hr). The mixture is at 100 atm and 300°C. Calculate the specific volume using RKS:

	CO	H2	CH3OH
Tc (K)	**132.92**	**33.19**	**512.5**
Pc (atm)	**34.532**	**12.958**	**79.783**
ω	**0.049**	**- 0.22**	**0.559**

1. Solution Using Matlab

```
% molar volume of mixture
%data
T=300+273;P=100;R=0.082;
TC=[132.92 33.19 512.5]
PC=[34.532 12.958 79.783]
W=[0.049 -0.22 0.559]
Y=[0.25 0.5 0.25]

%CALCUL OF PARAMETERS
amel=0
bmel=0

for i=1:3
    TR(i)=T/TC(i)
    M(i)=0.480+1.574*W(i)+0.176*W(i)^2
    alpha(i)=(1+M(i)*(1-TR(i)^0.5))^2
    A(i)=0.42748*R^2*TC(i)^2*alpha(i)/PC(i)
    b(i)=0.08664*R*TC(i)/PC(i)
    am(i)=A(i)^0.5*Y(i)
    bm(i)=b(i)*Y(i)
    amel=amel+am(i)
    bmel=bmel+bm(i)
end

amel=amel^2

%RKS EQUATION
eq=[P -R*T (amel-P*bmel^2-R*T*bmel) -amel*bmel]

v=roots(eq)

v =   0.4799 + 0.0000i
```

2. *Solution Using Excel Table*

Specific volume of a mixture using RKS Method

T(K)	**573**
P(atm)	**100**
R	**0,082**

Gaz	Pc	Tc	w	Tr	m	alpha	a	b	y	amel	bmel
CO	34,532	132,92	0,049	4,31086368	0,55754858	0,15994535	0,2352199	0,02734645	0,25	0,12124869	0,00683661
H2	12,958	33,19	-0,22	17,2642362	0,1422384	0,30385918	0,07424931	0,01819707	0,5	0,13624363	0,00909854
CH3OH	79,783	512,5	0,559	1,11804878	1,41486266	0,84422587	7,98876298	0,04563686	0,25	0,70661	0,01140922
										0,96410231	0,02734437
										0,92949326	**0,02734437**

Obj. Function	**-2,3062E-06**
Variable	**0,4799248**

I.2.2. L - V equilibrium calculation and flash separation using Excel and Matlab

1. Flash Separation

Flash separation (or flash vaporization) is a single-stage separation process where a liquid mixture is partially vaporized to produce a vapor phase and a liquid phase in equilibrium. It is commonly used in the oil, gas, and chemical industries.

Applications:

- Initial separation of crude oil in refineries.
- Dehydration or recovery of valuable components in natural gas processing.
- Simplifying separation steps in chemical and petrochemical processes.

Process Description:

- The feed (a liquid or a liquid-vapor mixture) is introduced into a flash drum or separator at specific pressure and temperature conditions.
- Due to the sudden change in conditions, a portion of the liquid vaporizes, resulting in the formation of a vapor phase and a residual liquid phase.
- These two phases are separated based on the equilibrium compositions predicted by L-V calculations.

2. L-V Equilibrium Calculation for isothermal Flash

L-V equilibrium (liquid-vapor equilibrium) calculations involve determining the distribution of components between liquid and vapor phases at a given temperature and pressure.

Assumptions

- The system operates under isothermal and isobaric conditions.
- Vapor and liquid phases are in thermodynamic equilibrium.

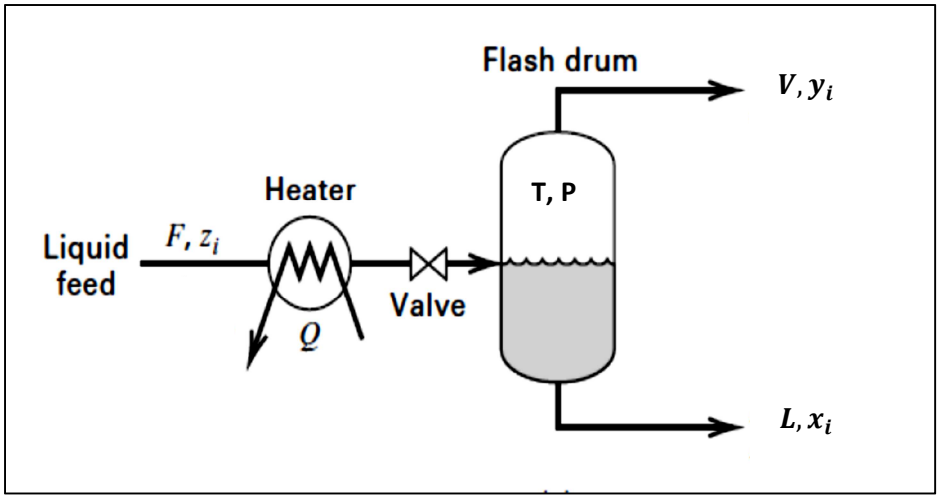


Figure I.1. Flash separator.

Equilibrium Ratio (Ki): This is the ratio of the mole fraction of a component i in the vapor phase (yi) to that in the liquid phase (xi):

$$y_i = K_i x_i$$

Step 1: Material Balance

For a feed F, the liquid flow rate is L, and the vapor flow rate is V. The overall material balance:

$$F = L + V$$

Introducing the vapor fraction (ϕ), where φ =V/F, we get:

$$L = F(1-\varphi)\ ,\ \ V = F\varphi$$

Step 2: Component Balance

The component mole balance for any component i is:

$$z_i F = x_i L + y_i V$$

Substituting L and V in terms of $\boldsymbol{\varphi}$, we get:

$$z_i = x_i(1-\varphi) + y_i\varphi$$

Step 3: Equilibrium relation

Using the equilibrium relation: $y_i = K_i x_i$, the above equation becomes:

$$z_i = x_i(1-\varphi) + Ki\ xi\varphi$$

Factor out x_i: $x_i = \frac{z_i}{(1-\varphi)+Ki\varphi}$ *(1)*

Step 4: Summation Constraint

The mole fractions of the liquid phase (x_i) and vapor phase (y_i) must sum to 1:

$$\sum_{i=1}^{Ncomp} y_i = 1 \; et \; \sum_{i=1}^{Ncomp} x_i = 1 \Rightarrow \sum_{i=1}^{Ncomp} y_i - \sum_{i=1}^{Ncomp} x_i = 0 \Rightarrow \sum_{i=1}^{Ncomp} K_i x_i - \sum_{i=1}^{Ncomp} x_i = 0$$

$$\Rightarrow \sum_{i=1}^{Ncomp} (K_i - 1)\, x_i = 0$$

Substituting eq (1) in eq (2):

$$\sum_{i=1}^{Ncomp} \frac{(K_i-1)z_i}{1+(K_i-1)\varphi} = 0 \quad \textbf{\textit{Rachford-Rice Equation}}$$

Step 5: Solving Rachford-Rice Equation

- The equation is nonlinear and typically solved iteratively using methods like Newton-Raphson
- Given z_i and K_i, the solutionφ, from which xi and yi can be calculated.

Note: We can use Raoult's and Dalton's Laws to calculate Ki (if not given):

- *Dalton's Law*: For ideal gaz mixtures, the partial pressure of a component in the vapor phase is proportional to its mole fraction in the vapor (y_i) and mixer vapor pressure:

$$P_i = y_i P$$

- Raoult's Law: for ideal mixtures, the partial pressure of a component in the vapor phase is proportional to its mole fraction in the liquid and its pure component vapor pressure:

$$P_i = x_i P_i^{sat}$$

- From the 2 laws, K_i can be calculated by: $K_i = \frac{P_i^{sat}}{P}$

Example:

The composition of the inlet flow F for a hydrocarbon mixture is provided in the following table:

Compounds	z_i	K_i
n-butane	0.25	2.13
n-pentane	0.45	1.10
n-hexane	0.30	0.59

Using Excel and Matlab, calculate the vapor flow rate V, the liquid flow rate L, and the molar fractions x_i and y_i if the inlet flow rate F=10.

1. Solution using Excel

L-V equilibrium composition in a flash separator

	Zi	Ki	N=(ki-1).zi	D=1+(ki-1)φ	N/D	xi	yi
n-butane	0,25	2,13	0,2825	1,94275078	0,14541237	0,12868352	0,27409589
n-pentane	0,45	1,1	0,045	1,08342927	0,04153478	0,41534783	0,45688262
n-hexane	0,3	0,59	-0,123	0,65793998	-0,18694714	0,45596864	0,2690215
				Rachford Eq	1,3124E-08		

F	10
varaible	0,83429272
v	8,34292724
L	1,65707276

2. *Solution using Matlab*

```
% L-V Flash separator
%Feed Data
F=10; %kmol/h
K1=2.13;K2=1.1;K3=0.59;
k1=K1-1;k2=K2-1;k3=K3-1;
z1=0.25;z2=0.45;z3=0.3;
%Equation
v=fzero(@(v) (K1-1)*z1/(1+(K1-1)*v)+(K2-
1)*z2/(1+(K2-1)*v)+(K3-1)*z3/(1+(K3-1)*v) ,0.5)
%Flow
V=F*v %Kmol/h
L=F-V  %kmol/h
% Liquid compostion
x1=z1/(1+k1*v);x2=z2/(1+k2*v);x3=z3/(1+k3*v)
X=x1+x2+x3
%Vap composition
y1=x1*K1;y2=x2*K2;y3=x3*K3
Y=y1+y2+y3
v =  0.8343
V =  8.3429
L = 1.6571
x1 =  0.1287  x2 =  0.4153  x3 =  0.4560  X =  1
y1 = 0.2741 y2 =  0.4569 y3 = 0.2690  Y = 1
```

I.2.3. Calculation of equilibrium compositions of a reaction using Excel and Matlab

Consider the reaction:

$$aA + bB \rightleftharpoons cC + dD$$

- The equilibrium constant expression is:

$$K = \frac{[C]^c.[D]^d}{[A]^a.[B]^b}$$

Where $[X]$ are equilibrium concentrations.

Example:

Let's assume:

- Initial concentrations of A and B = 1.0 M
- Initial concentrations of C and D = 0 M
- Equilibrium constant, K=10

Set Up the Excel Table:

Species	Initial	Change	Equilibrium
A	*1.0 M*	*-x*	*1.0 - x*
B	*1.0 M*	*-x*	*1.0 - x*
C	*0.0 M*	*+x*	*x*
D	*0.0 M*	*+x*	*x*

The equilibrium equation is:

$$K = \frac{[C]^c.[D]^d}{[A]^a.[B]^b} = \frac{[C][D]}{[A][B]} = \frac{x^2}{(1.0-x)^2}$$

The composition of the reaction at equilibrium is determined by solving the characteristic equation of chemical equilibrium:

$$\frac{x^2}{(1.0-x)^2} = 10$$

Note:

In more complicated situations, the equation is nonlinear and difficult to solve, so we need to solve it numerically to find the value of x. To solve the above equation numerically, we use Excel or MATLAB.

a. Use Excel's Solver

- Define K equation:

$$K = \frac{x^2}{(1.0-x)^2}$$

- Set the equation equal to the given K value (K=10).
- Use Solver to adjust ***x*** until the equation is satisfied.

- Use *x* to determine equilibrium concentrations.

b. Use Matlab

To solve the equilibrium equation above using MATLAB, you need to create a function to compute it:

$$K - \frac{x^2}{(1.0-x)^2} = 10 - \frac{x^2}{(1.0-x)^2} = 0$$

Then use either ***solve, fsolve*** *or* ***fzero***, to find the value of x and the equilibrium concentrations.

Solution:

a. Solution suing Excel

Calculation of chemical equilibrium compositions

A + B ↔ C + D

Initial cocentration

Initial concentration (A)	1
Initial concentration (B)	1
Initial concentration ©	0
Initial concentration (D)	0

Equilibrium composition

Eq concentration (A)	0,240253071
Eq concentration (B)	0,240253071
Eq concentration ©	0,759746929
Eq concentration (D)	0,759746929
Variable (x)	**0,759746929**
Obj func (Eq)	10,00000028

b.Solution by Matlab

```
%Calcul of chemcial equilibrium composition
%parameters
%initial Conditions
CAin=1;
CBin=1;
CCin=0;
CDin=0;
% Equilirium constant
K=10;
%ALgebraic Equation
x=fzero(@(x) K-(CCin+x)*(CDin+x)/((CAin-
x)*(CBin-x)),0.5)
% Equilibrium composition
CAeq=CAin-x
CBeq=CBin-x
CCeq=CCin+x
CDeq=CDin+x

x = 0.7597
CAeq = 0.2403
CBeq = 0.2403
CCeq = 0.7597
CDeq = 0.7597
```

I.3. Methods for solving systems of differential equations

I.3.1. Calculation of semi-continuous stirred-tank

- The reaction occurring in the reactor is described: $A \rightarrow B$
- The reaction rate is: $r=k.C_A^2$ (mol/l.s)
- Initially, the reactor contains a known volume V_0 of an inert liquid.

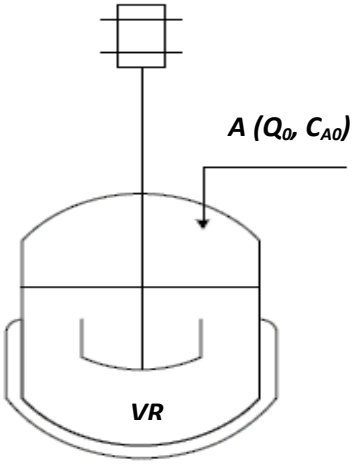


Figure I.2. semi-continuous stirred-tank

1. Material Balance:

❖ Material Balance for A in non-steady state is written as:

Accumulation = Input – Output + Reaction

With: ***Reaction = Generation - Consumption***

$$\frac{dn_A}{dt} = Q_0.C_{A0} + \vartheta_A.r.V_R(t) = Q_0.C_{A0} - \frac{k.n_A^2}{.V_R(t)}$$

❖ The non-steady-state mass balance for all the reactor:

$$\frac{dm_t}{dt} = F_e \Rightarrow \frac{d}{dt}(\rho.V_R) = \rho.Q_0$$

With the assumption that the density does not change (ρ=cst), we obtain:

$$\frac{dV_R}{dt} = Q_0 \quad \text{with} \quad V_R(0) = 0$$

By integration, we find:

$$V_R = V_0 + Q_0 t$$

The equation (1) above then becomes:

$$\boldsymbol{\frac{dn_A}{dt} = Q_0.C_{A0} - \frac{k.n_A^2}{V_0+Q_0t}} \quad \text{with} \quad \boldsymbol{n_A(0) = 0}$$

We obtain a differential equation of Cauchy problem type:

$$\frac{dn_A}{dt} = f(t, n_A) \quad \text{with} \quad n_A(0) = 0$$

Note:

- A Cauchy problem is another term for an initial value problem in the context of differential equations. Both refer to finding a solution to a differential equation that satisfies specific initial conditions:

$$\frac{dx}{dt} = f(t, x) \quad \text{with} \quad x(0) = x_0$$

Cauchy problem or initial value problem

2. Solving differential Equation

2.1. Using the explicit Euler method

The explicit Euler method approximates the solution iteratively using:

$$\boldsymbol{x_{i+1} = x_i + \Delta t.\, f(t_i, x_i)}$$

Algorithm:

1. t=0, $n_A(0)=0$
2. We calculate $n_A(t+\Delta t)$ with:

$$n_A(t+\Delta t) = n_A(t) + \Delta t.\left[Q_0.C_{A0} - \frac{k.n_A(t)^2}{V_0 + Q_0 t}\right]$$

3. We take t=t+Δt
4. We continue until convergence

2.2. Using Runch Kutta methods (RK)

The Runge-Kutta methods order 4 (RK4) is more accurate numerical method for solving Cauchy problems. Here's how to implement RK4:

. Evaluate RK4 coefficients

$$\boldsymbol{k_1 = \Delta t.\, f(t_i, x_i)}$$

$$\boldsymbol{k_2 = \Delta t.\, f\left(t_i + \frac{\Delta t}{2}, x_i + \frac{k_1}{2}\right)}$$

$$\boldsymbol{k_3 = \Delta t.\, f\left(t_i + \frac{\Delta t}{2}, x_i + \frac{k_2}{2}\right)}$$

$$\boldsymbol{k_4 = \Delta t.\, f(t_i + \Delta t, x_i + k_3)}$$

. Calculate $\mathbf{x_{i+1}}$ with:

$$\boldsymbol{x_{i+1} = x_i + \frac{1}{6}.(k_1 + 2k_2 + 2k_3 + k_4)}$$

Algorithm:

2. Start with initial value i=0, $t_0=0$, $n_{A0}=0$

3. Evaluate RK4 coefficients

$$k_1 = \Delta t.\left[Q_0.C_{A0} - \frac{k.n_A(i)^2}{V_0 + Q_0 t_i}\right]$$

$$k_2 = \Delta t.\left[Q_0.C_{A0} - \frac{k.\left(n_A(i) + \frac{k_1}{2}\right)^2}{V_0 + Q_0\left(t_i + \frac{\Delta t}{2}\right)}\right]$$

$$k_3 = \Delta t.\left[Q_0.C_{A0} - \frac{k.\left(n_A(i) + \frac{k_2}{2}\right)^2}{V_0 + Q_0\left(t_i + \frac{\Delta t}{2}\right)}\right]$$

$$k_4 = \Delta t.\left[Q_0.C_{A0} - \frac{k.(n_A(i) + k_3)^2}{V_0 + Q_0(t_i + \Delta t)}\right]$$

4. Calculate $\mathbf{x_{i+1}}$ with:

$$n_A(t + \Delta t) = n_A(t) + \frac{1}{6}.(k_1 + 2k_2 + 2k_3 + k_4)$$

5. Take t=t+Δt
6. Continue until convergence

Example:

$$C_{A0} = \frac{1mol}{s} \quad , \quad k = 0.1\frac{L}{mol.s} \quad , \quad Q_0 = \frac{10l}{s}, \quad V_0 = 50l$$

$$\frac{dn_A}{dt} = 10 - \frac{0.1.n_A^2}{50 + 10t}$$

- Solve the problem using Euler explicit and RK4 for a time duration of 100 seconds and 5 second step size.

a. Euler code in MATLAB

```
% Inputs
f = @(t, y) 10-(0.1*y^2/(50+10*t));  % Differential eq
t0 = 0;              % Initial time
y0 = 0;              % Initial value
h = 5;              % Step size (deltat)
T = 100;             % Final time

% Initialization
N = (T - t0) / h;    % Number of steps
t = t0;              % Start time
y = y0;              % Start value

% Explicit Euler Iteration
for n = 1:N
    y = y + h * f(t, y);  % Update y
    t = t + h;            % Update t
end
 result(n+1, :) = [t, y]

% Plot the solution
plot(result(:, 1), result(:, 2), '-o');
xlabel('Time (t)');
ylabel('Solution (y)');
grid on;
```

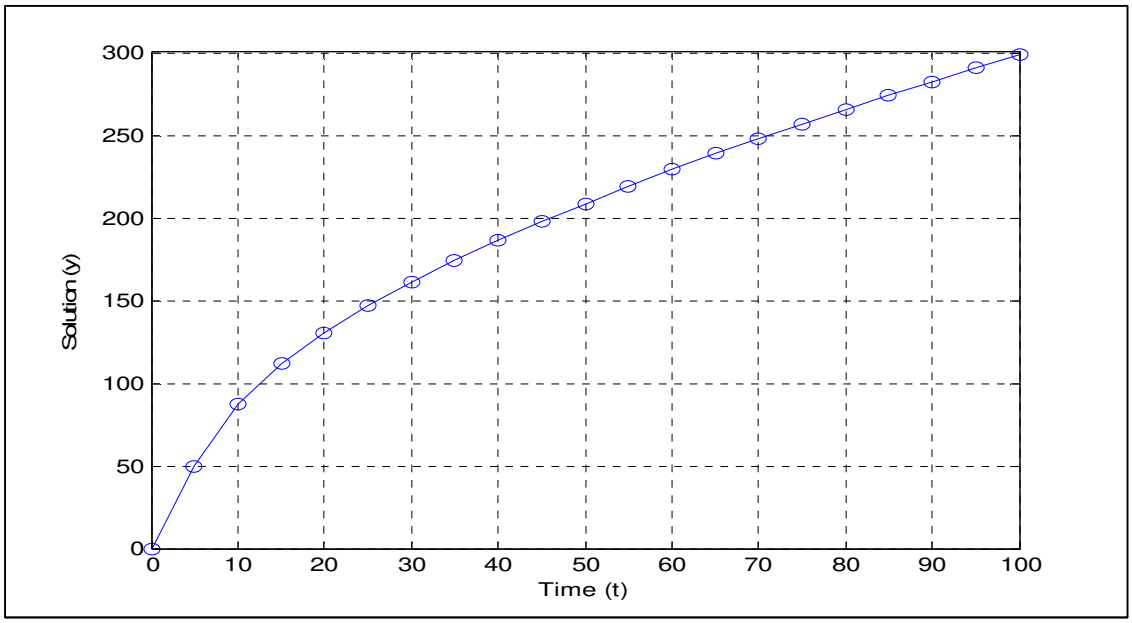


Figure I.3. Simulation results using Euler method.

b. RK4 code in MATLAB

```
% Inputs
f = @(t, y) 10-(0.1*y^2/(50+10*t));  % Differential equation
t0 = 0;           % Initial time
y0 = 0;           % Initial value
h = 5;          % Step size
T = 100;          % Final time
% Initialization
N = (T - t0) / h;    % Number of steps
t = t0;           % Start time
y = y0;           % Start value
% RK4 Iteration
for n = 1:N
   k1 = f(t, y);
   k2 = f(t + h / 2, y + h / 2 * k1);
   k3 = f(t + h / 2, y + h / 2 * k2);
   k4 = f(t + h, y + h * k3);
   y = y + (h / 6) * (k1 + 2 * k2 + 2 * k3 + k4);  % Update y
   t = t + h;                          % Update t
end
 result(n+1, :) = [t, y]
% Plot the solution
plot(result(:, 1), result(:, 2), '-o');
xlabel('Time (t)');
ylabel('Solution (y)');
grid on;
```

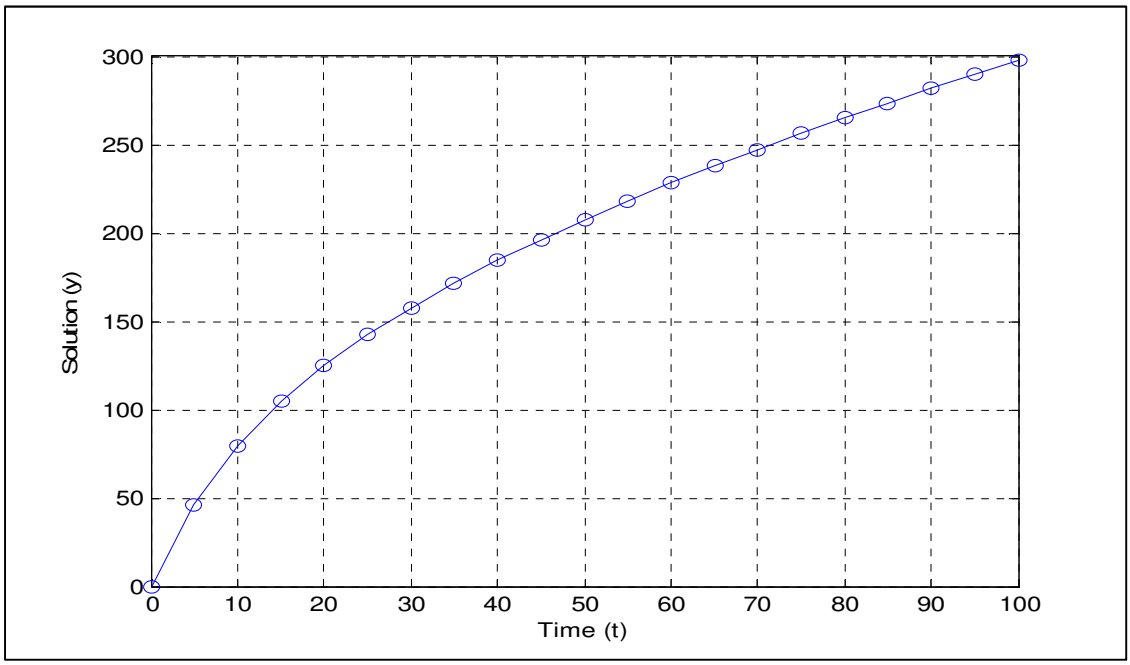


Figure I.4. Simulation results using RK4 method.

II.3.2. Calculation of an isothermal tubular reactor (PFR)

1. Material Balance

Consider an isothermal tubular reactor:

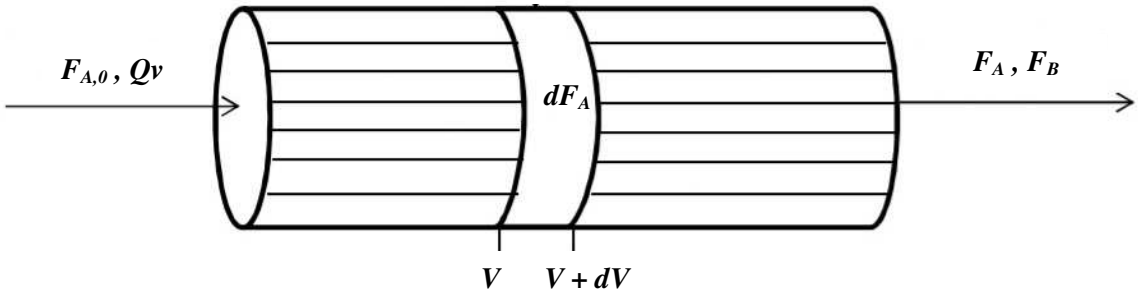


Figure I.5. Tubular reactor (PFR).

Apply mole balance equation: ***Accumulation = Input – Output + Reaction***

The tubular reactor is operated at steady state. The concentration varies continuously down the tube, and, therefore, so does the reaction rate. Therefore, the mole balance is given as the following for species i:

$$F_{A,0} - F_A + \vartheta_A r\, dV = 0 \quad \Rightarrow \quad F_{A,0} + \vartheta_A . r \cdot dV = F_A$$

Replacing $F_A = F_{A0} + dF_A$, we get:

$$F_{A0} + \vartheta_A . r \cdot dV = F_{A,0} + dF_A \quad \Rightarrow \quad \vartheta_A . r \cdot dV = dF_A$$

Replacing $F_A = Qv \cdot C_A$, then dividing by dV:

$$\vartheta_A . r \cdot dV = d(Qv \cdot C_A) = Qv \cdot d C_A$$

The isothermal tubular reactor mole balance equation for a Plug Flow Reactor (PFR):

$$\frac{dC_A}{dV} = \frac{\vartheta_A r}{Qv}$$

Second-order reaction case:

- The reaction occurring in the reactor is described: A → B
- The reaction rate is: $r = k.C_A^2$ (mol/l.s)

The isothermal tubular reactor mole balance equation:

$$\frac{dC_A}{dV} = \frac{-k.C_A{}^2}{Qv}$$

2. Solving differential Equation

2.1. Using the Euler method

Using a finite difference approximation, we discretize the equation as:

$$C_A(V + \Delta V) = C_A(V) + \left(\frac{-k.C_A^2}{Qv}\right).\Delta V$$

Where:

ΔV is a small step size,

The equation is solved iteratively for different values of V

2.2. Using Runch Kutta 4th order (RK4)

A more accurate method is Runge-Kutta (RK4), where we compute:

$$k_1 = f(C_A, V)$$
$$k_2 = f\left(C_A + \frac{\Delta V k_1}{2}, V + \frac{\Delta V}{2}\right)$$
$$k_3 = f\left(C_A + \frac{\Delta V k_2}{2}, V + \frac{\Delta V}{2}\right)$$
$$k_4 = f(C_A + \Delta V k_3\ , V + \Delta V)$$

. Calculate $\mathbf{x_{i+1}}$ with:

$$C_A(V + \Delta V) = C_A(V) + \frac{\Delta V}{6}.(k_1 + 2k_2 + 2k_3 + k_4)$$

Where: $f(C_A, V) = \left(\frac{-k.C_A^2}{Qv}\right)$

2.3. Algorithm for Numerical Resolution

1. Initialize Parameters:
   - Set $C_A(0)=C_{A0}$
   - Set V=0
   - Choose step size ΔV

2. Loop Over Reactor Volume:
   - Compute next concentration using Euler's method or RK4.
   - Update volume V=V+ΔV
   - Store values.
3. Plot the Results.

Example:

$$C_{A0} = \frac{1mol}{s} \quad , \quad k = 0.1\frac{L}{mol.s} \quad , \quad Q_0 = \frac{5l}{s}, \quad V_0 = 50l$$

Solve the problem using Euler explicit and RK4 for a volume step size about 0.5l.

a. Euler code in MATLAB

```
% Parameters
k = 0.1; % Reaction rate constant (L/mol·s)
Qv = 5; % Initial molar flow rate of A (l/s)
CA0 = 1.0; % Initial concentration of A (mol/L)
V = 50; % Maximum reactor volume (L)
deltaV = 0.5; % Step size for numerical integration
```

```
% Initialization
V = 0:deltaV:V; % Volume grid
CA = zeros(size(V)); % Concentration array
CA(1) = CA0;  % Initial concentration
```

```
% Euler's Method Loop
for i = 1:length(V)-1
dCAdV = - (k * CA(i)^2) / Qv; % Differential equation
CA(i+1) = CA(i) + dCAdV * deltaV; % Euler step
end
```

```
% Plot Results
figure;
plot(V, CA, 'bo-', 'LineWidth', 2);
xlabel('Reactor Volume (L)');
ylabel('Concentration of A (mol/L)');
title('Second-Order Reaction in PFR (Euler Method)');
grid on;
```

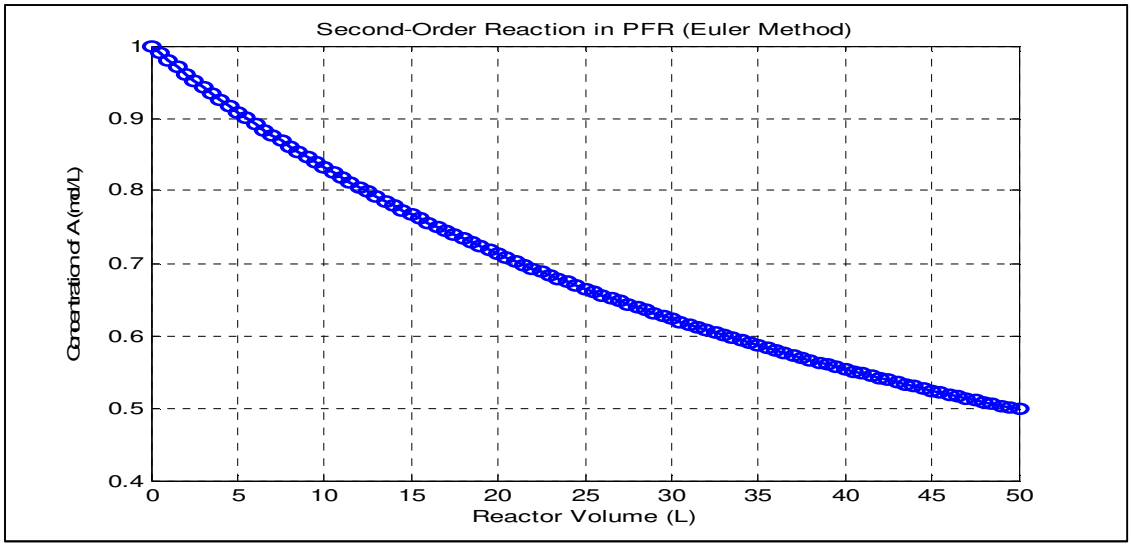


Figure I.6. Simulation results using Euler method.

b. RK4 code in MATLAB

```
% Parameters

 k = 0.1; % Reaction rate constant (L/mol·s)
Qv = 5; % Initial flow rate of A (L/s)
CA0 = 1.0; % Initial concentration of A (mol/L)
 V = 50; % Maximum reactor volume (L)
deltaV = 0.5; % Step size for numerical integration

% Initialization

V = 0:deltaV:V; % Volume grid
CA = zeros(size(V)); % Concentration array
CA(1) = CA0; % Initial concentration

 % Runge-Kutta 4th Order Method

for i = 1:length(V)-1
k1 = - (k * CA(i)^2) / Qv;
k2 = - (k * (CA(i) + (deltaV/2) * k1)^2) / Qv;
k3 = - (k * (CA(i) + (deltaV/2) * k2)^2) / Qv;
k4 = - (k * (CA(i) + deltaV * k3)^2) / Qv;
CA(i+1) = CA(i) + (deltaV/6) * (k1 + 2*k2 + 2*k3 + k4);
end

% Plot Results
```

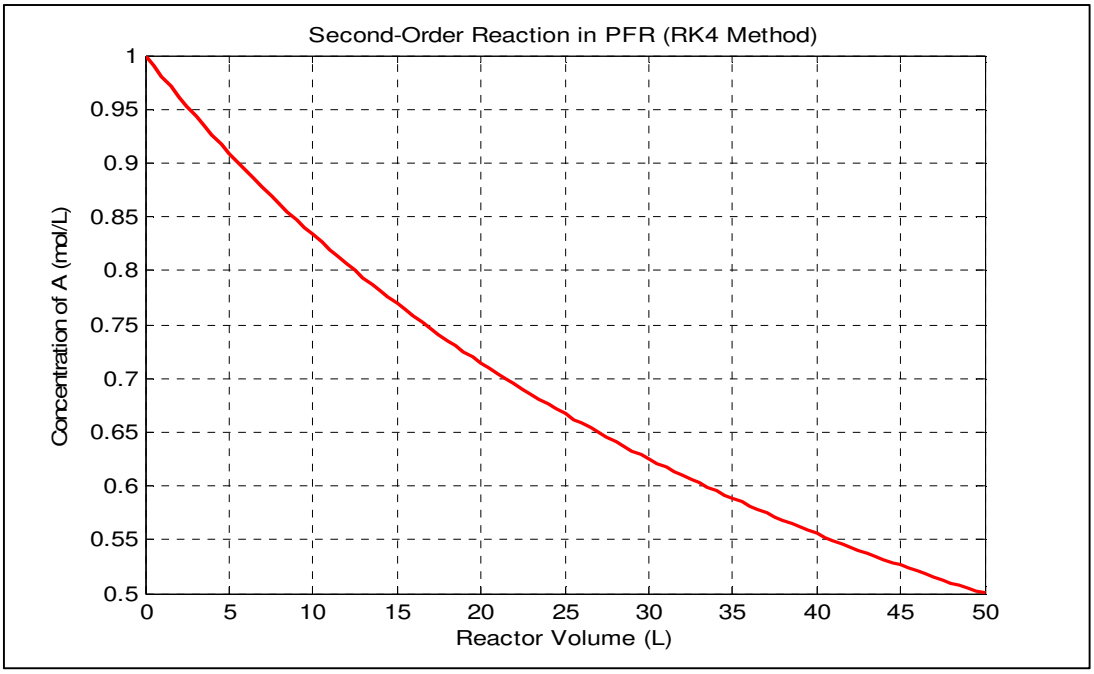


Figure I.6. Simulation results using Euler method.

Chapitre II:

Optimization

CHAPTER II. OPTIMIZATION

II.1. Optimization problems in process engineering

II.1.1. Definition of optimization

- Optimization is the use of specific methods to determine the most cost-effective and efficient solution to a problem or design for a process.

- *A typical engineering problem can be posed as follows:* A process can be represented by some equations or perhaps by experimental data. You have a single performance criterion in mind. The goal of optimization is to find the values of the variables in the process that yield the best value of the performance criterion.

II.I.2. Utility of optimization

Typical projects in which optimization have been used:

1. Determining the best sites for plant location.

2. Sizing or Designing equipment and an entire plant.

3. Scheduling maintenance and equipment replacement.

4. Operating equipment, such as reactors, columns, and absorbers.

5. Evaluating plant data to construct a model of a process.

6. Minimizing inventory charges.

7. Allocating resources or services among several processes.

8. Planning and scheduling construction.

II.1.3. Examples in applications of optimization

Example 1. Optimal insulation thickness

The addition of insulation should save money through reduced heat losses; on the other hand, the insulation material can be expensive. The amount of added insulation needed can be determined by optimization.

Assume that the bare surface of a vessel is at 370°C with an ambient temperature of 20°C. The surface heat loss is 2.6 kW/m^2. Add 1 inch (2.5 cm) of calcium silicate insulation and the loss will drop to 0.8 kW/m^2. At an installed cost of $43.00 m^2 and a cost of energy at $5.00/$10^6$ kJ, a savings of $284 per year (8760 hours of operation) per square meter would be realized.

Note: As additional inches of insulation are added, the increments must be justified by the savings obtained.

- Figure 1 shows the outcome of adding more layers of insulation. Since insulation can only be added in 0.5 in. increments, the possible capital costs are shown as a series of dots; these costs are prorated because the insulation lasts for several years before having to be replaced.

- The energy loss cost is a continuous curve because it can be calculated directly from heat transfer principles.

The total cost is also shown as a continuous function. Note that at some point total costs begin increasing as the insulation thickness increases because little or no benefit in heat conservation results.

The trade-off between energy cost and capital cost, and the optimum insulation thickness, can be determined by optimization.

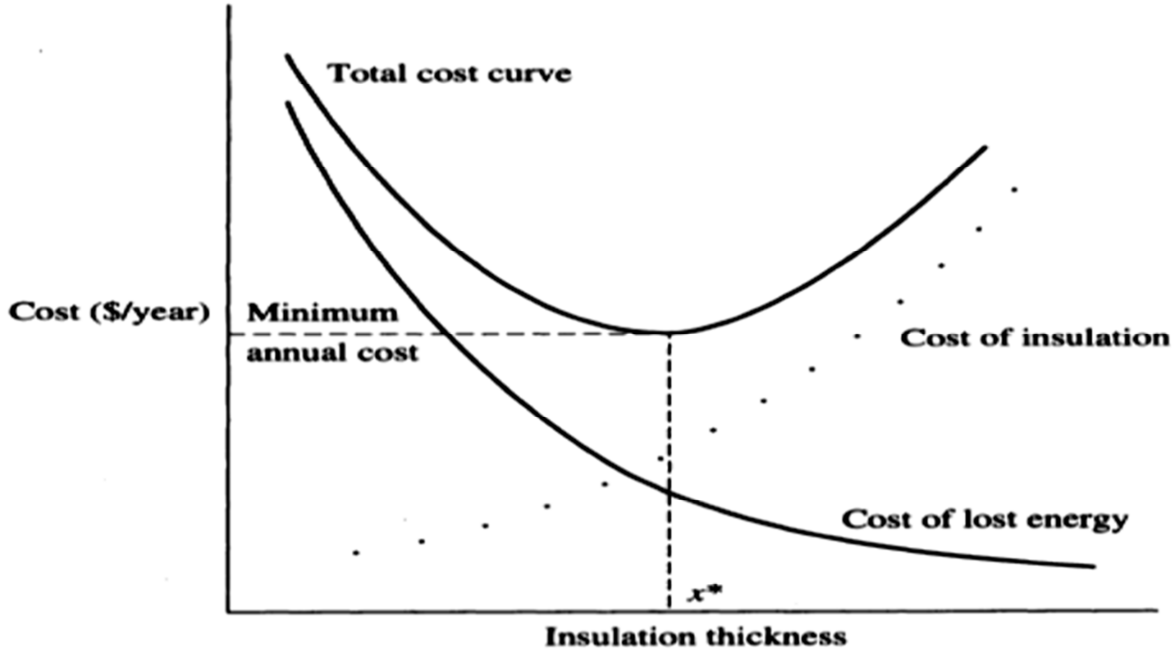


Figure II.1. The effect of insulation thickness on total cost (x* = optimum thickness).

Example 2. Optimal operating conditions of a boiler

In a boiler it is desirable to optimize the air-fuel ratio so that the thermal efficiency is maximized;

However, environmental regulations encourage operation under fuel-rich conditions and lower combustion temperatures in order to reduce the emissions of nitrogen oxides (NOx). Such operating conditions also decrease efficiency because some unburned fuel escapes through the stacks, resulting in an increase in undesirable hydrocarbon (HC) emissions.

Thus, a conflict in operating criteria arises!

- Figure 2 illustrates the trade-offs between efficiency and emissions, suggesting that more than one performance criterion may exist: We are forced to consider maximizing efficiency versus minimizing emissions, resulting in some compromise of the two objectives.

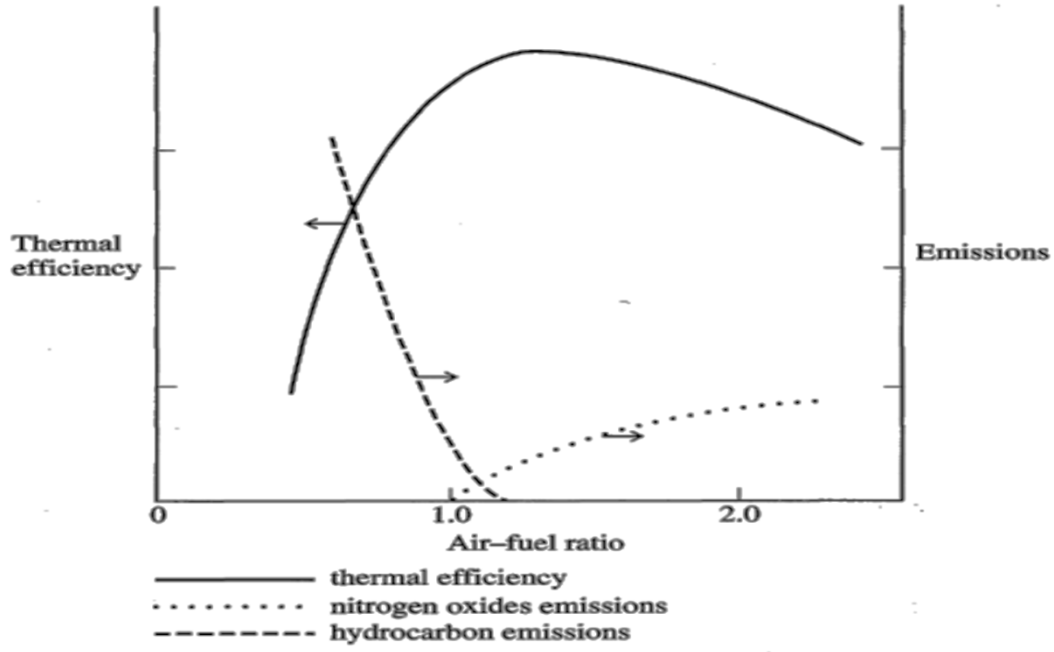


Figure II. 2. Effeciency and emissions of a boiler as a function of air-fuel ratio.

Example 3. Optimum distillation reflux

The need for energy conservation suggests a different objective, namely minimizing the reflux ratio in distillation column.

In this circumstance, one can ask: How low can the reflux ratio be set?

From the viewpoint of optimization, there is an economic minimum value below which the energy savings are less than the cost of product quality degradation.

Figures 3a and 3b illustrate both alternatives. Operators tend to over reflux a column because this strategy makes it easier to stay well within the product specifications.

Often columns are operated with a fixed flow control for reflux so that the reflux ratio is higher than needed when feed rates drop off.

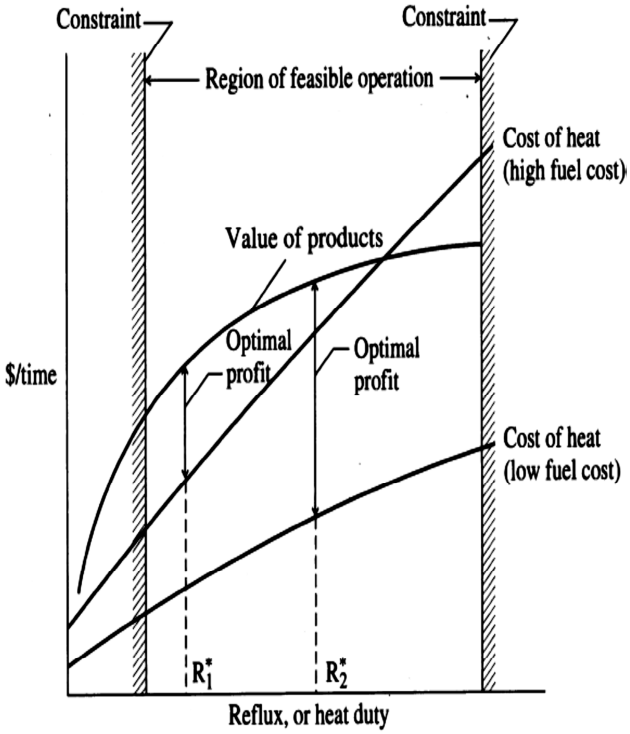


Figure II. 3a . Illustration of optimal reflux for different fuel costs

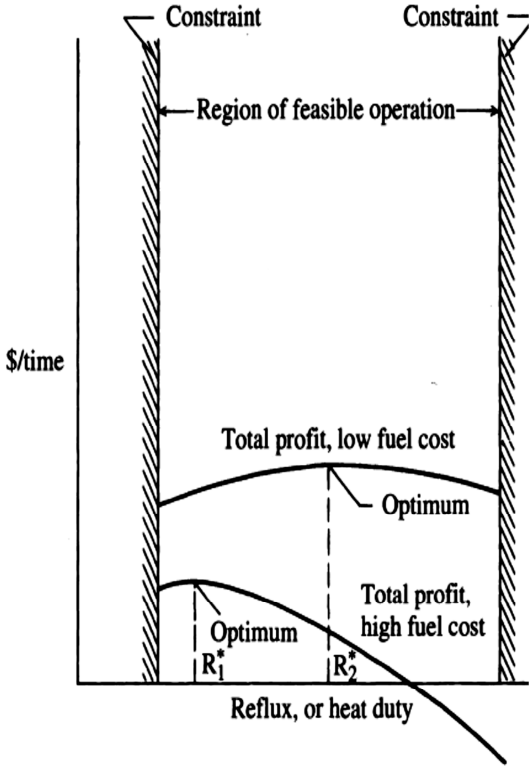
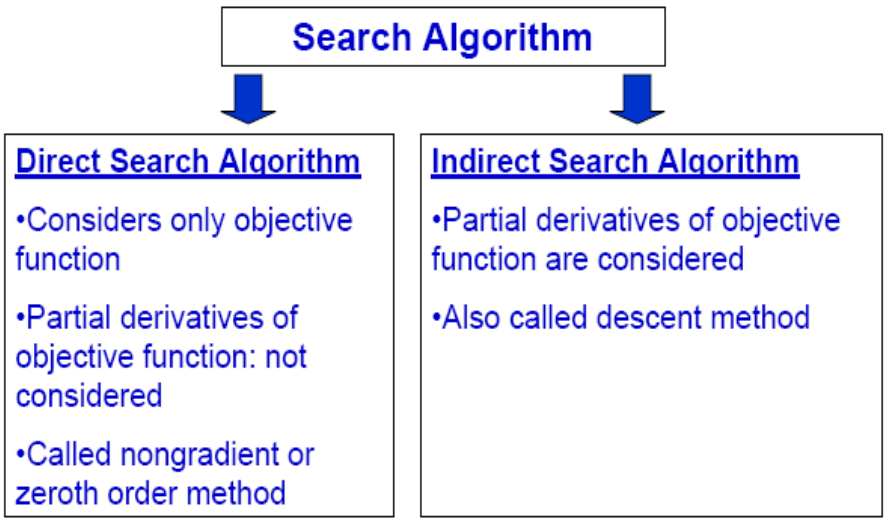


Figure II. 3b . Total profit for different fuel costs

II.2. Direct search for one-dimensional and multi-dimensional optimization

Classification of Search Algorithm:



Direct methods are usually applied in the following circumstance:

• The function f(x) is not differentiable;

• The derivatives of *f* are complicated to compute or even do not exist;

• The function has few variables;

• The location of an optimal solution is roughly known.

II.2.1. Direct search for one-dimensional optimization

In this section we discuss the solution of the unconstrained mono-variable optimization problem.

1. Golden section method

We consider the following minimization problem of function f of one variable x:

$$\textbf{Min} \quad f(x), \quad \textbf{Subject to:} \quad x \in [a0 \ , \ b0]$$

The only property that we assume of the objective function *f* is that it is unimodal, which means that *f* has only one local minimizer.

We choose the intermediate points in such a way that the reduction in the range is symmetric, in the sense that:

$$a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0) \quad , \qquad \rho < \frac{1}{2}$$

Determination of ρ

To find the value of ρ that results in only one new evaluation of f: $\rho(b_1 - a_0) = b_1 - b_2$

Because: $b_1 - a_0 = 1 - \rho$ *and* $b_1 - b_2 = 1 - 2\rho$, *we have*: $\rho(1 - \rho) = 1 - 2\rho$

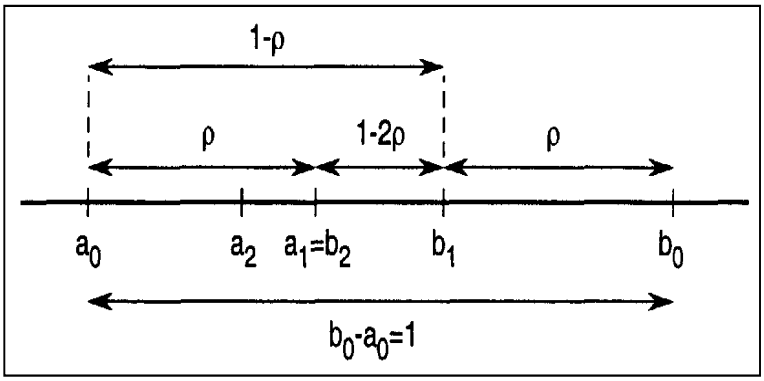


Figure II.4. Finding value of ρ

We write the above quadratic function of ρ as:

$\rho^2 - 3\rho + 1 = 0$

The solutions are:

$$\rho_1 = \frac{3+\sqrt{5}}{2}, \qquad \rho_2 = \frac{3-\sqrt{5}}{2}.$$

Because we require $\boldsymbol{\rho < \frac{1}{2}}$, we take:

$$\boldsymbol{\rho = \frac{3-\sqrt{5}}{2} \approx 0,382}$$

- The uncertainty range is reduced by the ratio ***1 — ρ ≈ 0.61803*** at every stage.

- Hence, N steps of reduction using the Golden Section method reduces the range by the factor:

$$\boldsymbol{(1-\rho)^N \approx (0.61803)^N}$$

Algorithm

Step 1: Start with a closed interval $[a_0,b_0]$ such that f(x) is unimodal within this interval;

Step 2: Evaluate two points of the interval:

$$\boldsymbol{a_1 = a_0 + \rho(b_0 - a_0)} \ , \ \boldsymbol{b_1 = a_0 + (1-\rho)(b_0 - a_0)}$$

Step 3: Evaluate the function at $f(a_1)$ and $f(b_1)$

- *If* $f(a_1) < f(b_1)$ then the minimizer must lie in the range $[a_0, b_1]$.
- *If* $f(a_1) \geq f(b_1)$ then the minimizer is located in the range $[a_1, b_0]$:

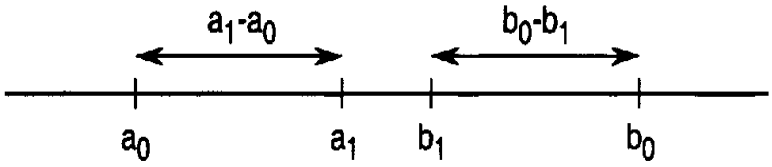


Figure II.5. Evaluating f at two intermedia.

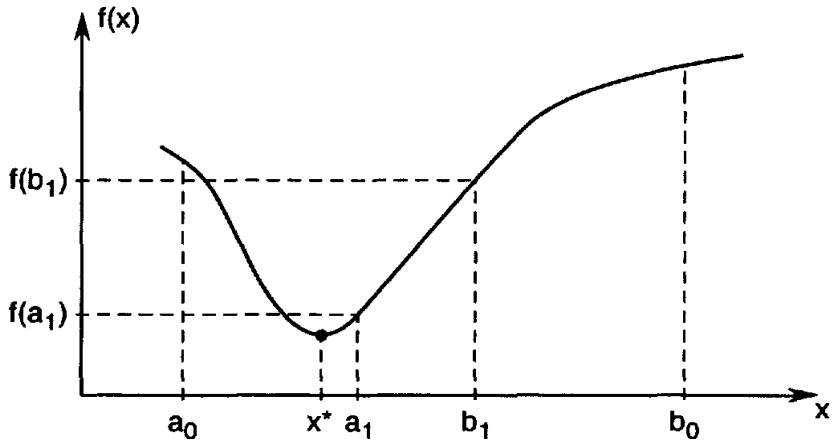


Figure II.6. The case where f(a1) < f(b1); x*∈ [a0 , b1].

Step 4: Repeat steps 2–3 until the interval [a,b] becomes smaller than a specified tolerance ϵ ($|b-a| < \epsilon$), and the midpoint m is taken as an approximation of the minimum.

Note: If the optimization problem is a maximization problem, the condition for choosing the interval is reversed:

- *If* $f(a_1) < f(b_1)$ then the maximizer must lie in the range $[a_1, b_0]$:
- *If* $f(a_1) \geq f(b_1)$ then the minimizer is located in the range $[a_0, b_1]$.

Example:

Use the Golden Section search to find the value of x that minimizes:

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

In the range [0,2], Locate this value of x to within a range of 0.3.

Solution

After N stages the range [0,2] is reduced by (0.61803)^N. So, we choose N so that:

$$(0.61803)^N \leq 0.3/2.$$

Four stages of reduction will do; that is, N = 4.

Iteration 1. We evaluate *f* at two intermediate points' a_1 and b_1. We have:

$a_1 = a_0 + \rho(b_0 - a_0) = 0.7639$

$b_1 = a_0 + (1 - \rho)(b_0 - a_0) = 1.236$

$We\ compute$:

$f(a_1) = -24.36$

$f(b_1) = -18.96$

Thus, $f(a_1) < f(b_1)$, and so the uncertainty interval is reduced to: $[a_0, b_1] = [0,1.236]$

Iteration 2. We choose b_2 to coincide with a_1, and *f* need only be evaluated at one new point:

$a_2 = a_0 + \rho(b_1 - a_0) = 0.4721$

We have:

$f(a_2) = -21.10$

$f(b_2) = f(a_1) - 24.36$

Now, $f(b_2) < f(a_2)$, so the uncertainty interval is reduced to: $[a_2, b_1] = [0.4721,1.236]$

Iteration 3. We set $a_3 = b_2$, and compute b_3:

$b_3 = a_2 + (1 - \rho)(b_1 - a_2) = 0.9443$

We have:

$f(a_3) = f(b_2) = -24.36$

$f(b_3) = -23.59$

So *f(b3) > f(a3).* Hence, the uncertainty interval is further reduced to: $[a_2, b_3] = [0.4721, 0.9443]$

Iteration 4. We set $b_4 = a_3$, and:

$$a_4 = a_2 + \rho(b_3 - a_2) = 0.6525$$

We have:

$$f(b_4) = f(a_3) = -24.36$$

$$f(a_4) = -23.84$$

Hence, $f(a_4) > f(b_4)$. Thus, the value of x that minimizes f is located in the interval:

$[a_4, b_3] = [0.6525, 0.9443]$ Note that $b_3 - a_4 = 0.292 < 0.3$

2.Fibonacci Search method

Suppose that we are allowed to vary the value ρ from stage to stage, so that at the kth stage in the reduction process we use a value ρk, at the next stage we use a value ρk+1, and so on.

- The goal is to select successive values of ρ_k, , $0 \leq \rho_k \leq 1/2$, such that only one new function evaluation is required at each stage.

- It is sufficient to choose the ρk such that:

$$\boldsymbol{\rho_{k+1}(1 - \rho_k) = 1 - 2\rho_k}$$

- After some manipulations, we obtain:

$$\boldsymbol{\rho_{k+1} = 1 - \frac{\rho_k}{1 - \rho_k}}$$

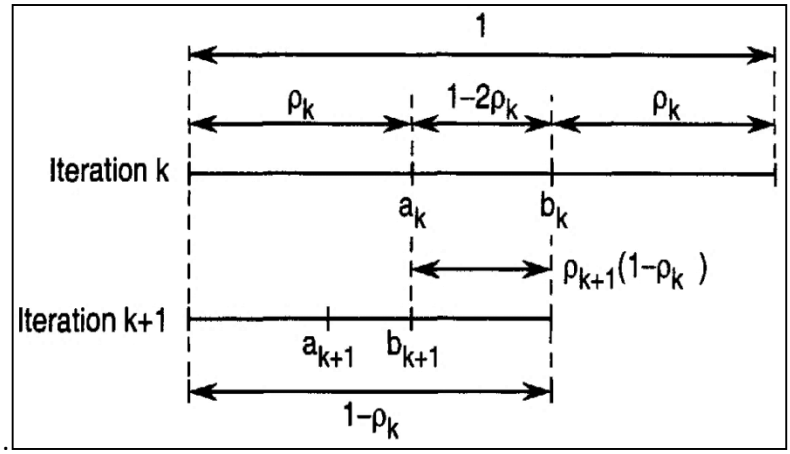


Figure II.7. Selecting evaluation points.

We are given a sequence ρ_1, ρ_2, …, satisfying the conditions $0 \leq \rho_k \leq 1/2$, and we use this sequence in our search algorithm.

After N iterations of the algorithm, the uncertainty range is reduced by a factor of:

$$(\mathbf{1}-\boldsymbol{\rho_1})(\mathbf{1}-\boldsymbol{\rho_2})\ldots(\mathbf{1}-\boldsymbol{\rho_N})$$

Depending on the sequence ρ1, ρ2, …, we get a different reduction factor.

What sequence ρ_1, ρ_2,.., minimizes the above reduction factor? This problem is a constrained optimization problem that can be formally stated:

$$\boldsymbol{minimize}:\ \left(\mathbf{1}-\boldsymbol{\rho_1}\right)\left(\mathbf{1}-\boldsymbol{\rho_2}\right)\ldots\left(\mathbf{1}-\boldsymbol{\rho_N}\right)$$

$$\boldsymbol{Subject\ to}:\ \boldsymbol{\rho_{k+1}} = \mathbf{1}-\frac{\boldsymbol{\rho_k}}{\mathbf{1}-\boldsymbol{\rho_k}}, \qquad \boldsymbol{k} = \mathbf{1},\ldots,\boldsymbol{N}-\mathbf{1}$$

$$\mathbf{0} \leq \boldsymbol{\rho_k} \leq \frac{\mathbf{1}}{\mathbf{2}}, \quad \boldsymbol{k} = \mathbf{1},\ldots,\boldsymbol{N}$$

To give the solution to the above optimization problem, we first need to introduce the Fibonacci sequence, F_1,F_2,F_3, ….

This sequence is defined as follows. First, let $F_{-1} = 0$ and $F_0 = 1$ by convention.

Then, for $k \geq 0$: $F_{k+1} = F_k + F_{k-1}$

Some values of elements in the Fibonacci sequence are as follows:

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
1	2	3	5	8	13	21	34

It turns out that the solution to the above optimization problem is:

$$\rho_1 = 1 - \frac{F_N}{F_{N+1}}$$

$$\rho_2 = 1 - \frac{F_{N-1}}{F_N}$$

.

$$\rho_k = 1 - \frac{F_{N-k+1}}{F_{N-k+2}}$$

.

$$\rho_N = 1 - \frac{F_1}{F_2}$$

Where the F_k are the elements of the Fibonacci sequence.

The resulting algorithm is called the Fibonacci search method.

-The reduction factor for the Fibonacci method is:

$$\frac{\mathbf{1 + 2\varepsilon}}{\boldsymbol{F_{N+1}}}$$

Note:

- The Fibonacci method is better than the Golden Section method in that it gives a smaller final uncertainty range.

Example:

Use the Fibonacci search method to find the value of x that minimizes f over the range [0,2]. Locate this value of x to within a range 0.3:

$$\boldsymbol{f(x) = x^4 - 14x^3 + 60x^2 - 70x}$$

Solution:

After N steps the range is reduced by $(1 + 2\varepsilon\}/F_{N+1}$ in the worst case. We need to choose N such that:

$$\frac{1 + 2\varepsilon}{F_{N+1}} \leq \frac{Final\ Range}{Initial\ range} = \frac{0.3}{2} = 0.15$$

Thus, we need:

$$F_{N+1} \geq \frac{1 + 2\varepsilon}{0.15}$$

If we choose $\varepsilon < 0.1, then\ N = 4\ will\ do.$

Iteration1. We start with : $1 - \rho_1 = \frac{F_4}{F_5} = \frac{5}{8}$,

We then compute:

$$a_1 = a_0 + \rho_1(b_0 - a_0) = \frac{3}{4}$$

$$b_1 = a_0 + (1 - \rho_1)(b_0 - a_0) = \frac{5}{4}$$

$$f(a_1) = -24.34$$

$$f(b_1) = -18.65$$

$$f(a_1) < f(b_1)$$

The range is reduced to: $[a_0, b_1] = \left[0, \frac{5}{4}\right]$

Iteration 2. We start with : $1 - \rho_2 = \frac{F_3}{F_4} = \frac{3}{5}$,

We then compute:

$$a_2 = a_0 + \rho_2(b_1 - a_0) = \frac{1}{2}$$

$$b_2 = a_1 = \frac{3}{4}$$

$$f(a_2) = -21.69$$

$$f(b_2) = f(a_1) = -24.34$$

$$f(a_2) > f(b_2)$$

The range is reduced to: $[a_2, b_1] = \left[\frac{1}{2}, \frac{5}{4}\right]$

Iteration 3. We start with : $1 - \rho_2 = \frac{F_3}{F_4} = \frac{3}{5}$,

We then compute:

$$a_2 = a_0 + \rho_2(b_1 - a_0) = \frac{1}{2}$$

$$b_2 = a_1 = \frac{3}{4}$$

$$f(a_2) = -21.69$$

$$f(b_2) = f(a_1) = -24.34$$

$$f(a_2) > f(b_2)$$

The range is reduced to: $[a_2, b_1] = \left[\frac{1}{2}, \frac{5}{4}\right]$

Iteration 4. We start with : $1-\rho_2=\frac{F_3}{F_4}=\frac{3}{5}$,

We then compute:

$a_2=a_0+\rho_2(b_1-a_0)=\frac{1}{2}$

$b_2=a_1=\frac{3}{4}$

$f(a_2)=-21.69$

$f(b_2)=f(a_1)=-24.34$

$f(a_2)>f(b_2)$

The range is reduced to: $[a_2,b_1]=\left[\frac{1}{2},\frac{5}{4}\right]$

Iteration 5. We start with : $1-\rho_3=\frac{F_2}{F_3}=\frac{2}{3}$,

We then compute:

$b_3=a_2+(1-\rho_3)(b_1-a_2)=1$

$a_3=b_2=\frac{3}{4}$

$f(a_3)=f(b_2)=-24.34$

$f(b_3)=f(a_1)=-23$

$f(a_3)>f(b_3)$

The range is reduced to: $[a_2,b_3]=\left[\frac{1}{2},1\right]$

Iteration 6. We choose $\varepsilon=0.05$, we have: $1-\rho_4=\frac{F_1}{F_2}=\frac{2}{3}$,

We then compute:

$a_4=a_2+(\rho_4-\varepsilon)(b_3-a_2)=0.725$

$b_4=b_3=\frac{3}{4}$

$f(a_4)=-24.27$

$f(b_4)=f(a_3)=-24.34$

$f(a_4)>f(b_4)$

The range is reduced to: $[a_4,b_3]=[0.725,1]$

Note $b_3-a_4<0.3$

3. Dichotomy Method (Bisection Method)

Minimize a unimodal function f(x) over an interval [a,b] by iteratively halving the interval until it is sufficiently small.

Algorithm

Step 1: Start with a closed interval $[a_0,b_0]$ such that f(x) is unimodal within this interval;

Step 2: Choose a small positive number δ (this is a small tolerance value for how much we'll move from the midpoint on either side to test for the minimum);

Step 3: Evaluate the function at two points slightly offset from the middle of the interval:

$$a_1 = \frac{b_0+a_0}{2} - \delta \quad and \quad b_1 = \frac{b_0+a_0}{2} + \delta$$

Step 4: Evaluate the function at $f(a_1)$ and $f(b_1)$

- *If* $f(a_1) < f(b_1)$ then the minimizer must lie in the range $[a_0, b_1]$: update $b_0=(a_0+b_0)/2$
- *If* $f(a_1) \geq f(b_1)$ then the minimizer is located in the range $[a_1, b_0]$: update $a_0=(a_0+b_0)/2$

Step 5: Repeat steps 2–3 until the interval [a,b] becomes smaller than a specified tolerance ϵ ($|b-a| < \epsilon$), and the midpoint m is taken as an approximation of the minimum.

Convergence

- The method converges linearly, reducing the interval by half with each iteration.

- It is efficient for unimodal functions and offers a robust approach when derivatives are available.

Note

Compared to methods like the Golden Section and Fibonacci search, which are also interval-reduction techniques, the dichotomy method is simpler but may require more iterations for similar precision, especially if δ is not well-chosen.

4.Equal intervals Method

A simple optimization technique that divides the search interval into a fixed number of subintervals to locate the minimum (or maximum) of a function.

The objective is to minimize (or maximize) a function f(x) on a closed interval $[a_0,b_0]$ by dividing it into a series of equal subintervals and evaluating the function at each subinterval point.

Algorithm:

Step 1: Choose the number of divisions n for the interval *[a,b]*. Larger n values give higher precision but require more function evaluations.

Step 2: Calculate the step size $\boldsymbol{h= (b-a) /n}$.

Step 3: Evaluate the function f(x) at each point $\boldsymbol{x_i = a + i\cdot h \quad for \;\; i=0,1,2,\ldots,n.}$

Step 4: Identify the point *x(min)* where *f(x)* has the lowest value among the points.

Step 5: The algorithm stops after evaluating f(x) at each of the n points. The interval containing the minimum is then refined based on the smallest *f(x)* found.

II.2.2. Direct search for multi-dimensional optimization

In this section we discuss the solution of the unconstrained multivariable optimization problem:

$$\boldsymbol{Find: X^* = [x_1^* \;\; x_2^* \ldots . x_n^*]^T \; that \; minimizes \;\; f(x_1, x_2 , \ldots , x_n) \equiv f(X)}$$

1. Orthogonal direction methods

Class of optimization techniques that iteratively search for the minimum or maximum of a function by moving in mutually orthogonal (perpendicular) directions at each step.

1. *Reduction to One-Dimensional Problems*: method is to optimize one variable at a time while keeping the others fixed. By following each coordinate axis or other orthogonal direction, the method turns a complex multidimensional optimization problem into a sequence of simpler, one-dimensional searches.

2. Orthogonality: Each step or iteration is taken along a direction that is orthogonal to the previous ones. Orthogonality ensures that each direction is independent and does not interfere with previously optimized directions, effectively covering the search space in an efficient manner.

3. Choice of Directions: The simplest choice for directions is the coordinate axes in the problem space, often known as coordinate descent.

4. Iterative Process: moving from one direction to the next until a certain convergence criterion is met.

Algorithm of Orthogonal Direction Method

1. *Initialization:* Start from an initial point, often denoted as X_0,

2. Direction Selection: Choose an initial direction along one of the coordinate axes (e.g., x-axis, y-axis in a 2D space). This direction is aligned with the variable to be optimized first.

3. Line Search in the Direction: Perform a line search along the chosen direction to find the best step size. A line search determines the optimal point along that specific direction by evaluating the function value at various points on the line. The outcome is the point that minimizes (or maximizes) the objective function along that line.

4. Update the Position: Move to the new position found through the line search. This position represents the point where the objective function has been optimized along the current direction.

5. Orthogonalization of New Directions: the next direction is chosen to be orthogonal to the previous one.

6. Repeat: continue with a new direction, often looping back to the initial direction once all directions have been used. This cycle repeats until a stopping criterion is met.

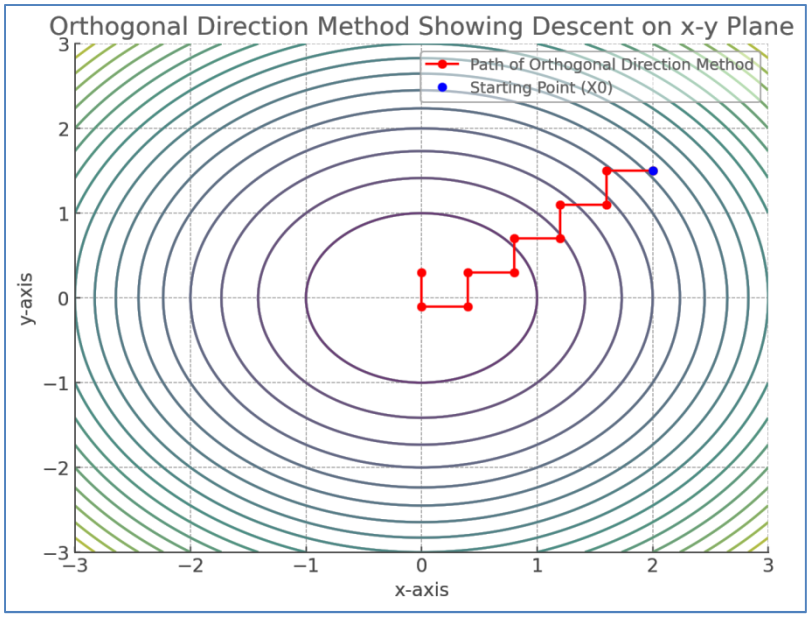


Figure II.8. Orthogonal Direction Method.

Advantages and Limitations of Orthogonal Direction Methods

Advantages

Simplicity: Orthogonal direction methods are conceptually straightforward and often easy to implement, particularly for simple coordinate-based searches.

No Need for Gradient Information: These methods do not require gradients, making them suitable for functions that are not differentiable or where gradients are hard to compute.

Stability: Moving orthogonally reduces oscillations and ensures steady progress toward the optimum.

Limitations

Slow Convergence for Non-Separable Functions: If the objective function has strong interdependencies between variables, moving along orthogonal directions can lead to slow convergence or zigzagging behavior.

Curse of Dimensionality: For very high-dimensional problems, exploring every orthogonal direction can become computationally expensive and may still not efficiently cover the search space.

Types of Orthogonal Direction Methods

a. Coordinate Descent:

In coordinate descent, each step involves moving along one coordinate direction at a time. It is easy to implement, when the objective function can be separated by each variable.

Algorithm

1. Initialization: Begin with an initial guess for all variables, $X(0)=(x_1(0),x_2(0),\ldots,x_n(0))$.

2. Coordinate Optimization: For each variable x_i:

- Perform a line search along the i-th coordinate direction, keeping the values of all other variables fixed.
- Update x_i to the optimal value found in the line search that minimizes (or maximizes) the objective function along that coordinate.

3. Cycle through All Variables: Complete a full cycle of line searches for each variable, one at a time. This constitutes one iteration.

4. Check for Convergence: Evaluate whether the objective function's value or the changes in the variables have met a convergence criterion. If so, terminate the algorithm; if not, continue with another iteration.

Example:

Suppose we want to minimize the following function: $f(x,y) = (x-3)^2 + (y+2)^2$ with an initial guess x=0, y = 0.

Solution:

Coordinate Descent Iteration: We'll minimize *f(x,y)* by optimizing x and y alternately, using line search in each direction.

Iteration 1:

Step 1: - Optimize x while keeping y fixed: Substitute y=0 into the function:

$$f(x,0) = (x-3)^2 + 4$$

- Now, minimize *f(x,0)* with respect to *x*: $df(x,0)/dx = 2(x-3) = 0 \Rightarrow x = 3$
- So, the optimal value of *x* for this step is ***x=3***.

Step 2: - Optimize y while keeping x fixed: Substitute *x=3* into the function:

$$f(3,y) = (y + 2)^2$$

- Now, minimize *f(3,y)* with respect to y: $df(3,y)/dy = 2(y+2) = 0 \Rightarrow y = -2$
- So, the optimal value of y for this step is ***y=−2***.
- After this iteration, we have *x=3 and y=−2*.

Step 3: - Check for Convergence: We can now evaluate the function *f(x,y)* at this point:

$$f(3,-2) = 0$$

- Since we reached a function value of zero, which is the minimum possible for this function, we can stop here.

This is also the exact global minimum for the function.

The minimum of $f(x,y)$ is at: ***(x,y)*=(3,−2)*** with a minimum function value of: ***f(3,−2)=0***

b. Pattern Search (Hooke-Jeeves Method)

The Hooke-Jeeves method refines the orthogonal direction approach by combining a pattern search with coordinate search.

It involves making exploratory moves in all coordinate directions to gather initial information, then following the pattern of improvement along these directions for faster convergence.

Algorithm

1. Exploratory Search:

- Starting from an initial point, the method probes the search space by making small adjustments along each variable's direction (positive and negative ± Δ).

- For each variable, it evaluates the objective function at these new points, seeking to improve upon the current solution.

- If an improved point is found, this becomes the new base point. Otherwise, the original point remains the base point.

2. Pattern Move:

- Once a better solution is found in the exploratory search, the method takes a step in the same direction, attempting a pattern move to accelerate progress toward an optimal solution.

- This step size is usually scaled up to take advantage of the direction of improvement, allowing faster convergence if the pattern holds.

3. Iteration and Step Adjustment:

- The method iterates, alternating between exploratory searches and pattern moves.

- If no improvement is found during the exploratory search, the step size is reduced, gradually refining the search to converge more precisely on the optimal solution.

4. Convergence Criteria:

- The algorithm continues until the step size falls below a predefined threshold, or there is no significant improvement in function values, indicating convergence to a (local) optimum.

Example

Suppose we want to minimize the function $f(x,y) = (x-3)^2 + (y+4)^2$ with an initial guess x=0, y = 0 and a step size Δ=1.

Solution

Step 1: - Initialize:

- Starting Point: $(x_0,y_0)=(0,0)$, Initial Step Size: Δ=1
- Exploration Directions: We will vary x and y by ± Δ in each direction.

Step 2: Exploratory Search

Starting from (0,0), we evaluate the function around this point by taking small steps in each variable's direction:

1. *Exploring x:*

$Try\ x = x0 + \Delta = 1$: $f(1,0) = 20$;

$Try\ x = x0 - \Delta = -1$: $f(-1,0) = 32$

The point *(1,0)* has a lower function value than the starting point, so we set ***x=1*** as the new base point for further exploration.

2. Exploring y:

$Try\ \ y\ = y0\ + \Delta\ = 1: f(1,1) = 29$

$Try\ \ y\ = y0\ - \Delta = -1:\ f(1,-1) = 13$

The point *(1,−1)* gives a lower function value than (1,0), so we update to the base point ***(1,−1).***

Step 3: Pattern Move

Since we found an improved point, we perform a pattern move in the direction of improvement by increasing the step in the direction from *(0,0) to (1,−1):*

- Pattern Move to $(2,-2) = \ (1\ +\ (1-0)\ ,-1\ +\ (-1-0))$
- Evaluate $f(2,-2) = 5$

The function at *(2,−2)* is lower than at *(1,−1),* so we accept ***(2,−2)*** as the new base point.

Step 4: Repeat Exploratory Search : Starting from (2,–2):

1. *Exploring x:* $Try\ x\ = 2\ +\ 1 =\ 3: f(3,-2) =\ 4\ ;\ Try\ x = 2-1 = 1: f(1,-2) = 8$

The best move is *(3,−2),* which becomes our new base point.

2. Exploring y: $Try\ y = -2+1 = -1: f(3,-1) = 9\ ;\ Try\ y = -2-1 = -3:\ f(3,-3) = 1$

The point ***(3,−3)*** has the lowest function value, so we update our base point to ***(3,−3).***

Step 5: Another Pattern Move:

- Perform a pattern move from *(2,−2) to (3,−3):* Move *to (4,−4): f(4,−4) = 1*
- The function value is equal to *(3,−3),* so we revert to ***(3,−3)*** and reduce the step size.

Step 6: Exploratory Search with Reduced Step Size

- *Δ=0.5*, Starting Point: *(x,y)=(3,−3)* ⇒ Exploratory Search around *(3,−3)*

1. *Exploring x:*

$Try\ x = 3 + 0.5 = 3.5 :\ f(3.5, -3) = 1.25\ ;\ Try\ x = 3 - 0.5 = 2.5 :\ f(2.5, -3) = 1.25$

Neither point has a lower function value than *f(3,−3)=1*, so we keep ***x=3***.

2. *Exploring y:*

$Try\ y = -3 + 0.5 = -2.5 :\ f(3, -2.5) = 2.25\ ;\ Try\ y = -3 - 0.5 = -3.5 :\ f(3, -3.5) = 0.25$

The point *(3,−3.5)* has a lower function value of 0.25, so we update our base point to ***(3,−3.5).***

Step 7: Pattern Move: from *(3,−3) to (3,−3.5)*

- Pattern Move to $(3, -4) = (3 + (3 - 3), -3.5 + (-3.5 - (-3))) = (3, -4)$

- Evaluate *f(3,−4): f(3,−4)=0*

The function value at *(3,−4)* is 0, which is lower than the value at *(3,−3.5)=0.25*

Since we reached a function value of zero, which is the minimum possible for this function, we can stop here.

The minimum of f(x,y) is at: ***(x,y)*=(3,−4)*** with a minimum function value of: ***f(3,−2)=0***

2. Simplex Method Search

The Nelder-Mead Simplex Method works by evaluating a simplex (a geometric shape with n+1 vertices in n-dimensional space) and iteratively adjusting this simplex to move toward the function's minimum or maximum value.

- Each vertex of the simplex represents a point in the search space, and the algorithm modifies the shape and location of the simplex based on the function values at its vertices.

Algorithm

1 Initialization: Start with an initial simplex, which consists of *n+1* points (vertices) for an n-dimensional problem.

- Typically, one vertex is the initial guess, and the other n points are generated by adding small perturbations to each dimension.

2. Function Evaluation:

Evaluate the objective function at each vertex of the simplex.

3. Ordering:

Order the vertices based on the function values from best (lowest for minimization problems) to worst (highest for minimization problems).

4. Transformations of the Simplex:

Based on the function values, transform the simplex by performing one of the following operations:

- ***Reflection:*** Reflect the worst point through the centroid of the other points to explore the search space opposite to the worst point:

$$\boldsymbol{Xreflect = Xcentroid + \alpha\,(Xcentroid - Xworst)}$$

Where α (typically $\boldsymbol{\alpha = 1}$) is the reflection coefficient.

- ***Expansion:*** If the reflection point gives a better function value than the best current point, expand the simplex further in that direction.

$$\boldsymbol{Xexpand = Xcentroid + \gamma\,(Xreflect - Xcentroid)}$$

Where $\gamma > 1$ (typically $\boldsymbol{\gamma = 2}$) is the expansion coefficient.

- ***Contraction:*** If the reflection point is not better than the second-worst point, contract the simplex towards the best point:

  - *Outside Contraction*: if the reflection point is better than the worst point but worse than the best point:

$$\boldsymbol{Xcontract = Xcentroid + \beta.(Xreflect - Xcentroid)}$$

  - *Inside Contraction*: if the reflection point is worse than the worst point:

$$\boldsymbol{Xcontract = Xcentroid + \beta.(Xworst - Xcentroid)}$$

Where β (typically $\boldsymbol{0 < \beta < 10}$) (often $\boldsymbol{\beta=0.5}$) is the contraction coefficient.

- ***Shrink:*** If contraction fails to improve the function value, shrink the entire simplex towards the best point to explore a smaller region:

$$\boldsymbol{Xi = Xbest + \delta(Xi - Xbest)\ for\ all\ i}$$

Where $\mathbf{0 < \delta < 1}$ (often $\boldsymbol{\delta = 0.5}$) is the shrink coefficient.

5. Iteration:

Repeat the process of reflection, expansion, contraction, or shrinkage until the simplex converges, meaning the difference between the highest and lowest function values within the simplex vertices is less than a tolerance.

Advantages and Limitations

Advantages

- Derivative-Free: Works without requiring the gradient, making it useful for optimizing non-differentiable or noisy functions.
- Simple to Implement: Straightforward transformations make it easy to understand and apply.

Limitations

- Local Optima: Since it's a local search method, it can converge to a local optimum, especially in non-convex problems.
- Scalability: Performance can be slow in high-dimensional spaces due to the simplex needing n+1 evaluations per iteration.

Example

Suppose we want to minimize the function $f(x,y) = x^2 + y^2 + 2xy + 3x + 4y$, starting with the initial simplex vertices $(x_0,y_0) = (0,0)$, $(x_1,y_1) = (1,0)$, and $(x_2,y_2) = (0,1)$.

Solution

1. Initialize: Compute $f(x,y)$ at each vertex: $f(0,0)=0$; $f(1,0)=4$; $f(0,1)=5$

2. Order: Arrange vertices by function values: ***Best: (0,0), Second-best: (1,0), Worst: (0,1)***

3. Reflection:

- Centroid: The current best two points are *(0,0)* and *(1,0)*.

- Centroid, excluding the worst point *(0,1)*: $Xcentroid = ((0+1)/2\,,(0+0)/2) = (0.5\,,0)$

- Reflect the worst point (0,1) through the centroid of (0, 0) and (1,0), using α=1 (standard for reflection): $Xreflect = (0.5\,,0) + (0.5\,,0) - (0,1) = (1\,,-1)$

- Evaluate the Function at the Reflection Point *f(1 , −1) = -1*

- Compare with Current function values: *f(0,0) = 0; f(1,0) = 4; f(0,1)= 5*

- The reflection point *(1,−1), f(1,-1)* is lower than *f(0,0)= 0* , the best current value.

4. Expansion (Since Reflection Was Successful)

- Since the reflection point *(1,−1)* is better than the current best point (0,0), we try an expansion to see if moving further in that direction improves the result.

- Calculate the Expansion Point: $Xexpand = (0.5,0) + 2 \cdot ((1,-1) - (0.5,0)) = (1.5,-2)$

- Evaluate the Function at the Expansion Point: *f(1.5,−2)=−3.25*

- Comparison and Update: Since f(1.5,−2) = −3.25 is better than f(1,−1) = −1, the expansion point is accepted, and (1.5 , −2) replaces the worst vertex in the simplex.

- New vertices are now: ***Best: (1.5,−2)*** , ***Second-best: (0,0)*** , ***Worst: (1,0)***

5. New iteration

1- Calculate the New Centroid (excluding the worst point *(1,0):*

$$Xcentroid = ((1.5+0)/2\,,(-2+0)/2) = (0.75,-1)$$

2- Reflect the Worst Point Across the Centroid Reflection Calculation: Reflect the worst point *(1,0)* through the centroid *(0.75,−1)*:

$$Xreflect = (0.75,-1) + 1 \cdot \big((0.75,-1) - (1,0)\big) = (0.5,-2)$$

- Evaluate the Function at the Reflection Point: *f(0.5,−2)=−4.25*

3 - Compare with Current Function Values: *f(1.5,−2)=−3.25 : f(0,0)=0 ; f(1,0)=4*

- The reflection point *(0.5,−2) has f(0.5,−2)=−4.25,* which is lower than the current best value *f(1.5,−2)=−3.25*

4- Expansion (Since Reflection Improved the Solution): Since the reflection point (0.5,−2) is better than the current best point (1.5,−2), we perform an expansion to see if moving further in that direction provides a better solution.

- Calculate the Expansion Point:

$$Xexpand = (0.75, -1) + 2 \cdot ((0.5, -2) - (0.75, -1)) = (0.25, -3)$$

- Evaluate the Function at the Expansion Point*: f(0.25,−3) = −3.6875*

5- Compare Reflection and Expansion Values: The reflection point *(0.5,−2)* has a lower function value than the expansion point *(0.25,−3)*

- Accept the Reflection Point *(0.5,−2)* as the new vertex and discard the worst point *(1,0).*

- Updated Simplex Vertices for the Next Iteration

The new simplex vertices are: ***Best: (0.5,−2)*** ;***Second-best: (1.5,−2) ; New worst: (0,0)***

Final Solution

After sufficient iterations, if further transformations and evaluations:

Optimal Solution***: (x*, y*) = (0.5, -2)*** , Optimal Function Value***: f(0.5,−2)=−4.25***

II.3. Mathematical approach to unconstrained optimization

II.3.1. Mathematical reminders

1. Local and global extremum

Figure 5 illustrates the character of flx) if the objective function is a function of a single variable. Usually we are concerned with finding the minimum or maximum of a multivariable function flx). The problem can be interpreted geometrically as finding the point in an n-dimension space at which the function has an extremum.

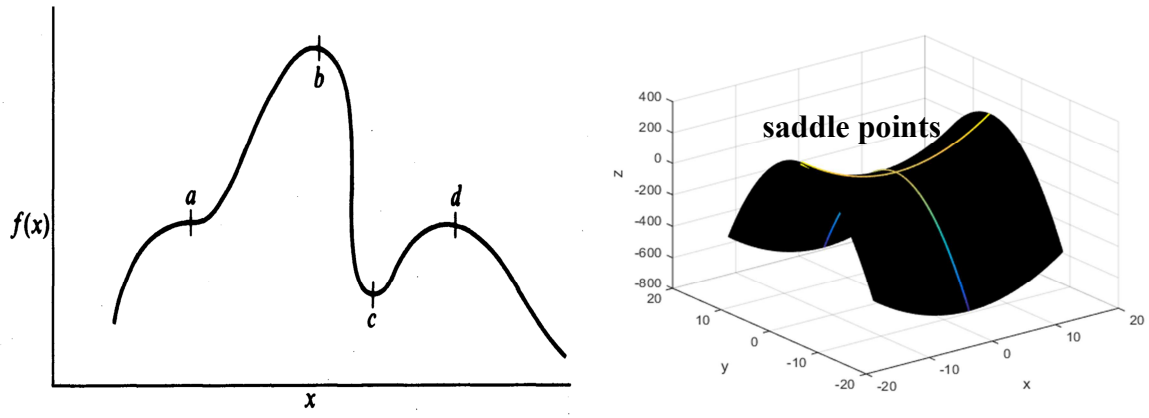


Figure II.9. A function exhibiting different types of stationary points.

a. Inflection point (scalar equivlent to a saddle points)

b. Global maximum (and local maximum)

c. Local minimum; d. Local maximum

Let f be a real-valued function defined as follows:

$$\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{R}$$
$$\boldsymbol{x} \rightarrow \boldsymbol{f}(\boldsymbol{x})$$

- *f(x*) is a local minimum* $\Leftrightarrow \exists\, \boldsymbol{\varepsilon} > \boldsymbol{0}, \forall \boldsymbol{x} \in \boldsymbol{X}: |\boldsymbol{x} - \boldsymbol{x}*| < \boldsymbol{\varepsilon} \Rightarrow \boldsymbol{f}(\boldsymbol{x}) \geq \boldsymbol{f}(\boldsymbol{x}*)\ \boldsymbol{and}\ \boldsymbol{x}* \in \boldsymbol{X}$

- *f(x*) is a global minimum* $\Leftrightarrow \quad \forall \boldsymbol{x} \in \boldsymbol{X}: \boldsymbol{f}(\boldsymbol{x}) \geq \boldsymbol{f}(\boldsymbol{x}*)\ \boldsymbol{and}\ \boldsymbol{x}* \in \boldsymbol{X} \ \Rightarrow\ \boldsymbol{f}(\boldsymbol{x}*) = \boldsymbol{y}* = \boldsymbol{min}_{\boldsymbol{x} \in \boldsymbol{X}}[\boldsymbol{f}(\boldsymbol{x})]$

Note: In mathematical programming, there is generally no guarantee that the solution obtained is global, except in a particular case, which is convex programming.

2. Unimodal and Multimodal Functions

A function is said to be unimodal (has a single optimum) if there exists a real number $x \in [x_a; x_\beta]$ for which the function is strictly decreasing on $[x_a; x^*]$, and strictly increasing on $[x^*; x_\beta]$. The point x is then the global minimum of *f*.

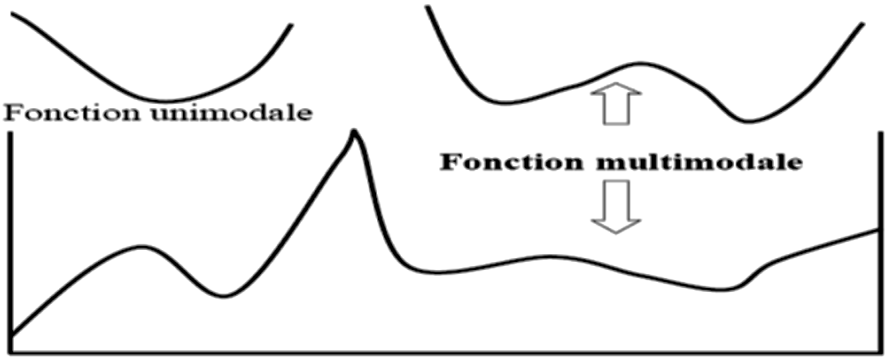


Figure II.10. Unimodal and Multimodal Functions.

1. Gradients

Suppose $f: R^n \rightarrow R$ be a continuous function. The function denoted $\nabla f(x): R \rightarrow R^n$ is called the gradient of f and is defined by:

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

Notes: it may not exist for certain $x \in R^n$.

2. Hessian matrix

Let $f: R^n \rightarrow R$ be a twice differentiable function. The function denoted $\nabla^2 f(x): R^n \rightarrow R^{nxn}$ is called the Hessian matrix of f and is defined by:

$$\mathbf{H}(\mathbf{x}) = \nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 x_n} \\ \frac{\partial^2 f(x)}{\partial x_n x_1} & \frac{\partial^2 f(x)}{\partial x_n x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$$

Note:

- The Hessian matrix is always symmetric
- If f is a C²-class function (has continuous second-order partial derivatives), then:

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(\hat{x}) = \frac{\partial^2 f}{\partial x_j \partial x_i}(\hat{x}), \quad i, j = 1,\ldots,n$$

Example:

$$f(x_1, x_2, x_3) = e^{x1} + {x_1}^2 . x_3 - x_1 x_2 x_3$$

Gradient of the function:

$$\nabla f(x_1, x_2, x_3) = \begin{pmatrix} e^{x1} + 2x_1 . x_3 - x_2 x_3 \\ -x_1 . x_3 \\ {x_1}^2 - x_1 x_2 \end{pmatrix}$$

Hessian of function *f*:

$$H(x) = \nabla^2 f(x) = \begin{pmatrix} e^{x_1} + 2x_3 & -x_3 & 2x_1 - x_2 \\ -x_3 & 0 & -x_1 \\ 2x_1 - x_2 & -x_1 & 0 \end{pmatrix}$$

5. Determinant of matrix

The determinant (denoted by det [A] or |A|) is reasonably easy to calculate by hand for matrices up to size 3 X 3:

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{32}a_{21} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12}$$

Example:

$$det \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = det \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} - 2det \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} + det \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} = -3$$

6.Eigenvalues of matrix

An nxn matrix A has n eigenvalues (e) such that:

$$\det(A - eI) = 0$$

Where:

I: an identity matrix of the same dimension of A.

Note: Eigenvalues provide unambiguous information about the nature of functions used in optimization:

- If all eigenvalues of A are positive, then A is positive-definite.
- If all eigenvalues of A are negative, then A is negative-definite.

Example:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \qquad (\mathbf{A} - e\mathbf{I}) = \begin{bmatrix} 1-e & 2 \\ 2 & 1-e \end{bmatrix}$$

$$\det\begin{bmatrix} 1-e & 2 \\ 2 & 1-e \end{bmatrix} = (1-e)^2 - 4 = e^2 - 2e - 3 = 0$$

$$(e-3)(e+1) = 0 \qquad e = 3, -1$$

Therefore, the eigenvalues are 3 and - 1.

1. Leading principal minors of matrix

Let A be the following 3×3 matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The leading principal minors of A are the following determinants:

$$a_{11}, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

- If all these determinants are positive, then the matrix A is positive definite;
- If they alternate in sign (with the first is negative, the second positive, the third negative), then the matrix A is negative definite.

2. Convex and concave function

A function f(x): $R^n \rightarrow R$ (defined on a continuous convex set), continuous and differentiable, is convex if: $\forall\, x_1, x_2 \; and \;\; \gamma \in [0,1]$, we have:

$$\boldsymbol{f[\gamma x_1 + (1-\gamma)x_2] \leq \gamma f(x_1) + (1-\gamma) f(x_2)}$$

A convex function cannot have any value larger than the values of the function obtained by linear interpolation between x_1 and x_2 (the cord between x1 and x_2).

Notes:

- If only the inequality sign holds, the function is said to be strictly convex.

- If f (x) is convex, - f (x) is concave.

- Convex function (so that the constraints form a convex set), the following property can be shown to be true: The local minimum of f(x) is also the global minimum.

- Analogously, a local maximum is the global maximum of f (x) if the objective function is concave and the constraints form a convex set.

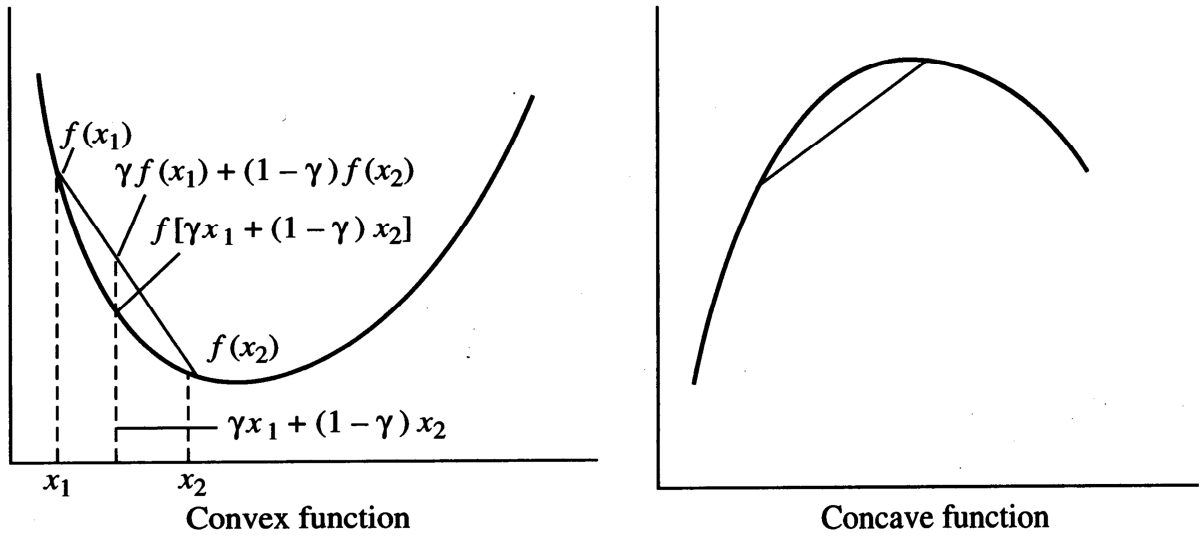


Figure II.11. Convex and concave functions of one variable.

- A function f(x) is convex if the Hessian matrix H(x) is a real symmetric positive-semidefinite matrix.

- Relationship between the character of f(x) and the state of H(x) to identify the character of extrema can best be evaluated by examining the eigenvalues of H(x):

Table II.1. Relationship between the character of f(x) and the state of H(x)

f(x) is	H(x) is	All the eigenvalues of H(x) are
Strictly convex	Positive-definite	**>0**
Convex	Positive-semidefinite	**≥0**
Concave	Negative-semidefinite	**≤0**
Strictly concave	Negative-definite	**<0**

Example:

For each of these functions, determine if f(x) is concave, trictly convex, strictly concave, all, or none of these classes in the range $-\infty \leq x \leq +\infty$

a. $f(x) = 3.x^2$
b. $f(x) = 2.x$
c. $f(x) = -5.x^2$
d. $f(x) = 2.x^2 - x^3$
e. $f(x) = 2.{x_1}^2 - 3x_1x_2 + 2{x_2}^2$

Solution

a. $f''(x) = 6$ ⇒ always positive, hence f(x) is both strictly convex and convex.

b. $f''(x) = 0$ ⇒ f(x) is convex and concave.

c. $f''(x) = -10$ ⇒ always negative, hence f(x) is both strictly concave and concave.

d. $f''(x) = 4 - 6x$ ⇒ may be positive or negative depending on the value of x, hence f(x) is not convex or concave over the entire range of x.

e.

$$\frac{\partial f(x)}{\partial x_1} = 4x_1 - 3x_2 \qquad \frac{\partial^2 f(x)}{\partial x_1^2} = 4 \qquad \frac{\partial^2 f(x)}{\partial x_2^2} = 4$$

$$\frac{\partial f(x)}{\partial x_2} = -3x_1 + 4x_2 \qquad \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} = \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} = -3 \qquad \mathbf{H(x)} = \begin{bmatrix} 4 & -3 \\ -3 & 4 \end{bmatrix}$$

The eigenvalues of H(x) are 7 and 1, hence H(x) is positive-definite. Consequently, f(x) is strictly convex (as well as convex).

II.3.2. Necessary and sufficient conditions for an extremum of an unconstrained function

- We illustrate the character of flx) if the objective function is a function of a single variable. Usually we are concerned with finding the minimum or maximum of a multivariable function flx).

- An optimal point x* is completely specified by satisfying what are called the necessary and sufficient conditions for optimality:

a. Necessary condition for a minimum or maximum of f (x) is that the gradient of f (x) vanishes at x*:

$$\nabla f(x^*) = 0$$

that is, x* is a stationary point (Thus, satisfaction of the necessary conditions does not guarantee a minimum or maximum).

b. ***Sufficient conditions:*** x* can be classified as:

Table II.2. conditions for an extremum of an unconstrained function.

$\nabla^2 f(x^*) = H(x^*)$	x^*
Positive-definite	Unique (isolated) minimum
Negative-definite	Unique (isolated) maximum

Summary

- These two conditions are known as the suficiency conditions.
- In summary, the necessary conditions (items 1 and 2 in the following list) and the sufficient condition (3) to guarantee that x* is an extremum are as follows:

1. **F(x)** is twice differentiable at x*.

2. ∇f(x*) = 0, that is, a stationary point exists at x*.

3. H(x*) is positive-definite for a minimum to exist at x*, and negative-definite for a maximum to exist at x*.

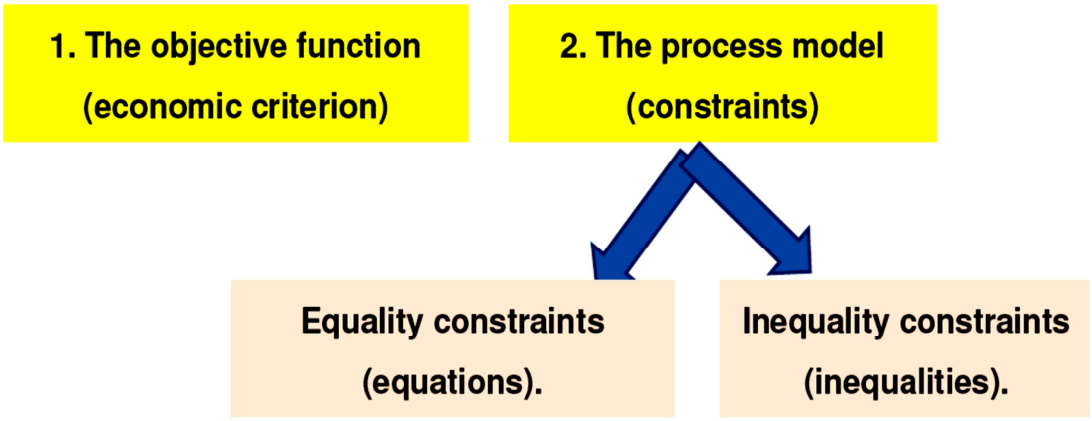
Note: A minimum or maximum may exist at x* even though it is not possible to demonstrate the fact using the three conditions.

II.4. Problems with equality constraints and inequality constraints

II.4.1. Problem formulation

Formulating the problem is the most crucial step in optimization.

Problem formulation requires identifying the essential elements of a conceptual or verbal statement of a given application and organizing them into a prescribed mathematical form, namely,



In this text the following notation will be used for each category of the optimization problem:

$$\begin{aligned} &\boldsymbol{Minimize}: \ \boldsymbol{f(x)} && \boldsymbol{Objective\ function} && (\boldsymbol{a}) \\ &\boldsymbol{Subject\ to}: \ \boldsymbol{h(x) = 0} && \boldsymbol{equality\ contraints} && (\boldsymbol{b}) \\ &\qquad\qquad \boldsymbol{g(x) \geq 0} && \boldsymbol{inequality\ consraints} && (\boldsymbol{c}) \end{aligned}$$

Where:

x is a vector of n variables $(x_1, x_2, \ldots\ldots, x_n)$,

$h(x)$ is a vector of equations of dimension m_1,

$g(x)$ is a vector of inequalities of dimension m_2. The total number of constraints is $m=m_1+m_2$

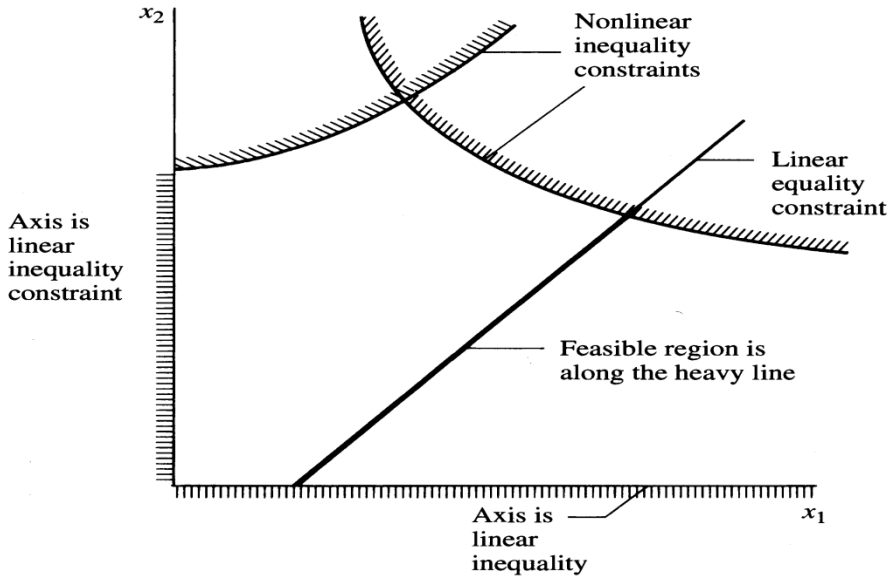
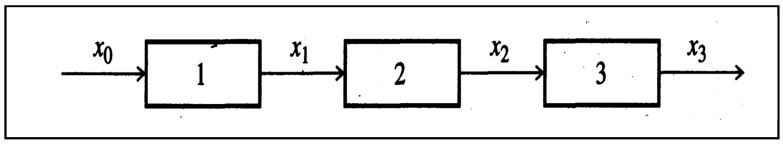


Figure II.12. Feasible region for optimizatin problem involving two independent variables. The dashed lines represent the side of the inequality constraints in the plane that form part of the infeasible region. The heavy line shows the feasible region.

Example:

In a rough preliminary design for a waste treatment plant the cost of the components are as follows (in order of operation):



1. Primary clarifier: $19.4.x_1^{-1.47}$ (DA);
2. Trickling filter: $16.8.x_2^{-1.66}$ (DA);
3. Activated sludge unit: $91.5.x_3^{-0.30}$ (DA).

Where the x's are the fraction of the 5-day biochemical oxygen demand (BOD) exiting each respective unit in the process, that is, the exit concentrations of material to be removed.
The required removal in each unit should be adjusted so that the final exit concentration x3 must be less than 0.05.

- Formulate the optimization problem listing the objective function and constraints.

Solution:

Decision Variables:

The decision variables are x_1, x_2 *and* x_3, which represent the fraction of BOD exiting each respective treatment unit.

Objective Function:

The goal is to minimize the total cost of the waste treatment process:

$$Minimize\ Cost = 19.4x_1^{-1.47} + 16.8x_2^{-1.66} + 91.5x_3^{-0.3}$$

Constraints:

- Final BOD Constraint: The final exit concentration must be less than 0.05:

$$x_3 \leq 0.05$$

- Feasibility Bounds: Since the exit fractions represent BOD fractions, they must be in the range:

$$0 < x_1, x_2, x_3 \leq 1$$

II.4.2. General Procedure for solving optimization problems with constraints

No single method or algorithm of optimization can be applied efficiently to all problems. The method chosen for any particular case depends primarily on:

1. The character of the objective function and whether it is known explicitly;
2. The nature of the constraints;
3. The number of independent and dependent variables.

Steps used to solve optimization problems

1. Analyze the process itself so that the process variables and specific characteristics of interest are defined; that is, make a list of all of the variables;

2. Determine the criterion for optimization, and specify the objective function in terms of the variables defined in step 1 together with coefficients. This step provides the performance model (economic model);

3. Using mathematical expressions, develop a valid process or equipment model that relates the input-output variables of the process and associated coefficients. Include both equality and

inequality constraints. Use well-known physical principles (mass balances, energy balances), empirical relations, implicit concepts, and external restrictions;

4. If the problem formulation is too large in scope: break it up into manageable parts or (b) simplify the objective function and model;

5. Apply a suitable optimization technique to the mathematical statement of the problem;

6. Check the answers, and examine the sensitivity of the result to changes in the coefficients in the problem and the assumptions;

II.5. Solving optimization problems with constraints using Excel

II.5.1. Presentation of Solver Excel

- The Solver is a simulation analysis tools (scenario analysis: a procedure that involves changing the values of cells to see how they affect the outcome of the formulas in the spreadsheet.

- The Solver allows you to find an optimal value (maximum or minimum) for a formula in a single cell, called the objective cell, based on constraints or limits applied to the values of other cells in the formula within a spreadsheet.

- The Solver uses a group of cells, called decision variables or simply variable cells, which are involved in the calculations of the objective and constraint cells' formulas.

Note: In Solver, the objective cell is called the "target cell," and the decision variable cells are called "variable cells."

II.5.2. Define and Solve a Problem

1. On the Data tab, in the Analysis group, click on Solver.

Note:

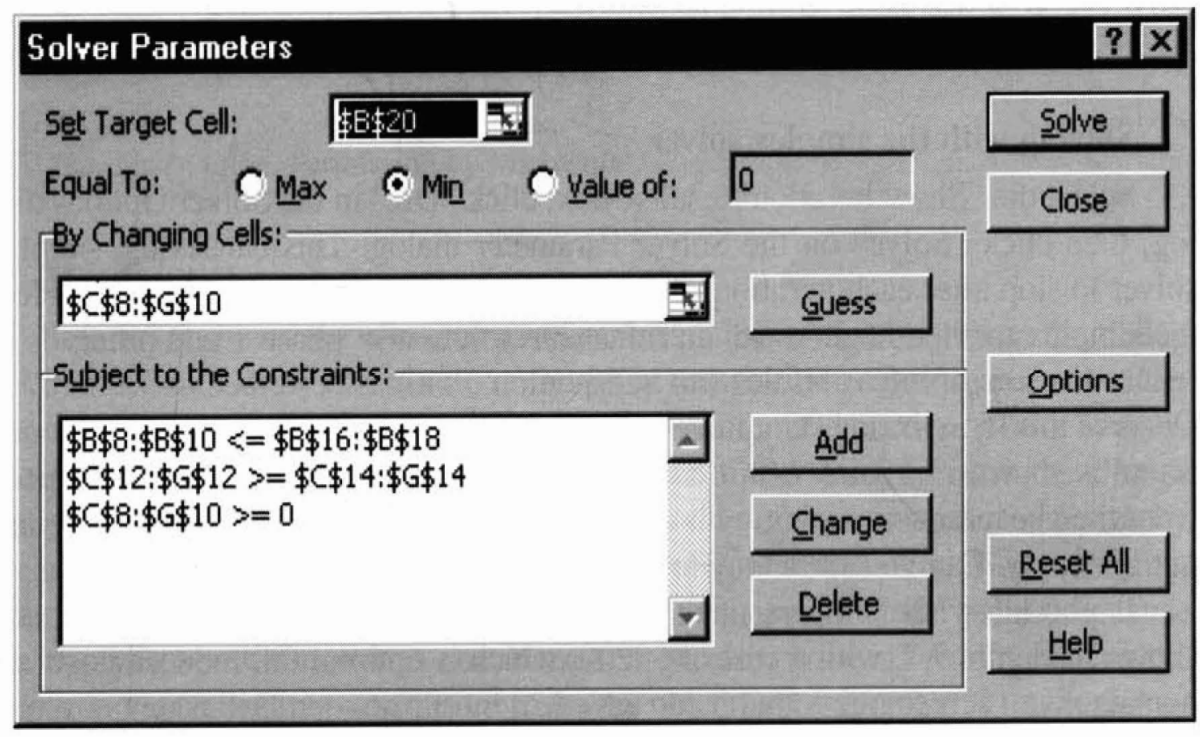
If the Solver command or the Analysis group is not available, you need to load the Solver add-in: Click the File tab, then Options, and go to the Add-Ins category. In the Available Add-ins box, check the Solver Add-In box, and then click OK.

2. In the Set Objective box, enter the reference of the objective cell. This cell must contain a formula.

3. Do one of the following:

- To make the objective cell value as large as possible, click Max.
- To make the objective cell value as small as possible, click Min.
- To set the objective cell to a specific value, click Value Of, then type the desired value in the box.

4. In the By Changing Variable Cells box, enter the name or reference for each decision variable cell range. Separate nonadjacent references with commas. Variable cells must be directly or indirectly related to the objective cell. You can specify up to 200 variable cells.



5. In the Subject to the Constraints box, enter the constraints as follows:

- In the Solver Parameters dialog box, click Add.
- In the Cell Reference box, enter the cell reference or name of the range of cells whose value you want to constrain.
- Click the relation (<=, =, >=) that you want to set between the referenced cell and the constraint. Enter a number, a cell reference, or a formula.
- Do one of the following:
  - To accept the constraint and add another, click Add.
  - To accept the constraint and return to the Solver Parameters dialog box, click OK.

- To modify or delete an existing constraint, follow these steps:
  - In the Solver Parameters dialog box, select the constraint you want to modify or delete.
  - Click Change to make your edits or Delete to remove it.

6. Click Solve, then do one of the following:
   - To keep the solution values in the worksheet, in the Solver Results dialog box, click Keep Solver Solution.
   - To restore the values that were in place before clicking Solve, click Restore Original Values.

Notes:

- To create a report based on your solution after Solver finds a solution, you can click a report type in the Reports box, then click OK.

- To save the decision variable cell values as a scenario you can review later, click Save Scenario in the Solver Results dialog box, then type a name for the scenario in the Scenario Name box.

II.5.3. Viewing Intermediate Solutions from Solver

- After defining a problem, click Options in the Solver Parameters dialog box.
- In the Options dialog box, check Show Iteration Results to display the values for each intermediate solution, then click OK.
- In the Solver Parameters dialog box, click Solve.
- In the Show Trial Solution dialog box, do one of the following:

   - To stop the solving process and display the Solver Results dialog box, click Stop.
   - To continue the solving process and display the next trial solution, click Continue.

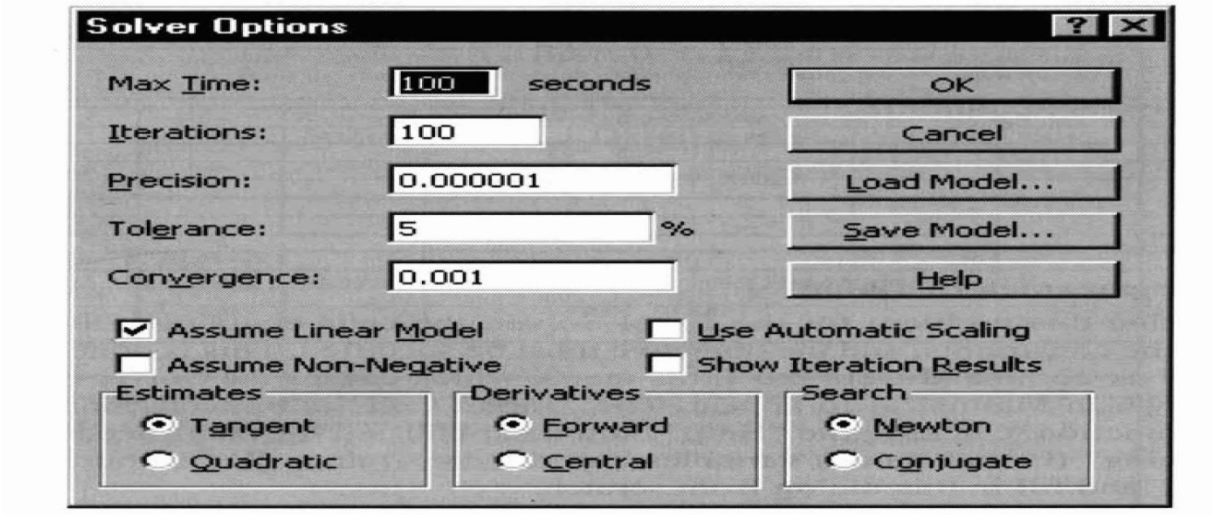
II.5.4. Solving Methods Used by Solver

You can choose any of the following three algorithms or solving methods in the Solver Parameters dialog box:

GRG Nonlinear: Designed for simple nonlinear problems.

Simplex LP: Designed for linear problems.

Evolutionary: Designed for complex problems.



II.6. Linear Programming (LP)

- It is a mathematical optimization technique used to find the best outcome (maximum or minimum) for a problem with linear constraints and a linear objective function.

- Linear programming is widely used in various fields such as economics, business, engineering, ...

II.6.1. Key Components of Linear Programming

Objective Function: A linear function of decision variables that needs to be maximized or minimized.

Constraints: A set of linear inequalities or equalities that define the feasible region for the decision variables.

Feasible Region: The set of all possible solutions that satisfy the constraints. It is often a convex polyhedron in the solution space defined by the constraints.

Optimal Solution: The solution that maximizes or minimizes the objective function within the feasible region. If the feasible region is bounded and the objective function is linear, the optimal solution will occur at one of the vertices (corner points) of the feasible region.

Example:

We want to schedule the production in two plants, A and B, each of which can manufacture two products: 1 and 2. How should the scheduling take place to maximize profits while meeting the market requirements based on the following data:

Plant	kg/day		DA/kg	
	1	2	1	2
A	MA1	MA2	SA1	SA2
B	MB1	MB2	SB1	SB2

How many days per year (365 days) should each plant operate processing each kind of material? Hints: Does the table contain the variables to be optimized? How do you use the information mathematically to formulate the optimization problem? What other factors must you consider?

Solution

How should we start to convert the words of the problem into mathematical statements? First, let us define the variables. There will be four of them **(t_{Al}, t_{A2}, t_{B1}, t_{B2})** designated as a set by the vector t, representing, respectively, the number of days per year each plant operates on each material as indicated by the subscripts. What is the objective function? We select the annual profit so that:

$$f(t) = t_{A1}M_{A1}S_{A1} + t_{A2}M_{A2}S_{A2} + t_{B1}M_{B1}S_{B1} + t_{B2}M_{B2}S_{B2} \qquad \textbf{(a)}$$

Next, do any equality constraints evolve from the problem statement or from implicit assumptions? If each plant runs 365 days per year, two equality constraints arise:

$$t_{A1} + t_{A2} = 365 \qquad \textbf{(b)}$$

$$t_{B1} + t_{B2} = 365 \qquad \textbf{(c)}$$

Finally, do any inequality constraints evolve from the problem statement or implicit assumptions? On first glance it may appear that there are none, but further thought indicates t must be nonnegative since negative values of t have no physical meaning:

$$t_{A1} \geq 0,\ t_{A2} \geq 0 \qquad \textbf{(d)}$$

$$t_{B1} \geq 0,\ t_{B2} \geq 0 \qquad \textbf{(e)}$$

To find the optimal t, we need to optimize (a) subject to constraints (b) to (e).

II.6.2. Standard Form of a Linear Programming Problem

An LP problem can always be written in the following form. Choose $X = (x_1, x_2, \ldots, x_n)$ to:

$$f = \sum_{j=1}^{n} c_j . x_j$$

$$\textbf{Subject to: } \sum_{j=1}^{n} a_{ij} x_j = b_i \ , \qquad \textbf{with } i = 1, 2, \ldots, m$$

$$l_j \leq x_j \leq u_j \ , \ j = 1, \ldots, n$$

Where:

c_j: are the n objective function coefficients,

a_{ij} and b_i: are parameters in the m linear equality constraints,

l_j and u_j are lower and upper bounds.

Note: all the constraints in Equation are equalities.

- It is necessary to place the problem in this form to solve it most easily (equations are easier to work with here than inequalities).
- If the original system is not of this form, it may easily be transformed by use of so-called "slack variables".
- If a given constraint is an inequality, for example:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

Then define a slack variable $x_{n+i} \geq 0$ such that:

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i$$

And the inequality becomes an equality.

- Similarly, if the inequality is:

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i$$

We write :

$$\sum_{j=1}^{n} a_{ij}x_j - x_{n+i} = b_i$$

- Note that the slacks must be nonnegative to guarantee that the inequalities are satisfied.

Example: Transform the following linear program into standard form:

$$Minimize: f = x_1 + x_2$$
$$Subject\ to:\ 2x_1 + 3x_2 \leq 6$$
$$x_1 + 7x_2 \geq 4$$
$$x_1 + x_2 = 3$$
$$x_1 \geq 0, \qquad x_2 \geq 0$$

Solution:

$$Minimize: f = x_1 + x_2$$
$$Subject\ to:\ 2x_1 + 3x_2 + x_3 = 6$$
$$x_1 + 7x_2 - x_4 = 4$$
$$x_1 + x_2 = 3$$
$$x_1 \geq 0, \qquad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0$$

With x_3, x_4: slack variables.

II.6.3. Solving Linear Programming Problems

1. Graphical method

For linear programming is a technique used to solve optimization problems with two decision variables (usually denoted x_1 and x_2). This method allows you to find the maximum or minimum value of an objective function within a feasible region defined by a set of linear inequalities (constraints).

Steps for the Graphical Method

1- Define the Problem:

- Formulate the objective function: Maximize or Minimize $Z = \boldsymbol{c_1}.\boldsymbol{x_1} + \boldsymbol{c_2}.\boldsymbol{x_2}$

Where c_1 and c_2 are constants.

- Define the constraints as a set of linear inequalities (e.g., $\boldsymbol{a_1}.\boldsymbol{x_1} + \boldsymbol{b_1}.\boldsymbol{x}_2 \leq \boldsymbol{d_1}$, where a_1, b_1, *and* d_1 are constants).

2- Graph the Constraints:

- Convert each inequality constraint into an equation by replacing the inequality sign with an equals sign (e.g., $\boldsymbol{a_1}.\boldsymbol{x_1} + \boldsymbol{b_1}.\boldsymbol{x}_2 = \boldsymbol{d_1}$).
- Plot each equation on a coordinate plane. Each line divides the plane into two half-planes.
- Determine the side of the line that satisfies the inequality by testing a point (often the origin, if it's not on the line).
- Shade the region that not satisfies each inequality constraint.

3- Identify the Feasible Region:

- The feasible region is the area where not shaded regions (representing each constraint) overlap.

- This region represents all possible values of x_1 and x_2 that satisfy the constraints.

- If there is no overlapping region, the problem has no feasible solution.

4- Locate the Corner Points:

- Identify the vertices (corner points) of the feasible region. These points are typically found at the intersections of the constraint lines.

- Calculate these intersection points by solving pairs of equations.

5- Evaluate the Objective Function at Each Corner Point:

- Substitute each corner point's coordinates into the objective function to determine the value of Z at each point.

- If maximizing, find the corner point with the highest Z;
- if minimizing, find the lowest.

6- Select the Optimal Solution:

The corner point with the optimal Z value is the solution to the problem, giving the values of x_1 *and* x_2 that maximize or minimize the objective function.

Advantages and Limitations

Advantages:

The graphical method is visually intuitive and provides a clear understanding of the feasible region and optimal solution.

Limitations:

- This method is only practical for problems with two variables (or exceptionally three, though difficult to visualize).

- For larger problems, more advanced methods like the simplex algorithm are used.

- This method is especially useful for teaching and understanding the fundamentals of linear programming.

Example:

$$Maximize: f = x_1 + 3x_2$$
$$Subject\ to: -x_1 + x_2 \leq 1$$
$$x_1 + x_2 \leq 2$$
$$x_1 \geq 0, \qquad x_2 \geq 0$$

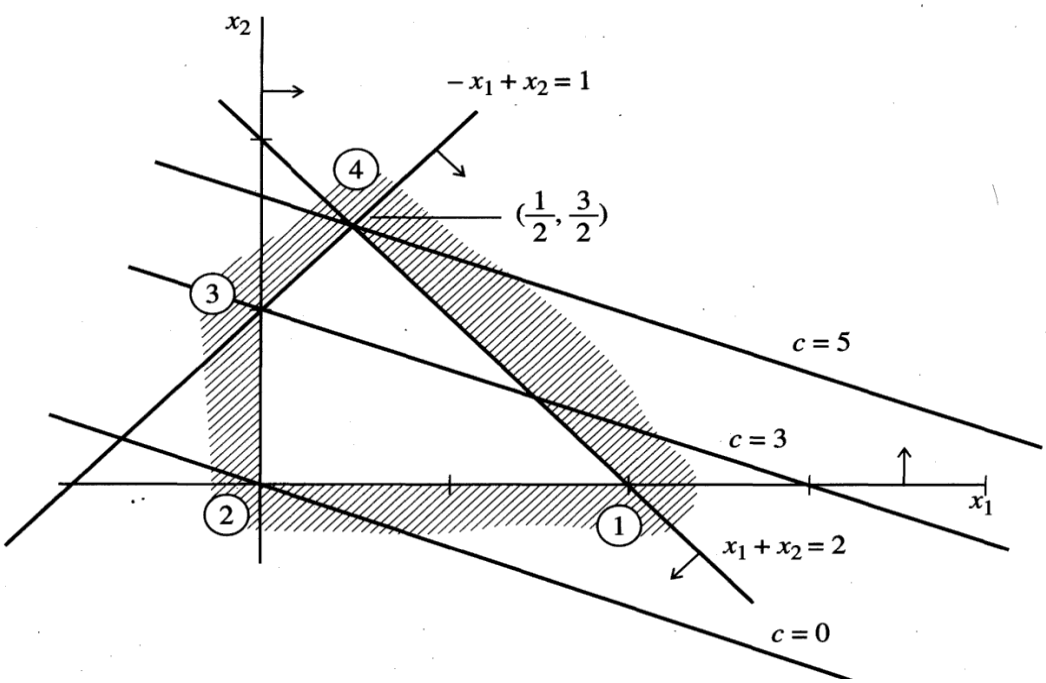
Solution

- The feasible region lies within the unshaded area defined by the intersections of the half spaces satisfying the linear inequalities.

- The numbered points are called extreme points, corner points, or vertices of this set. If the constraints are linear, only a finite number of vertices exist.

- Contours of constant value of the objective function f are defined by the linear Equation:

$$x_1 + 3x_2 = Constant = c$$



- As c varies, the contour is moved parallel to itself. The maximum value of *f* is the largest c for which the line has at least one point in common with the constraint set.

- This point occurs for ***c = 5***, and the optimal values of *x* are $x_1 = 0.5, x_2 = 1.5$.

- If the problem seeks to minimize f, the minimum is at the origin, which is again a vertex.

2. Simplex Method

The simplex method is a popular algorithm for solving linear programming (LP) problems, which involve maximizing or minimizing a linear objective function subject to a set of linear equality or inequality constraints. The method was developed by George Dantzig in 1947 and is foundational in optimization due to its efficiency and wide application.

Advantages of the Simplex Method

- It has an exponential time complexity in the worst case, the simplex method performs well on average and is highly efficient for practical problems.

- Each iteration provides a feasible solution, making it possible to understand the progression toward optimality.

Limitations

- The method can encounter issues like cycling in highly degenerate cases, although techniques like Bland's Rule help mitigate this.

- Despite being efficient in practice, the simplex method can, in rare cases, require an exponential number of steps to find an optimal solution.

a. Maximization by the Simplex Method

Key Steps in the Simplex Method

1. Formulating the Problem in Standard Form:

- Linear programming problems are usually converted into a standard form where the objective function is to be maximized, constraints are in equality form (using slack or surplus variables), and all variables are non-negative.

2. Setting up the Initial Table:

- The constraints and objective function are represented in a tabular format known as the simplex tableau. The table contains coefficients from the objective function and constraints, arranged to allow easy pivoting operations.

3. Iterative Improvement (Pivoting):

- The simplex method iteratively moves along the edges of the feasible region of the LP problem from one vertex (basic feasible solution) to another.

- In each iteration, the algorithm selects a non-basic variable to enter the basis (usually the one with the largest coefficient in the objective function row, indicating the steepest ascent or descent for maximization or minimization, respectively).

- A pivot operation is performed to maintain feasibility while improving the objective value.

4. Optimality and Termination:

- The algorithm terminates when there are positive coefficients in the objective row (for maximization problems), indicating that the optimal solution has been found. The values of the basic variables at this point represent the optimal solution to the LP problem.

- If a situation arises where the objective function can be improved indefinitely, it indicates an unbounded solution.

Example:

$$Maximize: Z = 3.x_1 + 2.x_2$$
$$Subject\ to:\ x_1 + x_2 \leq 4$$
$$x_1 \leq 2$$
$$x_2 \leq 3$$
$$x_1 \geq 0,\ x_2 \geq 0$$

Solution

Step 1: Convert to Standard Form

To use the simplex method, we need to convert inequalities into equalities by adding slack variables s_1, s_2, and S_3.

Rewriting the constraints:

$x_1 + x_2 + s_1 = 4$

$x_1 + s_2 = 2$

$x_2 + s_3 = 3$

Now, the objective function becomes:

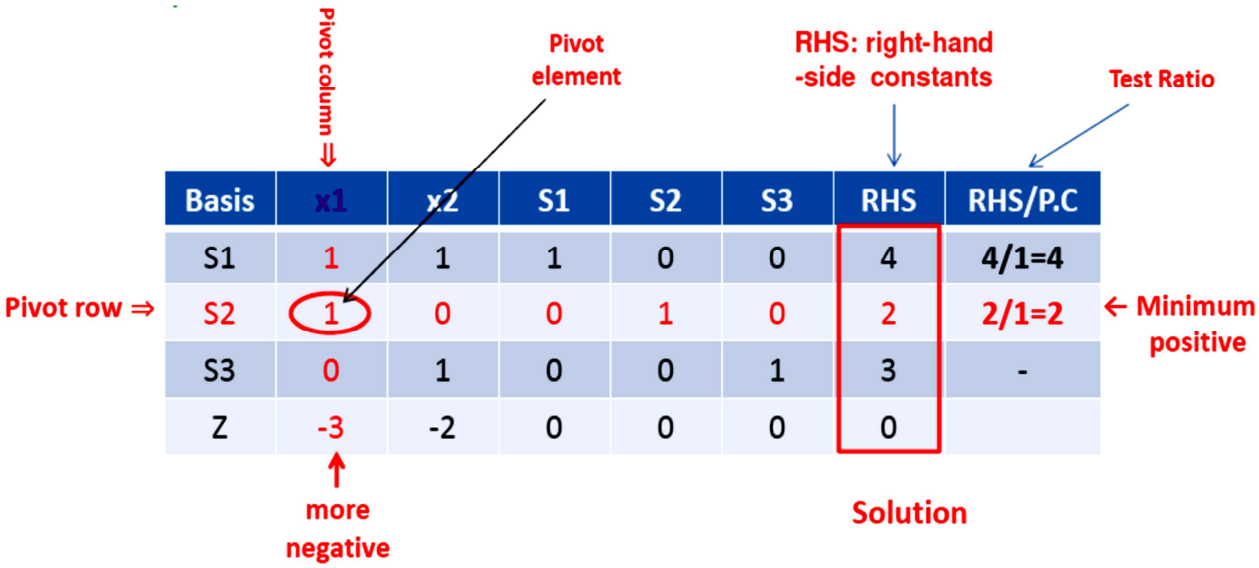
$$Z = 3.x_1 + 2.x_2 + 0.s_1 + 0.s_2 + 0.s_3$$

$$\Rightarrow Z - 3.x_1 + 2.x_2 + 0.s_1 + 0.s_2 + 0.s_3 = 0$$

Step 2: The initial simplex tableau:

- The non-basic variables are x_1 *and* x_2

- The basic variables are s_1, s_2, *and* s_3.



Basis	x1	x2	S1	S2	S3	RHS	RHS/P.C
S1	1	1	1	0	0	4	4/1=4
S2	1	0	0	1	0	2	2/1=2
S3	0	1	0	0	1	3	-
Z	-3	-2	0	0	0	0	

- The initial (or current) basic feasible solution is: $(x_1, x_2, s_1, s_2, s_3)=(0, 0, 4, 2, 3)$. This solution has an objective-function value **Z=0**.

- Since the coefficients of x_1 *and* x_2 (the non-basic variables) in that row are both negative, the current solution is not optimal.

Step 3: Identify the Pivot Column and Row

Pivot Column:

- Since the coefficient of x_1, namely (-3), is more negative than that of x_2, we will select x_1 as the entering variable (pivot column).

- We Choose the most negative coefficient in the objective row to enter the basis. So the pivot column is x_1.

Pivot Row:

-To determine the maximum possible increase in x_1, we conduct a ratio test.

- The ratio test will involve the coefficients in the pivot column and in the RHS column.

- Divide the RHS values by the pivot column values where positive to find the smallest ratio.

- Since the minimum ratio appears in R_2, the basic variable currently associated with that row, s_2 will be the leaving variable.

- We will refer to (S_2) as the pivot row

Pivot element:

- So, the pivot element is 1 in Row 2 (s_2) and Column $\mathbf{x_1}$.

- With s_2 leaving and x_1 entering, the new basis will be s_1, x_1, and s_3.

- We are now interested in constructing a new tableau that is targeted to assume the configuration specified below.

Step 4: Perform Row Operations (Second Tableau)

1. Pivot column (x_1): Make the pivot element 1 and adjust other elements in the column to zero.

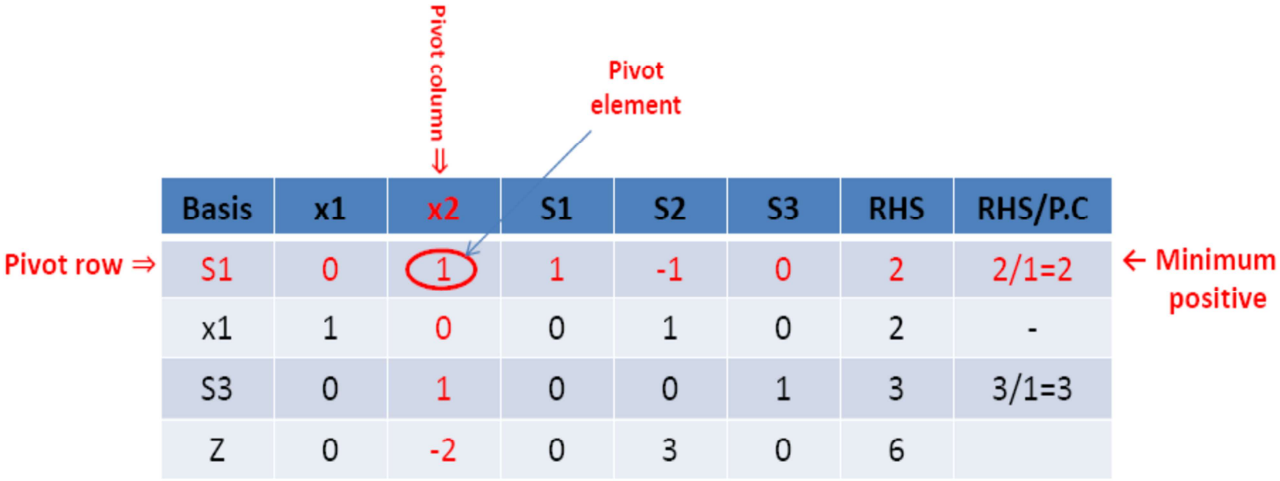
2. Pivot Row (Row 2): Divide all the row by pivot element to make the pivot element 1.

3. Column x2, S1, and S3: If the pivot row receives zeros, the columns remain unchanged.

Basis	x1	x2	S1	S2	S3	RHS
S1	0	1	1	?	0	?
x1	1	0	0	1	0	2
S3	0	1	0	?	1	?
Z	0	-2	0	?	0	?

4. The ?'s in the tableau represent blanks whose entries are to be determined, using the rectangle method with respect to the pivot element from the initial tableau:

? = a - (b . c / pivot)



Basis	x1	x2	S1	S2	S3	RHS	RHS/P.C
S1	0	1	1	-1	0	2	2/1=2
x1	1	0	0	1	0	2	-
S3	0	1	0	0	1	3	3/1=3
Z	0	-2	0	3	0	6	

Step 5: Check for Optimality

Since there is still a negative coefficient (–2) in the objective row, we continue with another pivot operation, now focusing on x_2.

1. Pivot Column: Choose x_2 (with –2 in Row z).

2. Pivot Row: Calculate ratios for RHS divided by the x_2 values.

So, the pivot element is 1 in Row 1 and Column x_2.

Step 6: Perform Final Row Operations (Third Tableau)

1. Pivot column (x_2): Make the pivot element 1 and adjust other elements in the column to zero.

2. Pivot Row (Row 1): Divide all the row by pivot element to make the pivot element 1.

3. Column x_1, and s_3: If the pivot row receives zeros, the columns remain unchanged.

Basis	x1	x2	S1	S2	S3	RHS
x2	0	1	1	-1	0	2
x1	1	0	?	?	0	?
S3	0	0	?	?	1	?
Z	0	0	?	?	0	?

4. The ?'s are determined using the rectangle method with respect to the pivot element from the second tableau: **? = a - (b . c / pivot)**

5. So the third Tableau is :

Basis	x1	x2	S1	S2	S3	RHS
x2	0	1	1	-1	0	2
x1	1	0	0	1	0	2
S3	0	0	-1	1	1	1
Z	0	0	2	1	0	10

Solution

Maximum solution:

Since there is non negative coefficient in the objective row, we stop calculation, the optimal tableau show: $x_1=2$, $x_2=2$, with an optimal value of ***Max z=10***.

b. Minimization by the Simplex Method

- The procedure to solve these problems involves solving an associated problem called the dual problem.

- To every minimization problem there corresponds a dual problem. The solution of the dual problem is used to find the solution of the original problem.

- The dual problem is a maximization problem, which we learned to solve in the last section.

- The objective function of the minimization problem reaches its minimum if and only if the objective function of its dual reaches its maximum. And when they do, they are equal.

- We first solve the dual problem by the simplex method.

- From the final simplex tableau, we then extract the solution to the original minimization problem.

Key Steps in the Simplex Method for minimization LP

1. Set up the problem.

2. Write a matrix whose rows represent each constraint with the objective function as its bottom row.

3. Write the transpose of this matrix by interchanging the rows and columns.

4. Now write the dual problem associated with the transpose.

5. Solve the dual problem by the simplex method for maximization.

6. The optimal solution is found in the bottom row of the final matrix in the columns corresponding to the slack variables, and the minimum value of the objective function is the same as the maximum value of the dual.

Example

$$Minimize: Z = 12.x_1 + 16.x_2$$
$$Subject\ to:\ x_1 + 2x_2 \geq 40$$
$$x_1 + x_2 \geq 30$$
$$x_1 \geq 0,\ x_2 \geq 0$$

Solution

1. Convert the minimization problem into its dual.

- To achieve our goal, we first express our problem as the following matrix.

1	2	40
1	1	30
12	16	0

- Observe that this table looks like an initial simplex tableau without the slack variables. Next, we write a matrix whose columns are the rows of this matrix, and the rows are the columns. Such a matrix is called a transpose of the original matrix. We get:

1	1	12
2	1	16
40	30	0

- The following maximization problem associated with the above matrix is called its dual:

$$Maximize: Z = 40.y_1 + 30.y_2$$
$$Subject\ to:\ y_1 + y_2 \leq 12$$
$$2y_1 + y_2 \leq 16$$
$$y_1 \geq 0,\ y_2 \geq 0$$

- Note that we have chosen the variables as y's, instead of x's, to distinguish the two problems.

2. Solving its dual using the simplex method

- Convert to Standard Form

To use the simplex method, we need to convert inequalities into equalities by adding slack variables: Rewriting the constraints: slack variables are x_1 and x_2:

$y_1 + y_2 + x_1 = 12$

$2y_1 + y_2 + x_2 = 16$

Now, the objective function becomes: $Z - 40.y1 - 30.y2 - 0.x1 - 0.x2 = 0$

- The first simplex tableau reads as follows:

Basis	y1	Y2	x1	x2	RHS
x1	1	1	1	0	12
x2	2	1	0	1	16
Z	-40	-30	0	0	0

- After using different steps for maximization problem, the final simplex tableau reads as follows:

Basis	y1	Y2	x1	x2	RHS
y2	1	1	1	0	8
y1	2	1	0	1	4
Z	0	0	20	10	400

- We restate the solution as follows: The minimization problem has a **minimum value of 400** at the corner point **(x_1=20, x_2=10)**

Références Bibliographiques

References

Thomas F. Edgar, David M. Himmelblau, Leon S. Lasdon. (2001). OPTIMIZATION OF CHEMICAL PROCESSES, 2nd edition , Published by McGraw-Hill Chemical Engineering Series.

Edwin K. P. Chong and Stanislaw H. Zak. (2001). AN INTRODUCTION TO OPTIMIZATION. 2nd edition, Published by John Wiley and sons.

Ismail Tosun. (2002). MODELLING IN TRANSPORT PHENOMENA: A CONCEPTUAL APPROACH. Elsevier Science Ed.

Huifu Xu. MATH3016: OPTIMIZATION. School of Mathematics, University of Southampton.

Michael T. (2019). NUMERICAL ANALYSIS: OPTIMIZATION. Heath Department of Computer Science University of Illinois at Urbana-Champaign.

Charles L. Byrne. A FIRST COURSE IN OPTIMIZATION. Department of Mathematical Sciences, University of Massachusetts Lowell.

David Vidal, Bruno Blais, François Bertrand. MODELISATION NUMERIQUE EN GENIE CHIMIQUE. Polytechnique Montreal.

Lætitia CESARI, Alain CHAMAYOU, Patrick COGNET, Marie DEBACQ, Éric SCHAER et Stéphane VITU. (2022). GENIE DE LA REACTION CHIMIQUE: LES REACTEURS HOMOGENES. Version 1.3, Cnam Paris, France.

Meche Abdelkrim. (2021/2022). MANUEL DE COURS 'OPTIMISATION'. Département d'Automatique, USTO-MB Oran.

Frédéric de Gournay & Aude Rondepierr. INTRODUCTION A L'OPTIMISATION NUMERIQUE. Département STPI, Toulouse.

Sari Triqui Lamia. (2016). Polycopié du cours " RECHERCHE OPERATIONNELLE ". Faculté de Technologie, Université Aboubakr Belkaïd Tlemcen.

Gilles Gasso. (2016). OPTIMISATION CONTINUE SANS CONTRAINTES: METHODES DE DESCENTE. INSA Rouen - Département ASI, Laboratoire LITIS.

BOUCHERIT Septi. (2017-2018). COURS OPTIMISATION. Département de Génie Mécanique, Université du 8 mai 1945 Guelma.