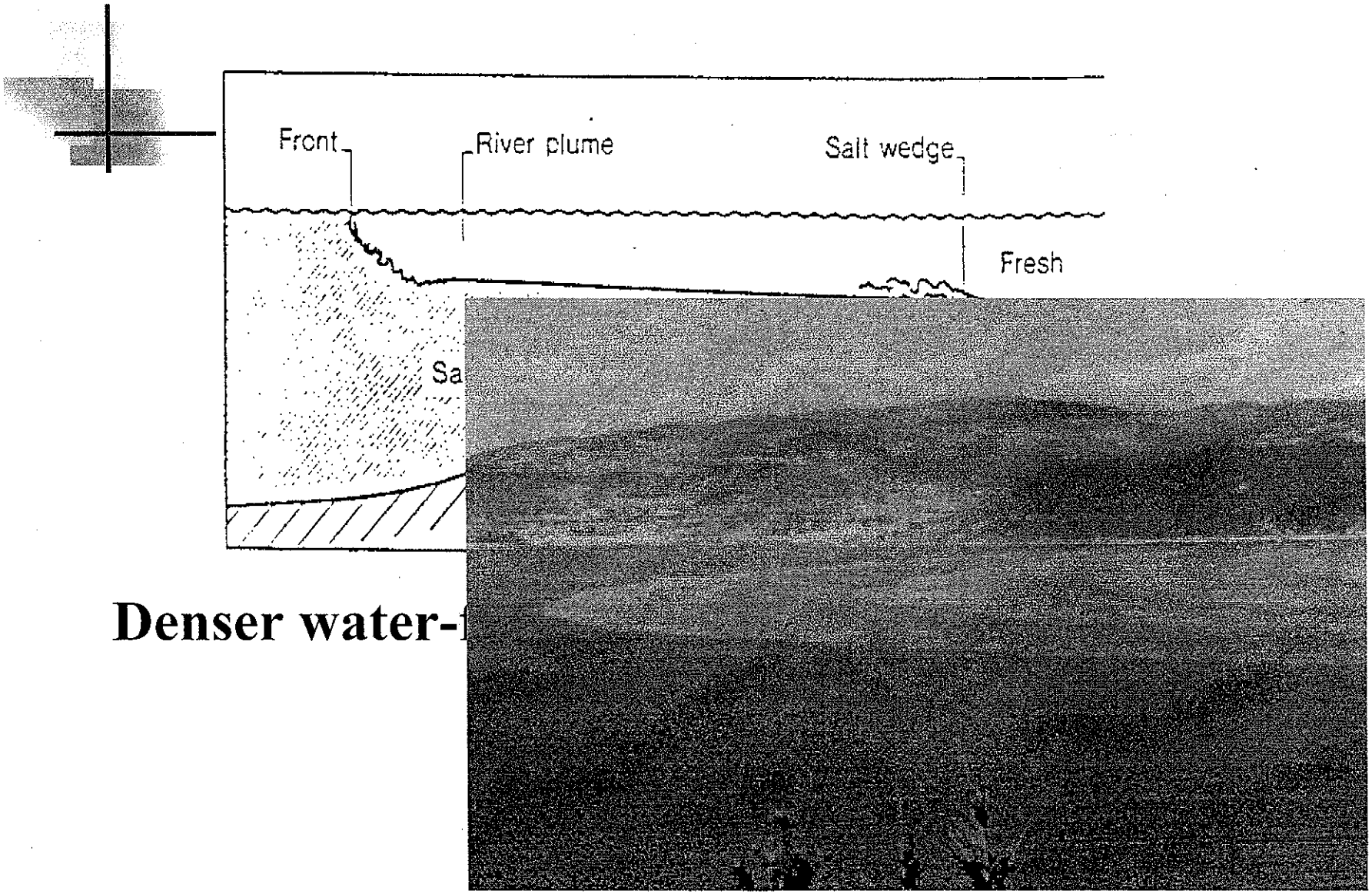
Estuaries: density currents



Denser water-

HAND OUT 2: Importance of water resources (Chapter 1 of our syllabus)
Source: Mays, L. (2006). *"Water resources engineering"* John Wiley and Sons

Chapter 1

Introduction

1.1 BACKGROUND

Water resources engineering (and management) as defined for the purposes of this book includes engineering for both *water supply management* and *water excess management* (see Figure 1.1.1). This book does not cover the *water quality management (or environmental restoration)* aspect of water resources engineering. The two major processes that are engineered are the *hydrologic processes* and the *hydraulic processes*. The common threads that relate to the explanation of the hydrologic and hydraulic processes are the fundamentals of fluid mechanics. The hydraulic processes include three types of flow: pipe (pressurized) flow, open-channel flow, and groundwater flow.

The broad topic of *water resources* includes areas of study in the biological sciences, engineering, physical sciences, and social sciences, as illustrated in Figure 1.1.1. The areas in biological sciences range from ecology to zoology, those in the physical sciences range from chemistry to meteorology to physics, and those in the social sciences range from economics to sociology. Water resources engineering as used in this book focuses on the engineering aspects of hydrology and hydraulics for water supply management and water excess management.

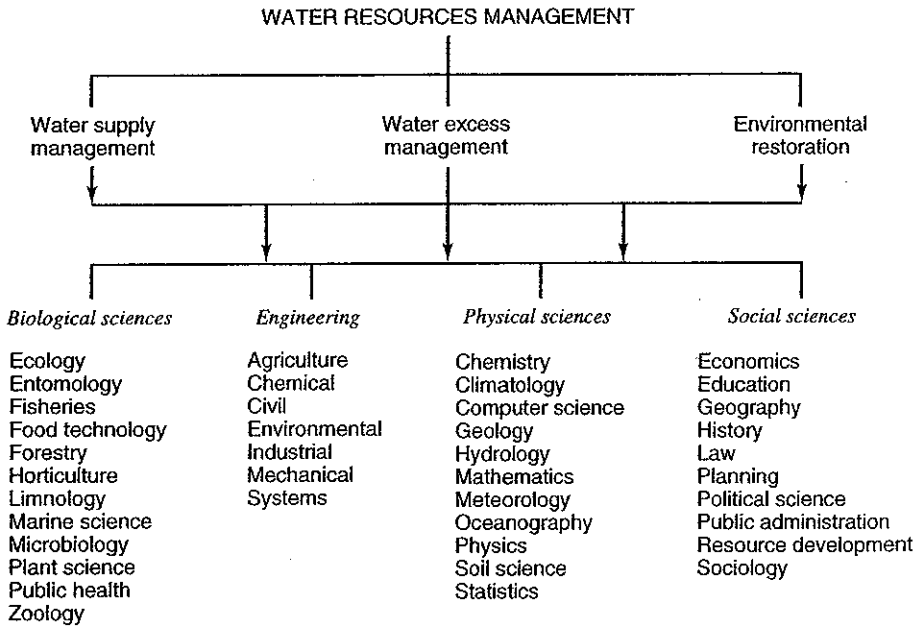


Figure 1.1.1 Ingredients of water resources management (from Mays (1996)).

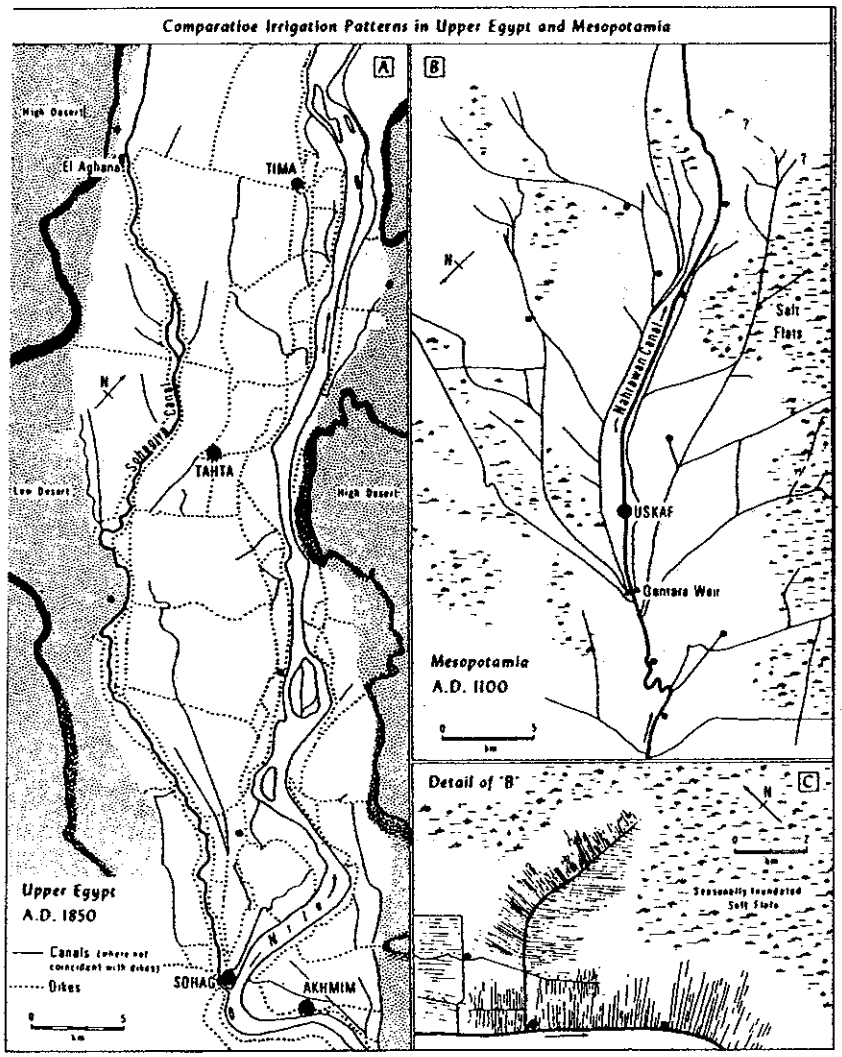


Figure 1.1.2 Comparative irrigation networks in Upper Egypt and Mesopotamia. (*a*) Example of linear, basin irrigation in Sohag province, ca. AD 1850; (*b*) Example of radial canalization system in the lower Nasharawan region southeast of Baghdad, Abbasid (A.D. 883–1150). (Modified from R. M. Adams (1965), Fig. 9. Same scale as Egyptian counterpart). (*c*) Detail of field canal layout in *b*. (Simplified from R. M. Adams (1965), Fig. 10. Figure as presented in Butzer (1976)).

Water resources engineering not only includes the analysis and synthesis of various water problems through the use of the many analytical tools in hydrologic engineering and hydraulic engineering but also extends to the design aspects.

Water resources engineering has evolved over the past 9,000 to 10,000 years as humans have developed the knowledge and techniques for building hydraulic structures to convey and store water. Early examples include irrigation networks built by the Egyptians and Mesopotamians (see Figure 1.1.2) and by the Hohokam in North America (see Figure 1.1.3). The world's oldest large dam was the Sadd-el-kafara dam built in Egypt between 2950 and 2690 B.C. The oldest known pressurized water distribution (approximately 2000 B.C.) was in the ancient city of Knossos on Crete (see Mays, 1999, 2000, for further details). There are many examples of ancient water systems throughout the world.

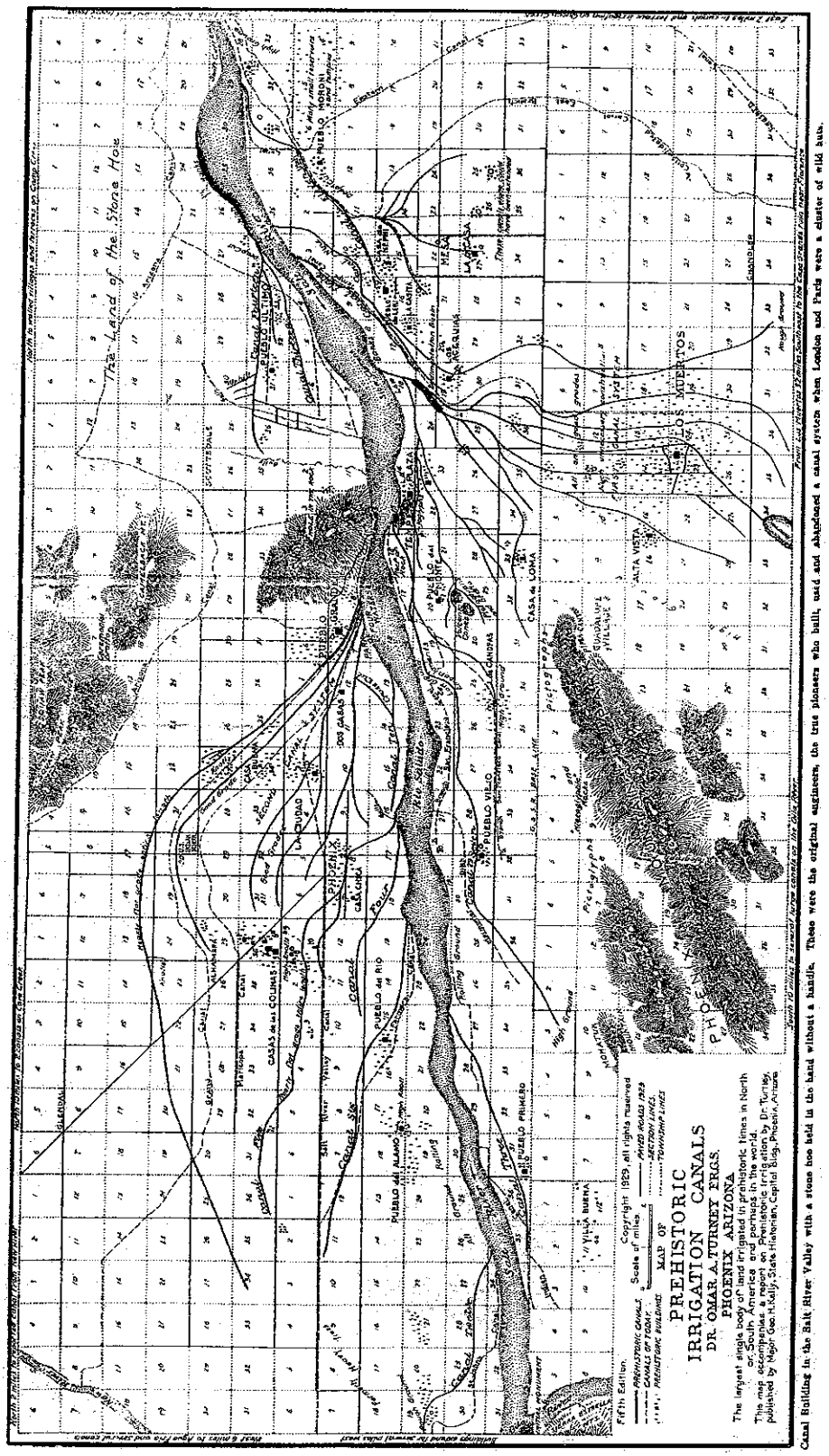


Figure 1.1.3 Canal building in the Salt River Valley with a stone hoe held in the hand without a handle. These were the original engineers, the true pioneers who built, used, and abandoned a canal system when London and Paris were clusters of wild huts (from Turney (1922)). (Courtesy of Salt River Project, Phoenix, Arizona.)

1.2 THE WORLD'S FRESHWATER RESOURCES

Among today's most acute and complex problems are water problems related to the rational use and protection of water resources (see Gleick, 1993). Associated with water problems is the need to supply humankind with adequate clean freshwater. Data collected on global water resources by Soviet scientists are listed in Table 1.2.1. These obviously are only approximations and should not be considered as accurate (Shiklomanov, 1993). Table 1.2.2 presents the dynamics of actual water availability in different regions of the world. Table 1.2.3 presents the dynamics of water use in the world by human activity. Table 1.2.4 presents the annual runoff and water consumption by continents and by physiographic and economic regions of the world.

Table 1.2.1 Water Reserves on the Earth

	Distribution area (10^3 km^2)	Volume (10^3 km^3)	Layer (m)	Percentage of global reserves	
				Of total water	Of freshwater
World ocean	361,300	1,338,000	3,700	96.5	—
Groundwater	134,800	23,400	174	1.7	—
Freshwater		10,530	78	0.76	30.1
Soil moisture		16.5	0.2	0.001	0.05
Glaciers and permanent snow cover	16,227	24,064	1,463	1.74	68.7
Antarctic	13,980	21,600	1,546	1.56	61.7
Greenland	1,802	2,340	1,298	0.17	6.68
Arctic islands	226	83.5	369	0.006	0.24
Mountainous regions	224	40.6	181	0.003	0.12
Ground ice/permafrost	21,000	300	14	0.022	0.86
Water reserves in lakes	2,058.7	176.4	85.7	0.013	—
Fresh	1,236.4	91	73.6	0.007	0.26
Saline	822.3	85.4	103.8	0.006	—
Swamp water	2,682.6	11.47	4.28	0.0008	0.03
River flows	148,800	2.12	0.014	0.0002	0.006
Biological water	510,000	1.12	0.002	0.0001	0.003
Atmospheric water	510,000	12.9	0.025	0.001	0.04
Total water reserves	510,000	1,385,984	2,718	100	—
Total freshwater reserves	148,800	35,029	235	2.53	100

Source: Shiklomanov (1993).

Table 1.2.2 Dynamics of Actual Water Availability in Different Regions of the World

Continent and region	Area (10^6 km^2)	Actual water availability (10^3 m^3 per year per capita)				
		1950	1960	1970	1980	2000
Europe	10.28	5.9	5.4	4.9	4.6	4.1
North	1.32	39.2	36.5	33.9	32.7	30.9
Central	1.86	3.0	2.8	2.6	2.4	2.3
South	1.76	3.8	3.5	3.1	2.8	2.5
European USSR (North)	1.82	33.8	29.2	26.3	24.1	20.9
European USSR (South)	3.52	4.4	4	3.6	3.2	2.4
North America	24.16	37.2	30.2	25.2	21.3	17.5
Canada and Alaska	13.67	384	294	246	219	189
United States	7.83	10.6	8.8	7.6	6.8	5.6
Central America	2.67	22.7	17.2	12.5	9.4	7.1

Table 1.2.2 Dynamics of Actual Water Availability in Different Regions of the World *(continued)*

Continent and region	Area (10^6 km^2)	Actual water availability (10^3 m^3 per year per capita)				
		1950	1960	1970	1980	2000
Africa	30.10	20.6	16.5	12.7	9.4	5.1
North	8.78	2.3	1.6	1.1	0.69	0.21
South	5.11	12.2	10.3	7.6	5.7	3.0
East	5.17	15.0	12	9.2	6.9	3.7
West	6.96	20.5	16.2	12.4	9.2	4.9
Central	4.08	92.7	79.5	59.1	46.0	25.4
Asia	44.56	9.6	7.9	6.1	5.1	3.3
North China and Mongolia	9.14	3.8	3.0	2.3	1.9	1.2
South	4.49	4.1	3.4	2.5	2.1	1.1
West	6.82	6.3	4.2	3.3	2.3	1.3
South-east	7.17	13.2	11.1	8.6	7.1	4.9
Central Asia and Kazakhstan	2.43	7.5	5.5	3.3	2.0	0.7
Siberia and Far East	14.32	124	112	102	96.2	95.3
Trans-Caucasus	0.19	8.8	6.9	5.4	4.5	3.0
South America	17.85	105	80.2	61.7	48.8	28.3
North	2.55	179	128	94.8	72.9	37.4
Brazil	8.51	115	86	64.5	50.3	32.2
West	2.33	97.9	77.1	58.6	45.8	25.7
Central	4.46	34	27	23.9	20.5	10.4
Australia and Oceania	8.59	112	91.3	74.6	64.0	50.0
Australia	7.62	35.7	28.4	23	19.8	15.0
Oceania	1.34	161	132	108	92.4	73.5

Source: Shiklomanov (1993).

Table 1.2.3 Dynamics of Water Use in the World by Human Activity

Water users[a]	1900 (km^3 per year)	1940 (km^3 per year)	1950 (km^3 per year)	1960 (km^3 per year)	1970 (km^3 per year)	1975 (km^3 per year)	1980		1990[b]		2000[b]	
							(km^3 per year)	(%)	(km^3 per year)	(%)	(km^3 per year)	(%)
Agriculture												
Withdrawal	525	893	1,130	1,550	1,850	2,050	2,290	69.0	2,680	64.9	3,250	62.6
Consumption	409	679	859	1,180	1,400	1,570	1,730	88.7	2,050	86.9	2,500	86.2
Industry												
Withdrawal	37.2	124	178	330	540	612	710	21.4	973	23.6	1,280	24.7
Consumption	3.5	9.7	14.5	24.9	38.0	47.2	61.9	3.2	88.5	3.8	117	4.0
Municipal supply												
Withdrawal	16.1	36.3	52.0	82.0	130	161	200	6.0	300	7.3	441	8.5
Consumption	4.0	9.0	14	20.3	29.2	34.3	41.1	2.1	52.4	2.2	64.5	2.2
Reservoirs												
Withdrawal	0.3	3.7	6.5	23.0	66.0	103	120	3.6	170	4.1	220	4.2
Consumption	0.3	3.7	6.5	23.0	66.0	103	120	6.2	170	7.2	220	7.6
Total (rounded off)												
Withdrawal	579	1,060	1,360	1,990	2,590	2,930	3,320	100	4,130	100	5,190	100
Consumption	417	701	894	1,250	1,540	1,760	1,950	100	2,360	100	2,900	100

[a] Total water withdrawal is shown in the first line of each category, consumptive use (irretrievable water loss) is shown in the second line.
[b] Estimated.

Source: Shiklomanov (1993).

Table 1.2.4 Annual Runoff and Water Consumption by Continents and by Physiographic and Economic Regions of the World

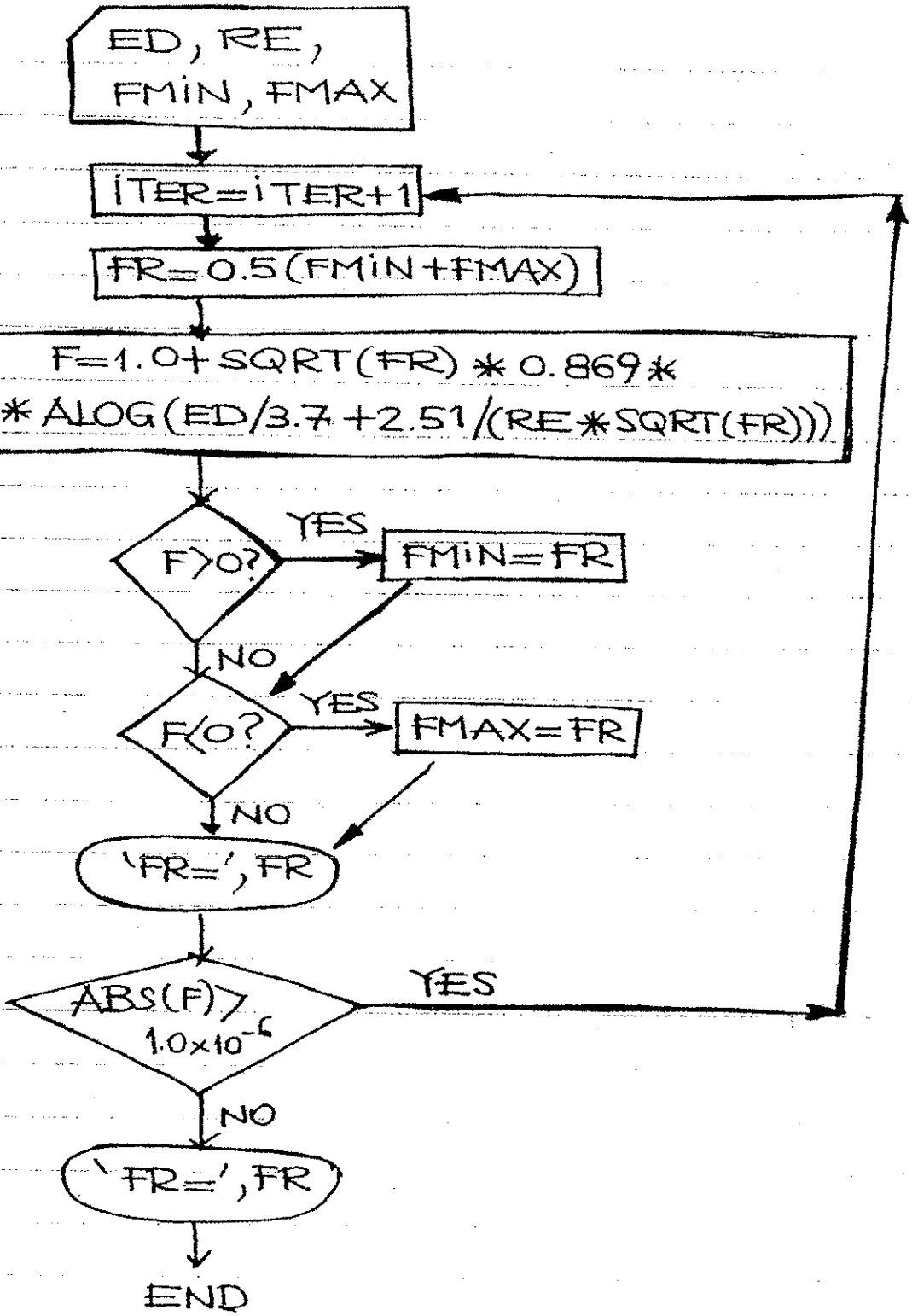
Continent and region	Mean annual runoff (mm)	Mean annual runoff (km^3 per year)	Aridity index (R/LP)	Water consumption (km^3 per year) 1980 Total	1980 Irretrievable	1990 Total	1990 Irretrievable	2000 Total	2000 Irretrievable
Europe	310	3,210	—	435	127	555	178	673	222
North	480	737	0.6	9.9	1.6	12	2.0	13	2.3
Central	380	705	0.7	141	22	176	28	205	33
South	320	564	1.4	132	51	184	64	226	73
European USSR (North)	330	601	0.7	18	2.1	24	3.4	29	5.2
European USSR (South)	150	525	1.5	134	50	159	81	200	108
North America	340	8,200	—	663	224	724	255	796	302
Canada and Alaska	390	5,300	0.8	41	8	57	11	97	15
United States	220	1,700	1.5	527	155	546	171	531	194
Central America	450	1,200	1.2	95	61	120	73	168	93
Africa	150	4,570	—	168	129	232	165	317	211
North	17	154	8.1	100	79	125	97	150	112
South	68	349	2.5	23	16	36	20	63	34
East	160	809	2.2	23	18	32	23	45	28
West	190	1,350	2.5	19	14	33	23	51	34
Central	470	1,909	0.8	2.8	1.3	4.8	2.1	8.4	3.4
Asia	330	14,410	—	1,910	1,380	2,440	1,660	3,140	2,020
North China and Mongolia	160	1,470	2.2	395	270	527	314	677	360
South	490	2,200	1.3	668	518	857	638	1,200	865
West	72	490	2.7	192	147	220	165	262	190
South-east	1,090	6,650	0.7	461	337	609	399	741	435
Central Asia and Kazakhstan	70	170	3.1	135	87	157	109	174	128
Siberia and Far East	230	3,350	0.9	34	11	40	17	49	25
Trans-Caucasus	410	77	1.2	24	14	26	18	33	21
South America	660	11,760	—	111	71	150	86	216	116
Northern area	1,230	3,126	0.6	15	11	23	16	33	20
Brazil	720	6,148	0.7	23	10	33	14	48	21
West	740	1,714	1.3	40	30	45	32	64	44
Central	170	812	2.0	33	20	48	24	70	31
Australia and Oceania	270	2,390		29	15	38	17	47	22
Australia	39	301	4.0	27	13	34	16	42	20
Oceania	1,560	2,090	0.6	2.4	1.5	3.3	1.8	4.5	2.3
Land area (rounded off)	—	44,500	—	3,320	1,450	4,130	2,360	5,190	2,900

Source: Shiklomanov (1993).

1.3 WATER USE IN THE UNITED STATES

Dziegielewski et al. (1996) define *water use* from a hydrologic perspective as all water flows that are a result of human intervention in the hydrologic cycle. The National Water Use Information Program (NWUI Program), conducted by the United States Geological Survey (USGS), used this perspective on water use in establishing a national system of water-use accounting. This accounting system distinguishes the following water-use flows: (1) water withdrawals for off-stream purposes, (2) water deliveries at point of use or quantities released after use, (3) consumptive use, (4) conveyance loss, (5) reclaimed wastewater, (6) return flow, and (7) in-stream flow (Solley et al., 1993). The relationships among these human-made flows at various points of measurement are illustrated in Figure 1.3.1. Figure 1.3.2 illustrates the estimated water use by tracking the sources,

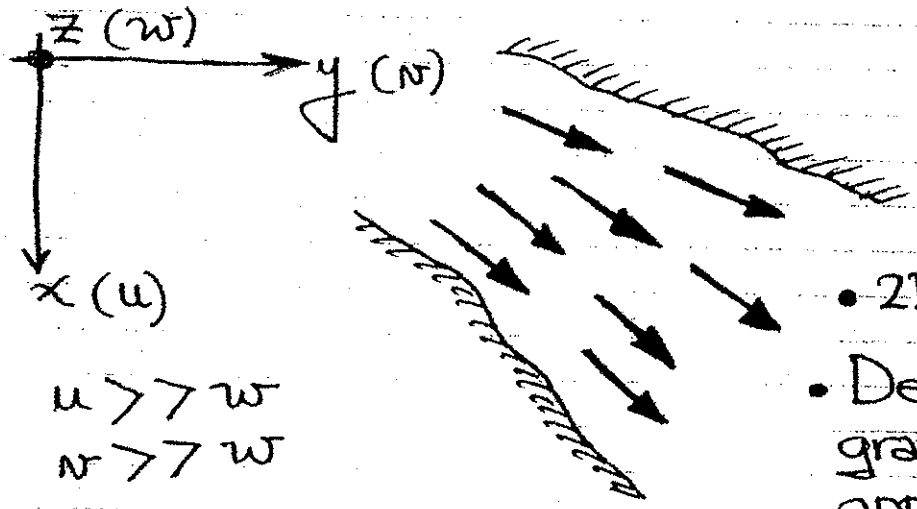
HAND OUT 3: Algorithm for the Computer Problem 1


ED, RE,
FMIN, FMAX
ITER=ITER+1
FR=0.5(FMIN+FMAX)
F=1.0+SQRT(FR)*0.869*
*ALOG(ED/3.7+2.51/(RE*SQRT(FR)))
F>0?
YES
FMIN=FR
NO
F<0?
YES
FMAX=FR
NO
'FR=', FR
ABS(F)
1.0x10^-6
YES
NO
'FR=', FR
END

HAND OUT 4: Classification of modeling approximations (Chapter 1 of our syllabus)

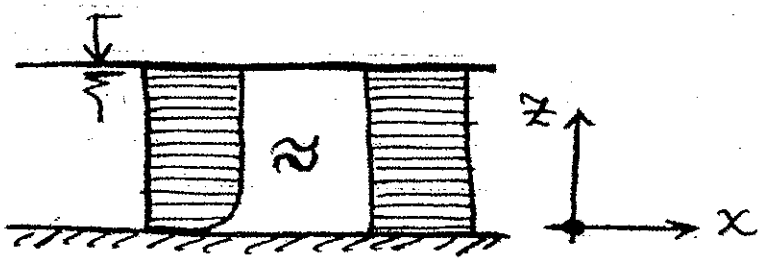
Examples of modeling:

1) ESTUARIES:

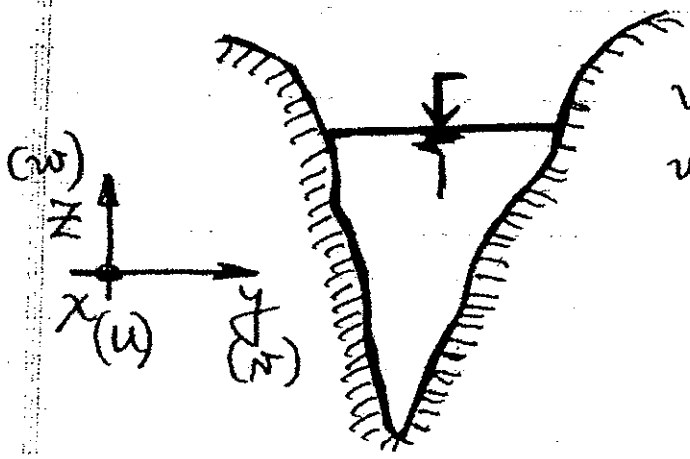


$u >> w$
$v >> w$

- 2D approx.
- Depth-integrated approaches



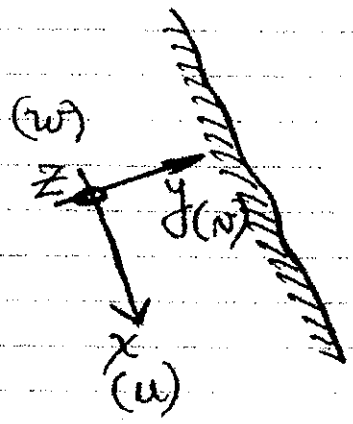
2) FJORDS:

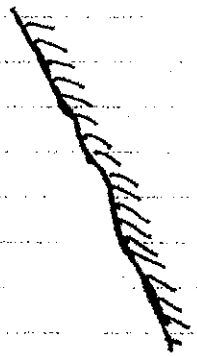


$w >> v$
$u >> v$

- 2D approx.
- Transversally integrated

3) RIVERS:



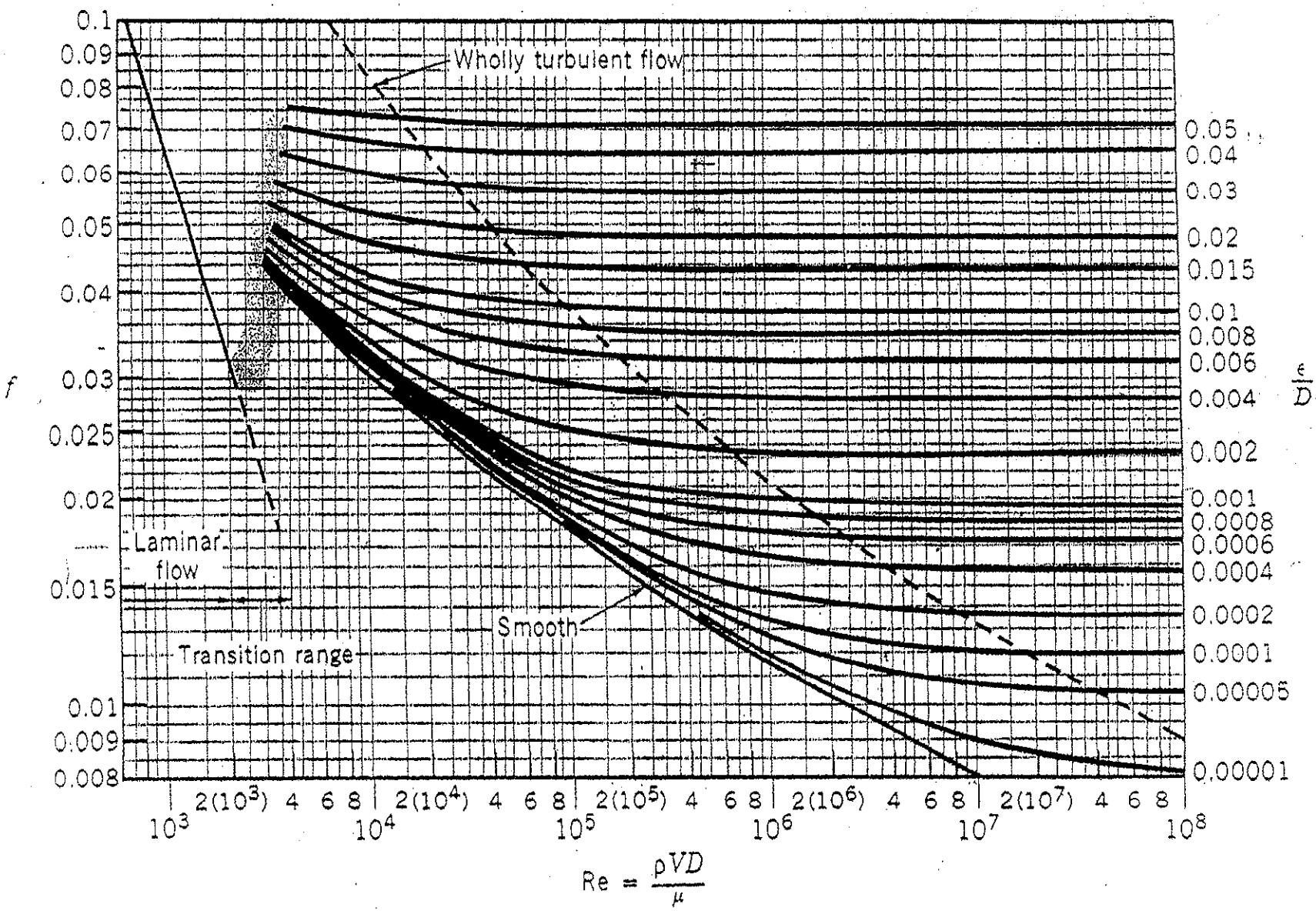


u ⊥ N

u ⊥ w

- 1D approx

HAND OUT 5: The Moody diagram (Chapter 1 of our syllabus)



■ **FIGURE 8.23** Friction factor as a function of Reynolds number and relative roughness for round pipes—the Moody chart (Data from Ref. 7 with permission).

HAND OUT 6: Methods for obtaining roots of algebraic equations (Chapter 2 of our syllabus). Source: Alexandrou, A. (2001). *"Principles of fluid mechanics."* Prentice Hall

and ideas are developed in sufficient depth so that the techniques can be readily used in practical problems.

13.1 Algebraic Equations

As encountered in several cases, mathematical models often reduce to single algebraic equations whose solution must be obtained numerically by iteration. Recall, for instance, from Chapter 8 that the friction factor could be obtained by using the Colebrook formula

$$\frac{1}{\sqrt{f}} = -0.869 \ln\left(\frac{e/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right). \tag{13.1}$$

Since f appears in both sides of the equation, the solution must be obtained numerically.

Similarly, in Prandtl-Meyer expansion, when the flow turns through a total angle ν, the resulting Mach number M is a complicated function given by

$$\nu = \sqrt{\frac{\gamma+1}{\gamma-1}}\tan^{-1}\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)} - \tan^{-1}\sqrt{M^2-1}. \tag{13.2}$$

The solution to this equation must also be obtained numerically.

Another example is the case of oblique shocks in which the geometry of the shock as a function of the incoming M is given by

$$\beta = \theta - \tan^{-1}\left[\frac{1}{\sin\theta\cos\theta}\left(\frac{\gamma-1}{\gamma+1}\sin^2\theta + \frac{2}{\gamma+1}\frac{1}{M_1^2}\right)\right]. \tag{13.3}$$

Given the complexity of the expression, the angle of the shock θ as a function of the Mach number M and deflection angle β must be found numerically.

13.1.1 Root of Equations

Consider a general algebraic equation of the form

$$y = f(x).$$

For most of such equations, the final objective is to find the root of the equation—that is, the value (or values) of the unknown x_i that satisfies

$$F = y - f(x_i) = 0. \tag{13.4}$$

The problem is shown schematically in Figure 13.3. As shown in the figure, depending on the order of F, the function F can have multiple solutions. An obvious choice to find the roots of Equation (13.4) is to start guessing values of x_i until we find those that satisfy $F(x_i) = 0$. However, this approach can be inefficient and lengthy. Fortunately, the search for the roots can be accelerated by using a number of numerical procedures. These methods can also be implemented easily in computer form. We review two such methods: (a) the *bisection method* and (b) the *Newton-Raphson method.*

Bisection Method

The bisection method formalizes the search for the root that lies in the range of (x_{min}, x_{max}), where x_{min}, and x_{max} are initial limits set by the user. During the iteration procedure,

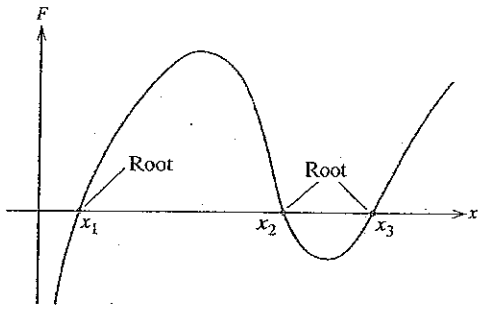


FIGURE 13.3 Multiple roots of a function.

the estimate for the root x_g is assumed to be the midpoint of the current range $x_g = \frac{x_{min} + x_{max}}{2}$, while the limits x_{min} and x_{max} are updated depending on the sign of $F(x_g)$ using

$$\text{If } F(x_g) > 0 \quad x_{min} = x_g,$$

and

$$\text{If } F(x_g) < 0 \quad x_{max} = x_g.$$

However, the above update is not unique and depends on the functional form of F. For instance, the proper update may be the opposite—that is,

$$\text{If } F(x_g) > 0 \quad x_{max} = x_g,$$

and

$$\text{If } F(x_g) < 0 \quad x_{min} = x_g.$$

The proper criterion for updating the range must be determined on a case-by-case basis, by keeping track of x_g: if x_g between iterations remains unchanged, the criterion must be reversed.

The procedure is terminated when $|F(x_g)| < \epsilon$, where ϵ is a predetermined small number typically of the same order as the machine accuracy. In general, the method works well and yields the root of the expression provided the range within which the solution lies is known.

EXAMPLE 13.1

Problem Statement Using the bisection method, find the friction factor by solving the Colebrook formula,

$$\frac{1}{\sqrt{f}} = -0.869 \ln\left(\frac{e/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right),$$

for flow through a pipe with $Re = 298{,}805$ and roughness ratio $e/D = 0.0004$.

SOLUTION For the solution procedure, the equation is expressed as

$$F = \frac{1}{\sqrt{f}} + 0.869 \ln\left(\frac{e/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right).$$

The root of $F(f)$ is the unknown friction factor f. Here we use $\epsilon = 10^{-6}$.

The method is implemented in the Fortran program found in Appendix D. Note the proper criterion for updating the limits f_{min} and f_{max}. For the given conditions with the limits initially at $f_{min} = 0$ and $f_{max} = 0.2$, the program gives the following intermediate estimates for the friction factor: 0.1, 0.05, 0.025, ... until it converges in 24 iterations to $f = 0.0176$.

Newton-Raphson Method

The Newton-Raphson method is based on Taylor's expansion series: if x_i is an estimate for the root x_r, the function expanded around x_i is then given as

$$F(x_r) = F(x_i) + F'(x_i)\,dx + F''(x_i)(dx)^2 + \cdots,$$

where primes such as F' denote differentiation with respect to x.

By definition, if x_r is the root of $F(x)$, then $F(x_r) = 0$. Therefore, by considering only the first two terms of the foregoing series, an appropriate correction $dx = x_{i+1} - x_i$ of the current estimate x_i is given as

$$dx = x_{i+1} - x_i = -\frac{F(x_i)}{F'(x_i)}.$$

As shown in Figure 13.4, geometrically the method is equivalent to approximating the function by a linear function using the local tangent. Formally, then, using an initial estimate of x_i, the iteration proceeds by correcting the estimate according to

$$x_{i+1} = x_i - \frac{F(x_i)}{F'(x_i)},$$

until $|x_{i+1} - x_i| < \epsilon$, where ϵ is again a small tolerance number (usually 10^{-6}).

In general, the method works very well having one of the fastest convergence rates (quadratic). However, the method has two limitations: (a) unless the initial guess is sufficiently close to the root, there is no guarantee that the procedure will converge; and (b) when $F' = 0$, the method breaks down.

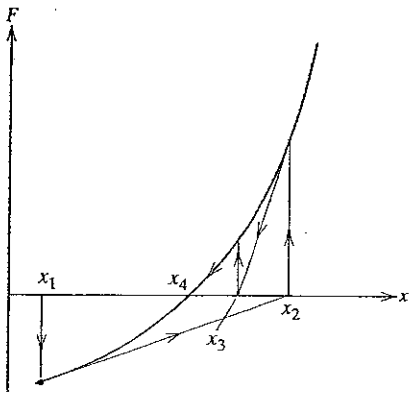


FIGURE 13.4 Geometric description of the Newton-Raphson method.

EXAMPLE 13.2

Problem Statement Using the Newton-Raphson method, find the friction factor by solving the Colebrook formula:

$$\frac{1}{\sqrt{f}} = -0.869 \ln\left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right)$$

for flow through a pipe with $Re = 298{,}805$ and roughness ratio $\varepsilon/D = 0.0004$.

SOLUTION For the solution procedure, the equation is again expressed as

$$f = \frac{1}{\sqrt{f}} + 0.869 \ln\left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right)$$

Again, the root of $F(f)$ is the unknown friction factor f. The tolerance is selected again as $\varepsilon = 10^{-6}$. This method is implemented in the Fortran program found in Appendix D. For the given conditions, and initial estimate (as discussed in Chapter 8) given by,

$$f_1 = 0.25\left[0.434 \ln\left(\frac{\varepsilon/D}{3.7} + \frac{5.74}{Re^{0.9}}\right)\right]^{-2}$$

the program after three (3) iterations converges to $f = 0.0176$. This example shows the faster convergence of the Newton-Raphson iteration procedure as compared with the bisection method.

13.1.2 Numerical Integration

In many fluid problems, the solution to the problem is the integral of a function $f(x)$ over a certain domain. For instance, in hydrostatics in order to determine the total force, we must integrate the pressure over submerged surfaces. Therefore, when the function $f(x)$ is complicated, the integral must be performed numerically. For a given function $f(x)$ the integral

$$A = \int_{x_1}^{x_2} f(x)\,dx$$

is the area under the graph, as shown in Figure 13.5.

An obvious approach is to subdivide the domain into smaller strips and add the area of each one of them. Indeed, numerical integration starts by dividing the range between x_1 and x_2 into n strips. Including the two ends, then, we have $n + 1$ points along the x-axis.

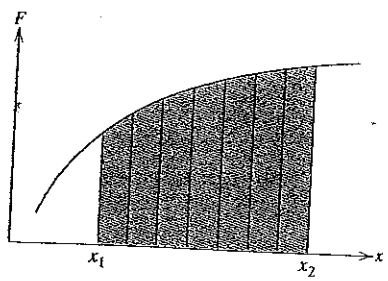


FIGURE 13.5 Schematic of numerical integration.

HAND OUT 7: Bracketing methods (Chapter 2 of our syllabus). Source: Chapra, S. C., and Canale, R. P. (2006). *"Numerical methods for engineers."* McGraw-Hill, fifth edition.

CHAPTER 5

Bracketing Methods

This chapter on roots of equations deals with methods that exploit the fact that a function typically changes sign in the vicinity of a root. These techniques are called *bracketing methods* because two initial guesses for the root are required. As the name implies, these guesses must "bracket," or be on either side of, the root. The particular methods described herein employ different strategies to systematically reduce the width of the bracket and, hence, home in on the correct answer.

As a prelude to these techniques, we will briefly discuss graphical methods for depicting functions and their roots. Beyond their utility for providing rough guesses, graphical techniques are also useful for visualizing the properties of the functions and the behavior of the various numerical methods.

5.1 GRAPHICAL METHODS

A simple method for obtaining an estimate of the root of the equation $f(x) = 0$ is to make a plot of the function and observe where it crosses the x axis. This point, which represents the x value for which $f(x) = 0$, provides a rough approximation of the root.

EXAMPLE 5.1 The Graphical Approach

Problem Statement. Use the graphical approach to determine the drag coefficient c needed for a parachutist of mass $m = 68.1$ kg to have a velocity of 40 m/s after free-falling for time $t = 10$ s. *Note:* The acceleration due to gravity is 9.8 m/s^2.

Solution. This problem can be solved by determining the root of Eq. (PT2.4) using the parameters $t = 10$, $g = 9.8$, $v = 40$, and $m = 68.1$:

$$f(c) = \frac{9.8(68.1)}{c}\left(1 - e^{-(c/68.1)10}\right) - 40$$

or

$$f(c) = \frac{667.38}{c}\left(1 - e^{-0.146843c}\right) - 40 \qquad \text{(E5.1.1)}$$

Various values of c can be substituted into the right-hand side of this equation to compute

c	f(c)
4	34.115
8	17.653
12	6.067
16	−2.269
20	−8.401

These points are plotted in Fig. 5.1. The resulting curve crosses the c axis between 12 and 16. Visual inspection of the plot provides a rough estimate of the root of 14.75. The validity of the graphical estimate can be checked by substituting it into Eq. (E5.1.1) to yield

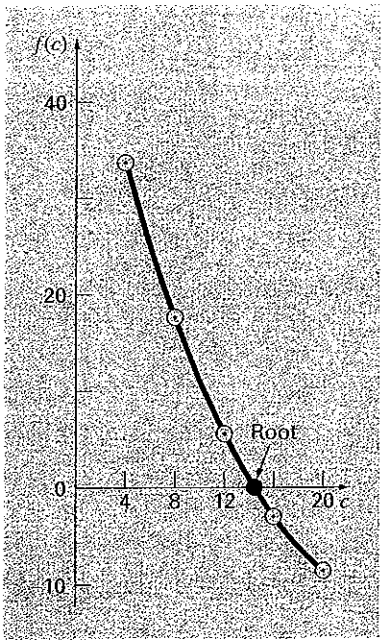
$$f(14.75) = \frac{667.38}{14.75}\left(1 - e^{-0.146843(14.75)}\right) - 40 = 0.059$$

which is close to zero. It can also be checked by substituting it into Eq. (PT2.4) along with the parameter values from this example to give

$$v = \frac{9.8(68.1)}{14.75}\left(1 - e^{-(14.75/68.1)10}\right) = 40.059$$

which is very close to the desired fall velocity of 40 m/s.

FIGURE 5.1
The graphical approach for determining the roots of an equation.



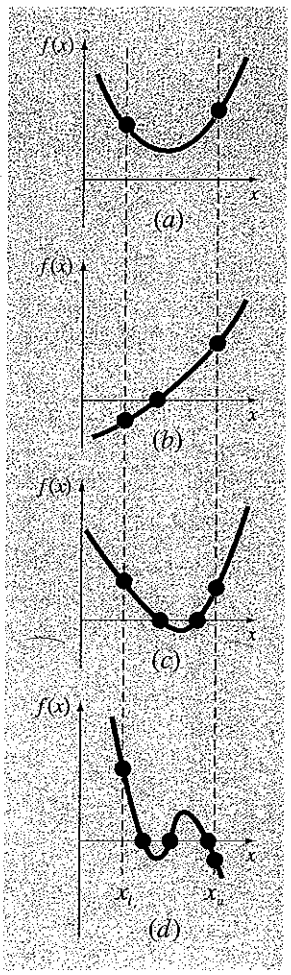


FIGURE 5.2
Illustration of a number of general ways that a root may occur in an interval prescribed by a lower bound x_l and an upper bound x_u. Parts (*a*) and (*c*) indicate that if both $f(x_l)$ and $f(x_u)$ have the same sign, either there will be no roots or there will be an even number of roots within the interval. Parts (*b*) and (*d*) indicate that if the function has different signs at the end points, there will be an odd number of roots in the interval.

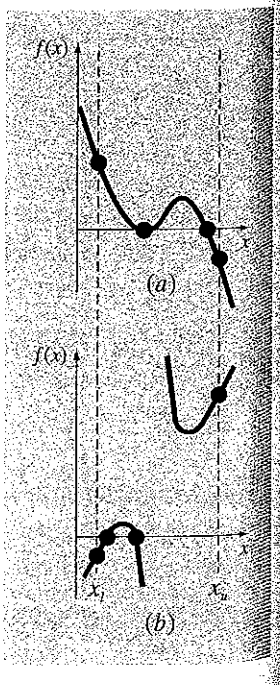
Graphical techniques are of limited practical value because they are not precise. However, graphical methods can be utilized to obtain rough estimates of roots. These estimates can be employed as starting guesses for numerical methods discussed in this and the next chapter.

Aside from providing rough estimates of the root, graphical interpretations are important tools for understanding the properties of the functions and anticipating the pitfalls of the numerical methods. For example, Fig. 5.2 shows a number of ways in which roots can occur (or be absent) in an interval prescribed by a lower bound x_l and an upper bound x_u. Figure 5.2*b* depicts the case where a single root is bracketed by negative and positive values of $f(x)$. However, Fig. 5.2*d*, where $f(x_l)$ and $f(x_u)$ are also on opposite sides of the x axis, shows three roots occurring within the interval. In general, if $f(x_l)$ and $f(x_u)$ have opposite signs, there are an odd number of roots in the interval. As indicated by Fig. 5.2*a* and *c*, if $f(x_l)$ and $f(x_u)$ have the same sign, there are either no roots or an even number of roots between the values.

Although these generalizations are usually true, there are cases where they do not hold. For example, functions that are tangential to the x axis (Fig. 5.3*a*) and discontinuous functions (Fig. 5.3*b*) can violate these principles. An example of a function that is tangential to the axis is the cubic equation $f(x) = (x - 2)(x - 2)(x - 4)$. Notice that $x = 2$ makes two terms in this polynomial equal to zero. Mathematically, $x = 2$ is called a *multiple root*. At the end of Chap. 6, we will present techniques that are expressly designed to locate multiple roots.

The existence of cases of the type depicted in Fig. 5.3 makes it difficult to develop general computer algorithms guaranteed to locate all the roots in an interval. However, when used in conjunction with graphical approaches, the methods described in the following

FIGURE 5.3
Illustration of some exceptions to the general cases depicted in Fig. 5.2. (*a*) Multiple root that occurs when the function is tangential to the x axis. For this case, although the end points are of opposite signs, there are an even number of axis intersections for the interval. (*b*) Discontinuous function where end points of opposite sign bracket an even number of roots. Special strategies are required for determining the roots for these cases.



sections are extremely useful for solving many roots of equations problems confronted routinely by engineers and applied mathematicians.

EXAMPLE 5.2 Use of Computer Graphics to Locate Roots

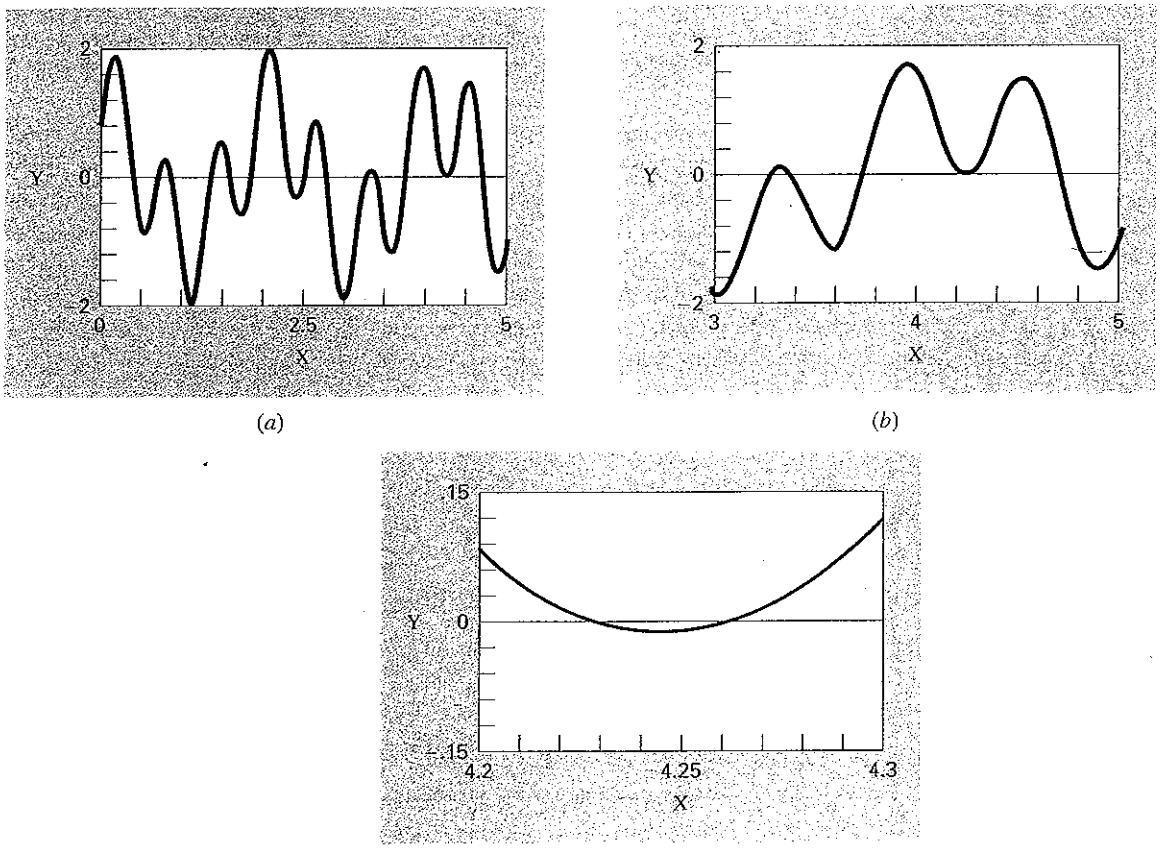
Problem Statement. Computer graphics can expedite and improve your efforts to locate roots of equations. The function

$$f(x) = \sin 10x + \cos 3x$$

has several roots over the range $x = 0$ to $x = 5$. Use computer graphics to gain insight into the behavior of this function.

Solution. Packages such as Excel and MATLAB software can be used to generate plots. Figure 5.4*a* is a plot of $f(x)$ from $x = 0$ to $x = 5$. This plot suggests the presence of several roots, including a possible double root at about $x = 4.2$ where $f(x)$ appears to be tangent to

FIGURE 5.4
The progressive enlargement of $f(x) = \sin 10x + \cos 3x$ by the computer. Such interactive graphics permits the analyst to determine that two distinct roots exist between $x = 4.2$ and $x = 4.3$.



the x axis. A more detailed picture of the behavior of $f(x)$ is obtained by changing the plotting range from $x = 3$ to $x = 5$, as shown in Fig. 5.4*b*. Finally, in Fig. 5.4*c*, the vertical scale is narrowed further to $f(x) = -0.15$ to $f(x) = 0.15$ and the horizontal scale is narrowed to $x = 4.2$ to $x = 4.3$. This plot shows clearly that a double root does not exist in this region and that in fact there are two distinct roots at about $x = 4.23$ and $x = 4.26$.

Computer graphics will have great utility in your studies of numerical methods. This capability will also find many other applications in your other classes and professional activities as well.

5.2 THE BISECTION METHOD

When applying the graphical technique in Example 5.1, you have observed (Fig. 5.1) that $f(x)$ changed sign on opposite sides of the root. In general, if $f(x)$ is real and continuous in the interval from x_l to x_u and $f(x_l)$ and $f(x_u)$ have opposite signs, that is,

$$f(x_l)\, f(x_u) < 0 \tag{5.1}$$

then there is at least one real root between x_l and x_u.

Incremental search methods capitalize on this observation by locating an interval where the function changes sign. Then the location of the sign change (and consequently, the root) is identified more precisely by dividing the interval into a number of subintervals. Each of these subintervals is searched to locate the sign change. The process is repeated and the root estimate refined by dividing the subintervals into finer increments. We will return to the general topic of incremental searches in Sec. 5.4.

The *bisection method*, which is alternatively called binary chopping, interval halving, or Bolzano's method, is one type of incremental search method in which the interval is always divided in half. If a function changes sign over an interval, the function value at the midpoint is evaluated. The location of the root is then determined as lying at the midpoint of the subinterval within which the sign change occurs. The process is repeated to obtain refined estimates. A simple algorithm for the bisection calculation is listed in Fig. 5.5, and a graphical depiction of the method is provided in Fig. 5.6. The following example goes through the actual computations involved in the method.

FIGURE 5.5

Step 1: Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)f(x_u) < 0$.

Step 2: An estimate of the root x_r is determined by

$$x_r = \frac{x_l + x_u}{2}$$

Step 3: Make the following evaluations to determine in which subinterval the root lies:

(a) If $f(x_l)f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

(b) If $f(x_l)f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

(c) If $f(x_l)f(x_r) = 0$, the root equals x_r; terminate the computation.

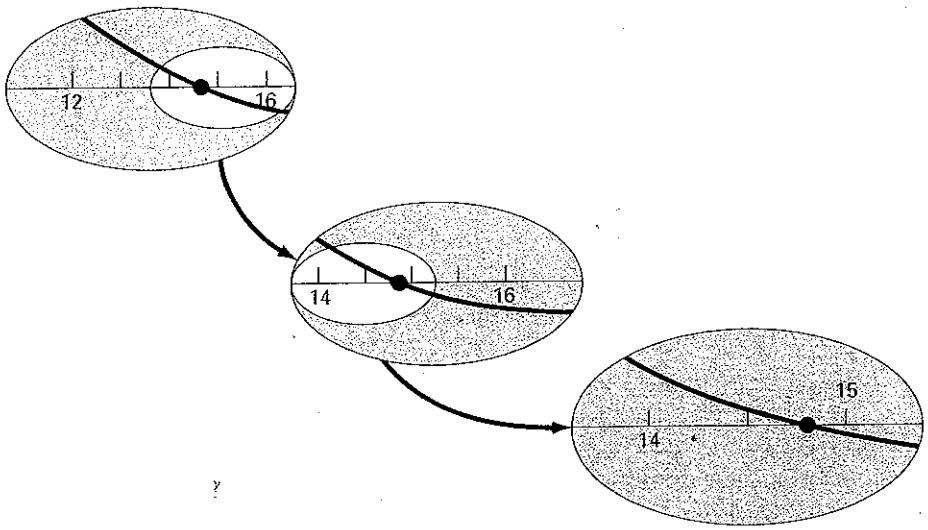


FIGURE 5.6
A graphical depiction of the bisection method. This plot conforms to the first three iterations from Example 5.3.

EXAMPLE 5.3 Bisection

Problem Statement. Use bisection to solve the same problem approached graphically in Example 5.1.

Solution. The first step in bisection is to guess two values of the unknown (in the present problem, c) that give values for $f(c)$ with different signs. From Fig. 5.1, we can see that the function changes sign between values of 12 and 16. Therefore, the initial estimate of the root x_r lies at the midpoint of the interval

$$x_r = \frac{12 + 16}{2} = 14$$

This estimate represents a true percent relative error of $\varepsilon_t = 5.3\%$ (note that the true value of the root is 14.7802). Next we compute the product of the function value at the lower bound and at the midpoint:

$$f(12)\,f(14) = 6.067(1.569) = 9.517$$

which is greater than zero, and hence no sign change occurs between the lower bound and the midpoint. Consequently, the root must be located between 14 and 16. Therefore, we create a new interval by redefining the lower bound as 14 and determining a revised root estimate as

$$x_r = \frac{14 + 16}{2} = 15$$

which represents a true percent error of $\varepsilon_t = 1.5\%$. The process can be repeated to obtain refined estimates. For example,

$$f(14)\,f(15) = 1.569(-0.425) = -0.666$$

Therefore, the root is between 14 and 15. The upper bound is redefined as 15, and the root estimate for the third iteration is calculated as

$$x_r = \frac{14 + 15}{2} = 14.5$$

which represents a percent relative error of $\varepsilon_t = 1.9\%$. The method can be repeated until the result is accurate enough to satisfy your needs.

In the previous example, you may have noticed that the true error does not decrease with each iteration. However, the interval within which the root is located is halved with each step in the process. As discussed in the next section, the interval width provides an exact estimate of the upper bound of the error for the bisection method.

5.2.1 Termination Criteria and Error Estimates

We ended Example 5.3 with the statement that the method could be continued to obtain a refined estimate of the root. We must now develop an objective criterion for deciding when to terminate the method.

An initial suggestion might be to end the calculation when the true error falls below some prespecified level. For instance, in Example 5.3, the relative error dropped from 5.3 to 1.9 percent during the course of the computation. We might decide that we should terminate when the error drops below, say, 0.1 percent. This strategy is flawed because the error estimates in the example were based on knowledge of the true root of the function. This would not be the case in an actual situation because there would be no point in using the method if we already knew the root.

Therefore, we require an error estimate that is not contingent on foreknowledge of the root. As developed previously in Sec. 3.3, an approximate percent relative error ε_a can be calculated, as in [recall Eq. (3.5)]

$$\varepsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\% \tag{5.2}$$

where x_r^{new} is the root for the present iteration and x_r^{old} is the root from the previous iteration. The absolute value is used because we are usually concerned with the magnitude of ε_a rather than with its sign. When ε_a becomes less than a prespecified stopping criterion ε_s, the computation is terminated.

EXAMPLE 5.4 Error Estimates for Bisection

Problem Statement. Continue Example 5.3 until the approximate error falls below a stopping criterion of $\varepsilon_s = 0.5\%$. Use Eq. (5.2) to compute the errors.

Solution. The results of the first two iterations for Example 5.3 were 14 and 15. Substituting these values into Eq. (5.2) yields

$$|\varepsilon_a| = \left| \frac{15 - 14}{15} \right| 100\% = 6.667\%$$

Recall that the true percent relative error for the root estimate of 15 was 1.5%. Therefore, ε_a is greater than ε_t. This behavior is manifested for the other iterations:

Iteration	x_l	x_u	x_r	ε_a (%)	ε_t (%)
1	12	16	14		5.279
2	14	16	15	6.667	1.487
3	14	15	14.5	3.448	1.896
4	14.5	15	14.75	1.695	0.204
5	14.75	15	14.875	0.840	0.641
6	14.75	14.875	14.8125	0.422	0.219

Thus, after six iterations ε_a finally falls below $\varepsilon_s = 0.5\%$, and the computation can be terminated.

These results are summarized in Fig. 5.7. The "ragged" nature of the true error is due to the fact that, for bisection, the true root can lie anywhere within the bracketing interval. The true and approximate errors are far apart when the interval happens to be centered on the true root. They are close when the true root falls at either end of the interval.

Although the approximate error does not provide an exact estimate of the true error, Fig. 5.7 suggests that ε_a captures the general downward trend of ε_t. In addition, the plot exhibits the extremely attractive characteristic that ε_a is always greater than ε_t. Thus, when

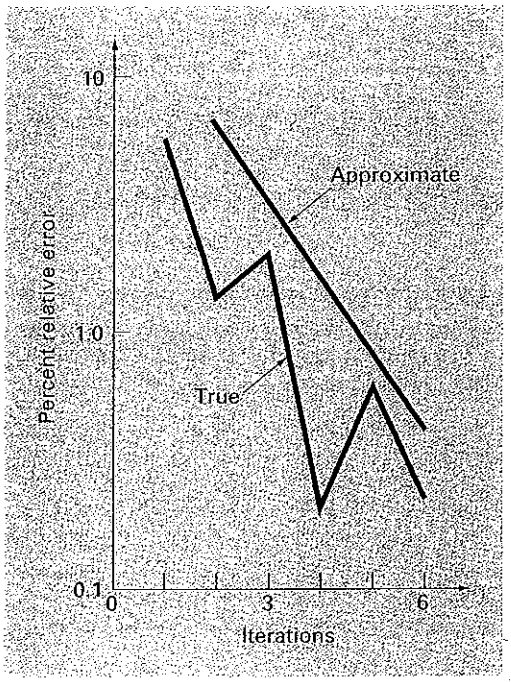


FIGURE 5.7
Errors for the bisection method. True and estimated errors are plotted versus the number of iterations.

ε_a falls below ε_s, the computation could be terminated with confidence that the root is known to be at least as accurate as the prespecified acceptable level.

Although it is always dangerous to draw general conclusions from a single example, it can be demonstrated that ε_a will always be greater than ε_t for the bisection method. This is because each time an approximate root is located using bisection as $x_r = (x_l + x_u)/2$, we know that the true root lies somewhere within an interval of $(x_u - x_l)/2 = \Delta x/2$. Therefore, the root must lie within $\pm\Delta x/2$ of our estimate (Fig. 5.8). For instance, when Example 5.3 was terminated, we could make the definitive statement that

$$x_r = 14.5 \pm 0.5$$

Because $\Delta x/2 = x_r^{\text{new}} - x_r^{\text{old}}$ (Fig. 5.9), Eq. (5.2) provides an exact upper bound on the true error. For this bound to be exceeded, the true root would have to fall outside the bracketing interval, which, by definition, could never occur for the bisection method. As illustrated in a subsequent example (Example 5.7), other root-locating techniques do not always behave as nicely. Although bisection is generally slower than other methods, the

FIGURE 5.8
Three ways in which the interval may bracket the root. In (*a*) the true value lies at the center of the interval, whereas in (*b*) and (*c*) the true value lies near the extreme. Notice that the discrepancy between the true value and the midpoint of the interval never exceeds half the interval length, or $\Delta x/2$.

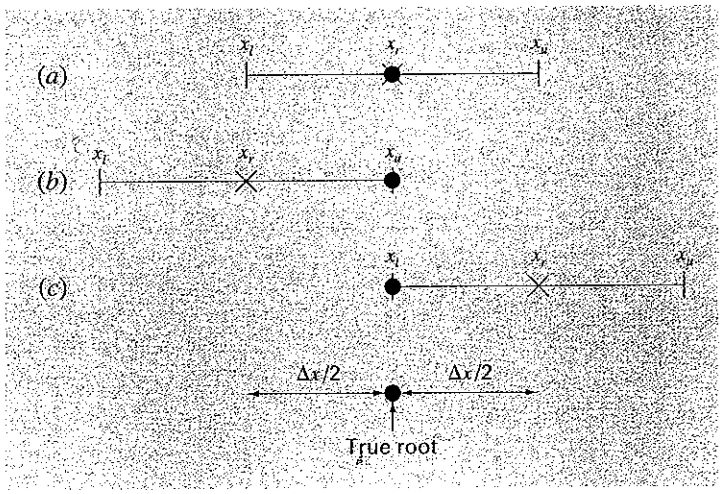
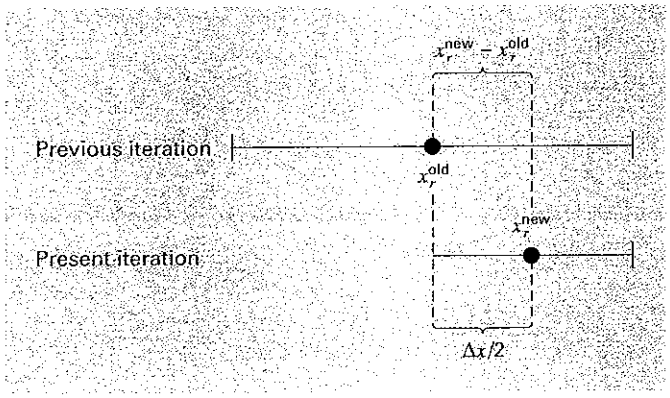


FIGURE 5.9
Graphical depiction of why the error estimate for bisection ($\Delta x/2$) is equivalent to the root estimate for the present iteration (x_r^{new}) minus the root estimate for the previous iteration (x_r^{old}).



neatness of its error analysis is certainly a positive aspect that could make it attractive for certain engineering applications.

Before proceeding to the computer program for bisection, we should note that the relationships (Fig. 5.9)

$$x_r^{\text{new}} - x_r^{\text{old}} = \frac{x_u - x_l}{2}$$

and

$$x_r^{\text{new}} = \frac{x_l + x_u}{2}$$

can be substituted into Eq. (5.2) to develop an alternative formulation for the approximate percent relative error

$$\varepsilon_a = \left| \frac{x_u - x_l}{x_u + x_l} \right| 100\% \tag{5.3}$$

This equation yields identical results to Eq. (5.2) for bisection. In addition, it allows us to calculate an error estimate on the basis of our initial guesses—that is, on our first iteration. For instance, on the first iteration of Example 5.2, an approximate error can be computed as

$$\varepsilon_a = \left| \frac{16 - 12}{16 + 12} \right| 100\% = 14.29\%$$

Another benefit of the bisection method is that the number of iterations required to attain an absolute error can be computed *a priori*—that is, before starting the iterations. This can be seen by recognizing that before starting the technique, the absolute error is

$$E_a^0 = x_u^0 - x_l^0 = \Delta x^0$$

where the superscript designates the iteration. Hence, before starting the method, we are at the "zero iteration." After the first iteration, the error becomes

$$E_a^1 = \frac{\Delta x^0}{2}$$

Because each succeeding iteration halves the error, a general formula relating the error and the number of iterations, n, is

$$E_a^n = \frac{\Delta x^0}{2^n} \tag{5.4}$$

If $E_{a,d}$ is the desired error, this equation can be solved for

$$n = \frac{\log\left(\Delta x^0 / E_{a,d}\right)}{\log 2} = \log_2\left(\frac{\Delta x^0}{E_{a,d}}\right) \tag{5.5}$$

Let us test the formula. For Example 5.4, the initial interval was $\Delta x_0 = 16 - 12 = 4$. After six iterations, the absolute error was

$$E_a = \frac{|14.875 - 14.75|}{2} = 0.0625$$

We can substitute these values into Eq. (5.5) to give

$$n = \frac{\log(4/0.0625)}{\log 2} = 6$$

Thus, if we knew beforehand that an error of less than 0.0625 was acceptable, the formula tells us that six iterations would yield the desired result.

Although we have emphasized the use of relative errors for obvious reasons, there will be cases where (usually through knowledge of the problem context) you will be able to specify an absolute error. For these cases, bisection along with Eq. (5.5) can provide a useful root-location algorithm. We will explore such applications in the end-of-chapter problems.

5.2.2 Bisection Algorithm

The algorithm in Fig. 5.5 can now be expanded to include the error check (Fig. 5.10). The algorithm employs user-defined functions to make root location and function evaluation more efficient. In addition, an upper limit is placed on the number of iterations. Finally, an error check is included to avoid division by zero during the error evaluation. Such would be the case when the bracketing interval is centered on zero. For this situation Eq. (5.2) becomes infinite. If this occurs, the program skips over the error evaluation for that iteration.

The algorithm in Fig. 5.10 is not user-friendly; it is designed strictly to come up with the answer. In Prob. 5.14 at the end of this chapter, you will have the task of making it easier to use and understand.

FIGURE 5.10
Pseudocode for function to implement bisection.

```
FUNCTION Bisect(xl, xu, es, imax, xr, iter, ea)
  iter = 0
  DO
    xrold = xr
    xr = (xl + xu) / 2
    iter = iter + 1
    IF xr ≠ 0 THEN
      ea = ABS((xr - xrold) / xr) * 100
    END IF
    test = f(xl) * f(xr)
    IF test < 0 THEN
      xu = xr
    ELSE IF test > 0 THEN
      xl = xr
    ELSE
      ea = 0
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Bisect = xr
END Bisect
```

5.2.3 Minimizing Function Evaluations

The bisection algorithm in Fig. 5.10 is just fine if you are performing a single root evaluation for a function that is easy to evaluate. However, there are many instances in engineering when this is not the case. For example, suppose that you develop a computer program that must locate a root numerous times. In such cases you could call the algorithm from Fig. 5.10 thousands and even millions of times in the course of a single run.

Further, in its most general sense, a univariate function is merely an entity that returns a single value in return for a single value you send to it. Perceived in this sense, functions are not always simple formulas like the one-line equations solved in the preceding examples in this chapter. For example, a function might consist of many lines of code that could take a significant amount of execution time to evaluate. In some cases, the function might even represent an independent computer program.

Because of both these factors, it is imperative that numerical algorithms minimize function evaluations. In this light, the algorithm from Fig. 5.10 is deficient. In particular, notice that in making two function evaluations per iteration, it recalculates one of the functions that was determined on the previous iteration.

Figure 5.11 provides a modified algorithm that does not have this deficiency. We have highlighted the lines that differ from Fig. 5.10. In this case, only the new function value at

FIGURE 5.11
Pseudocode for bisection subprogram which minimizes function evaluations.

```
FUNCTION Bisect(xl, xu, es, imax, xr, iter, ea)
  iter = 0
  fl = f(xl)
  DO
    xrold = xr
    xr = (xl + xu) / 2
    fr = f(xr)
    iter = iter + 1
    IF xr ≠ 0 THEN
      ea = ABS((xr - xrold) / xr) * 100
    END IF
    test = fl * fr
    IF test < 0 THEN
      xu = xr
    ELSE IF test > 0 THEN
      xl = xr
      fl = fr
    ELSE
      ea = 0
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Bisect = xr
END Bisect
```

the root estimate is calculated. Previously calculated values are saved and merely reassigned as the bracket shrinks. Thus, $n + 1$ function evaluations are performed, rather than $2n$.

5.3 THE FALSE-POSITION METHOD

Although bisection is a perfectly valid technique for determining roots, its "brute-force" approach is relatively inefficient. False position is an alternative based on a graphical insight.

A shortcoming of the bisection method is that, in dividing the interval from x_l to x_u into equal halves, no account is taken of the magnitudes of $f(x_l)$ and $f(x_u)$. For example, if $f(x_l)$ is much closer to zero than $f(x_u)$, it is likely that the root is closer to x_l than to x_u (Fig. 5.12). An alternative method that exploits this graphical insight is to join $f(x_l)$ and $f(x_u)$ by a straight line. The intersection of this line with the x axis represents an improved estimate of the root. The fact that the replacement of the curve by a straight line gives a "false position" of the root is the origin of the name, *method of false position,* or in Latin, *regula falsi.* It is also called the *linear interpolation method.*

Using similar triangles (Fig. 5.12), the intersection of the straight line with the x axis can be estimated as

$$\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u} \tag{5.6}$$

which can be solved for (see Box 5.1 for details).

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)} \tag{5.7}$$

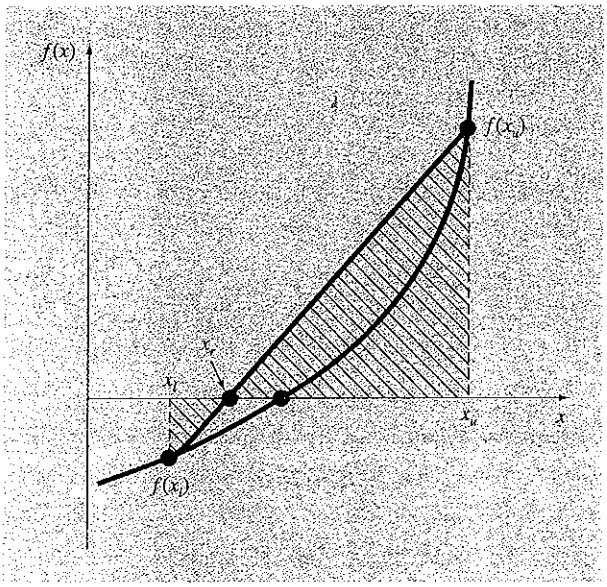


FIGURE 5.12
A graphical depiction of the method of false position. Similar triangles used to derive the formula for the method are shaded.