CHAPTER 5
COMPUTATIONAL CONSIDERATIONS

Charles R. Mischke, Ph.D., P.E.
Professor Emeritus of Mechanical Engineering
Iowa State University
Ames, Iowa

NOMENCLATURE

a	Range number
A	Spring wire strength constant, cross-sectional area, Jacobian matrix
allow	Diametral allowance
b	Range number
B	Bushing diameter
c	Distance to outer fiber, radial clearance
C	Spring index D/d
d	Wire diameter
d_{hole}	Hole diameter
d_{rod}	Rod diameter
D	Helix diameter, journal diameter
e	Eccentricity
E	Young's modulus
fom	Figure of merit
$f(x)$	Function
F	Spring force, cumulative distribution function, function
F_1	Spring working load
F_s	Spring load at closure (soliding)

G	Shear modulus
h	Ordinate spacing in Simpson's rule
i	Subscript
k	Spring rate, successive substitution convergence parameter
ℓ	Length
ℓ_o	Free length
ℓ_s	Solid length
L	Length
L_o	Free length
L_s	Solid length
m	Spring wire strength parameter
n	Number, factor of safety
n_s	Factor of safety at soliding
$\sim N$	Normal (or gaussian) distributed
N	Number of turns
N_a	Number of active turns
N_t	Total number of turns
OD	Outside diameter of spring coil
p	Probability
P	Load, probability
Q	Spring dead coil correction
Q'	Spring dead coil correction for solid height
r	Residual, radius
R	Richardson's correction to Simpson's first rule estimate
S_{ut}	Engineering ultimate tensile strength
S_{su}	Engineering ultimate shear strength
S_y	Engineering 0.2 percent yield strength in tension
S_{sy}	Engineering 0.2 percent yield strength in shear
$\mathbf{u}$	Uniform random number
$\sim U$	Uniform distributed
$\sim W$	Weibull distributed
x	Variable
y	Variable, end contraction of a spring
z	deviation in $N(0, 1)$
γ	Weight density
η	Factor of safety
θ	Weibull characteristic parameter, angle
μ	Population mean
ξ	Fractional overrun to closure, $y_s = y_1 + \xi y_1$
ρ	Link length

σ	Population standard deviation
σ	Normal stress
τ	Shear stress
τ_s	Shear stress at wire surface at closure of spring
ϕ	angle

5.1 INTRODUCTION

Machine design is the decision-making process by which specifications for machines are created. It is from these specifications that materials are ordered and machines are manufactured. The process includes

- Inventing the concept and connectivity
- Decisions on size, material, and method of manufacture
- Secondary decisions
- Adequacy assessment
- Documentation of the design
- Construction and testing of prototype(s)
- Final design

Computer-aided engineering (CAE) means computer assistance in the major decision-making process. *Computer-aided drafting* (CAD), often confused with CAE when called *computer-aided design,* means computer assistance in creating plans and can include estimates of such geometric properties as volume, weight, centroidal coordinates, and various moments about the centroid. Three-dimensional depictions and their manipulations are often routinely available. *Computer-aided analysis* (CAA) involves use of the computer in an "if this then that" mode.

Computer-aided manufacturing (CAM) includes preparing tool passes for manufacture, including generating codes for executing complicated tool paths for numerically controlled machine tools. All kinds of auxiliary accounting associated with material and parts flow in a manufacturing line are also done by computer. The data base created during computer-aided drafting can be used by computer-aided manufacturing. This is often called *CAD/CAM.*

Some of these computer aids are commercially available and use proprietary programming. They are sometimes called "turnkey" systems. They may be used interactively by technically competent people without programming knowledge after only modest instruction. The programming detail is not important to the users. They react to displays, make decisions on the task to be accomplished, and proceed by entering appropriate system commands. Such systems are available for a number of highly repetitive tasks found in analysis, drawing, detailing, and manufacturing.

"Turnkey" systems are available from vendors to do some important work.

The machine designer's effort, however, is composed of problem-specific tasks, for many of which no commercial programming is available. The designer or his or her assistants may have to create or supervise the creation of such programs. The basis for this programming must be their understanding of the problem. This section will view computer methods of direct use to the designer in making decisions using personal or corporate resources.

It is well to keep in mind what the computer can do:

- It can remember data and programs.
- It can calculate.
- It can branch unconditionally.
- It can branch conditionally based on whether a quantity is negative, zero, or positive, or whether a quantity is true or false, or whether a quantity is larger or smaller than something else. This capability can be described as *decision making.*
- It can do a repetitive task or series of tasks a fixed number of times or an appropriate number of times based on calculations it performs. This can be called *iteration.*
- It can read and write alphabetical and numerical information.
- It can draw.
- It can pause, interact, and wait for external decisions or thoughtful input.
- It does not tire.

Humans can

- Understand the problem
- Judge what is important and unimportant
- Plan strategies and modify them as they gain experience
- Weigh intangibles
- Be skeptical, suspicious, or unconvinced
- Program computers

The designer should try to delegate to the computer those things which the computer can do well and reserve for humans those things which they do well.

5.2 AN ALGORITHMIC APPROACH TO DESIGN

A design must be functional, safe, reliable, competitive, manufacturable, and marketable. It is axiomatic that the designer must have a quantitative procedural structure in mind before computer programming is attempted. An algorithm is a step-by-step process for accomplishing a task. The designer contemplating using the computer to help in making decisions undertakes a series of tasks that include

1. Identifying the specification set
2. Identifying the decision set
3. Examining the needs to be addressed, noting the a priori decisions
4. Identifying the design variables
5. Quantifying the adequacy assessment
6. Converting the a priori decisions and design decisions into a specification set
7. Quantifying a figure of merit
8. Choosing an optimization algorithm
9. Assembling the programs

A *specification set* for a machine or component is the ensemble of drawings, text, bill of materials, and other directions that assure function, safety, reliability, competitiveness, manufacturability, and marketability no matter who builds it, assembles it, and uses it. For example, consider a helical coil compression spring for static service, such as that depicted in Fig. 5.1. The spring maker needs to know (one possible form)

- Material and its condition
- End treatment
- Coil ID or OD and tolerance
- Total turns and tolerance
- Free length and tolerance
- Wire size and tolerance

The commercial tolerances are expressible as functions of mean or median values. There are six elements in the specification set. The specification set is not couched in terms of the designer's thinking parameters or concerns, and so the designer recasts it as a decision set. The sets are equivalent, and the specification set is deducible from the decision set using ordinary deductive analytic algebraic techniques.

A *decision set* is the set of decisions which, when made, establishes the specification set. The specification set is cast the way the spring maker likes to communicate, and the decision set is expressed in such a way that the engineer can focus on function, safety, reliability, and competitiveness. In the case of the spring, a corresponding decision set is

- Material and condition
- End condition
- Function: F_1 at y_1, or F_1 at L_1
- Safety: Design factor at soliding is $n_s = 1.2$

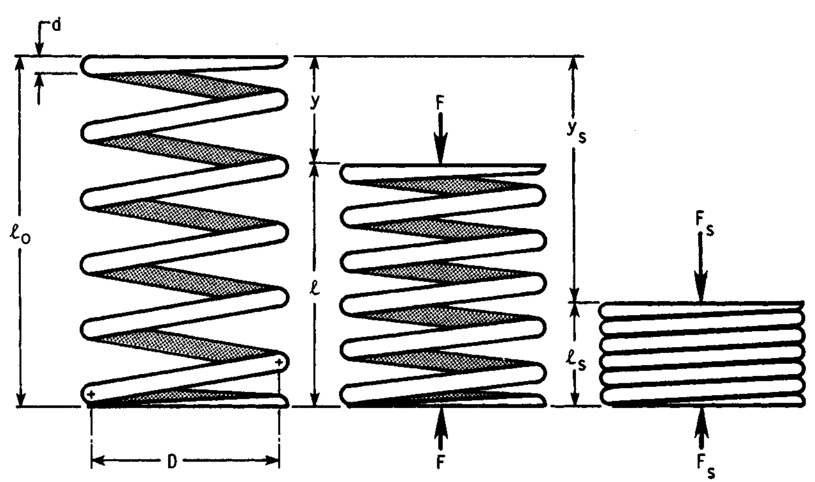

FIGURE 5.1 Nomenclature of a helical-coil compression spring with squared and ground ends.

- Robust linearity: Fractional overrun to closure $\xi = 0.15$
- Wire size: d

Note some duplication of elements in the decision set and the specification set, but also the appearance of "thinking parameters." The functional requirement of a force-geometry relationship occurs indirectly; it is important that the spring be robustly linear, and this requirement prevents the use of excess spring material while assuring no change in active turns as the coil clashes during the approach to soliding. Had the designer said that the design factor at soliding was to be greater than 1.2 (that is, $n_s > 1.2$), that would be a nondecision, and another decision would have to be added to the decision set. Any inequalities the designer is tempted to place in the decision set are moved to the adequacy assessment.

The cheapest spring is made from hard-drawn spring wire; the next stronger (and more expensive) material, 1065 OQ&T, costs 30 percent more. End treatment will almost always be squared and ground. The function is not negotiable, nor is the linearity requirement. The spring must survive closure without permanent deformation. These five decisions can be made a priori, and are consequently called *a priori decisions.* The remaining decision, that of wire size, is the decision through which the issue of competitiveness is addressed. Thus, there is one independent design variable, wire size. Note that the dimensionality of the task has been identified. Here, wire size is called the *design variable.*

An *adequacy assessment* consists of the cerebral, empirical, and related steps undertaken to determine if a specification set is satisfactory or not. In the case of the spring, the adequacy assessment can look as follows:

$4 \le C \le 16$	(formable and not too limber)
$3 \le N_a \le 15$	(sufficient turns for load precision)
$\xi \ge 0.15$	(robust linearity and little excess material)
$n_s \ge 1.2$	(spring can survive closure without permanent deformation)
$\vdots$	

Additional checks can examine natural frequencies, buckling, etc., as applicable.

A *figure of merit* is a number whose magnitude is a monotonic index to the merit or desirability of a specification set (or decision set). If several satisfactory springs are discovered, the figure of merit is used to compare them and choose the best. In the case where large numbers of satisfactory specification sets are expected, an optimization strategy is needed so that the best can be identified without exhaustive examination. In the spring example, since springs sell by the pound, a useful figure of merit is

$$\text{fom} = -(\text{relative material cost})\,\frac{\gamma\pi^2 d^2 N_t D}{4}$$

Often the competition is among steels and the weight density γ can be omitted.

An optimization strategy is chosen in the light of the number of design variables present (dimensionality), the number and kinds of constraints (equality, inequality, or mixed; few or many), and whether the decision variables are integer (or discrete) or continuous [5.1].

There are two related skills for the designer to master. The first skill is the ability to take a specification set and perform an adequacy assessment. The logic flow diagram for this skill is depicted in Fig. 5.2. Such a skill is analytic and deductive. The second skill is to create a specification set by surrounding the results of skill #1 with a decision set, a figure of merit, and an optimization strategy. Study Fig. 5.3 to see the interrelationships. This second skill is a synthesis procedure which is quantitative and computer-programmable. An example follows to show how simply this can be done, how a hand-held programmable calculator can be the only computational tool needed, and how some tasks can be done manually.

Example 1. A static-service helical-coil compression spring is to be made of 1085 music wire (food service application). The static load is to be 18 lbf when the spring is compressed 2.25 in. The geometric constraints are

$$0.5 \le \mathrm{ID} \le 1.25 \text{ in}$$

$$0.5 \le \mathrm{OD} \le 1.50 \text{ in}$$

$$0.5 \le L_s \le 1.25 \text{ in}$$

$$3 \le L_o \le 4 \text{ in}$$

Solution. The decision set with a priori decisions in place is

- Material and condition: music wire, $A = 186\ 000$ psi, $m = 0.163$, $E = 30 \times 10^6$ psi, $G = 11.5 \times 10^6$ psi
- End treatment: squared and ground, $Q = 2$, $Q' = 1$
- Function: $F_1 = 18$ lbf, $y_1 = 2.25$ in
- Soliding design factor: $n_s = 1.2$
- Fractional overrun to closure: $\xi = 0.15$
- Wire size: d

The decision variable is the wire size d. The figure of merit is cost relative to that of cold-drawn spring wire (Ref. [5.2], p. 20).

$$\text{fom} = -(\text{cost relative to CD})(\text{volume of wire used to make spring})$$

$$= -2.6\,\frac{\pi^2 d^2 (N_a + Q) D}{4}$$

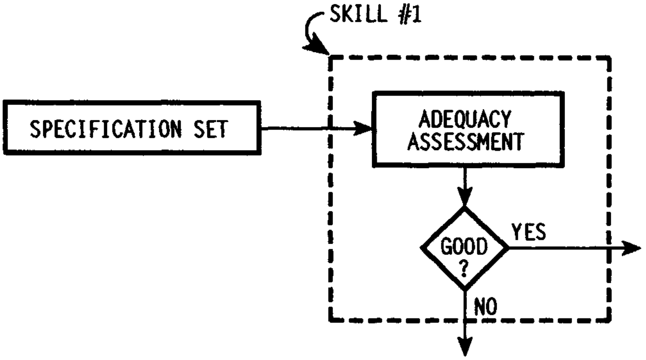

FIGURE 5.2 Designer's skill #1.

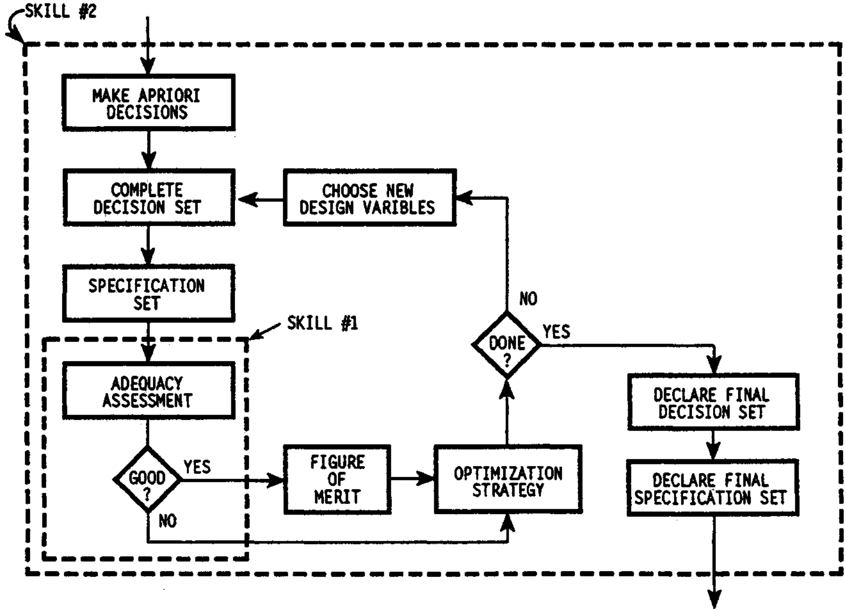

FIGURE 5.3 Designer's skill #2, which contains skill #1 imbedded.

Procedure: From the potential spring maker, get a list of available music wire sizes. Mentally choose a wire size d. The decision set is complete, so find a path to the specification set. What follows is one such path from the decision on wire size d with the three possibilities (spring works over a rod, spring is free to take on any diameter, spring works in a hole).

$$S_{sy} = 0.45A/d^m$$

Spring over a rod	Spring is free	Spring in a hole
$D = d_{\text{rod}} + d + \text{allow}$	$D = \dfrac{S_{sy}\,\pi d^3}{8(1+\xi)F_1} - \dfrac{d}{2}$	$D = d_{\text{hole}} - d - \text{allow}$

$$C = D/d$$

$$\tau_s = \frac{(1 + 0.5/C)8(1+\xi)F_1 D}{\pi d^3}$$

$$n_s = S_{sy}/\tau_s$$

$$\text{OD} = D + d$$

$$\text{ID} = D - d$$

$$N_a = d^4 G y_1/(8D^3F_1)$$

$$N_t = N_a + Q$$

$$L_s = (N_a + Q')d$$

$$L_o = L_s + (1 + \xi)y_1$$

The specification set has been identified; now perform the adequacy assessment:

$$0.5 \le \text{ID} \le 1.25 \text{ in}$$

$$0.5 \le \text{OD} \le 1.50 \text{ in}$$

$$0.5 \le L_s \le 1.25 \text{ in}$$

$$3 \le L_o \le 4 \text{ in}$$

$$4 \le C \le 16$$

$$3 \le N_a \le 12$$

$$\xi \ge 0.15$$

$$n_s \ge 1.2$$

$$\vdots$$

The above computational steps can be programmed on a computer using a language such as Fortran, or on a hand-held programmable pocket calculator. The following table comes from a pocket calculator when one inputs wire size and the remaining elements of the column are presented.

d	0.031	0.041	0.063	0.067	0.071	0.075	0.080	0.085	0.090
D	0.054	0.133	0.448	0.585	0.693	0.814	0.983	1.172	1.383
C	1.740	3.244	7.742	8.729	9.766	10.85	12.28	13.79	15.37
OD	0.085	0.174	0.551	0.652	0.764	0.889	1.063	1.257	1.463
N_a	1057	215.4	24.4	18.1	13.7	10.5	7.76	5.827	4.454
L_s	32.81	8.890	1.600	1.280	1.044	0.866	0.701	0.580	0.491
L_o	35.39	11.48	4.187	3.867	3.631	3.453	3.288	3.168	3.078
fom	−0.352	−0.312	−0.328	−0.339	−0.352	−0.368	−0.394	−0.425	−0.464

Figure 5.4 shows a plot of the figure of merit vs. wire size d. Only four wire diameters result in satisfactory springs, 0.071, 0.075, 0.080, and 0.085 in, and the largest figure of merit, −0.352, of these four springs corresponds to a wire diameter of 0.071 in. Ponder the structure that identified the dimensionality of the task and guided the component computational arrangement.

5.3 ANALYSIS TASKS

In the discussion in the previous section of the adequacy-assessment task and the conversion to a specification set as illustrated by the static-service spring example, there occurred a number of routine computational chores. These were simple algebraic expressions representing mathematical models of the reality we call a spring.

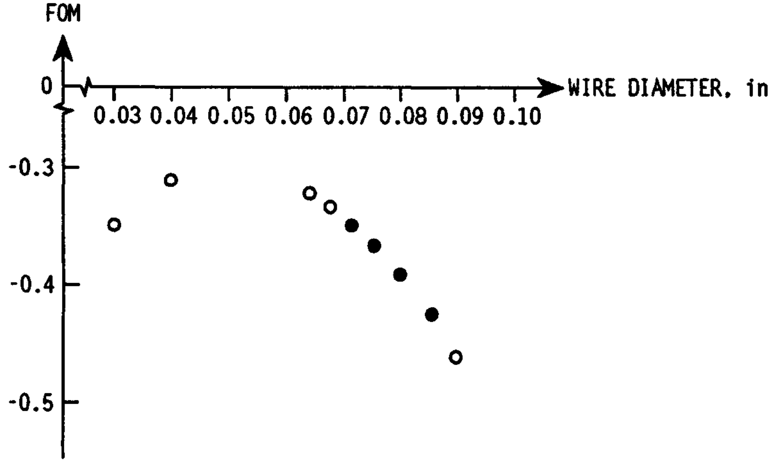

FIGURE 5.4 The figure of merit as a function of wire diameter in Example 5.1. The solid points are satisfactory springs.

The expression for spring rate is either remembered or easily found. In a more complex problem, the computational task may be more involved and harder to execute and program. However, it is of the same character. It is a calculation ritual that is known by the engineering community to be useful. It is an analysis-type "if this then that" algorithm that engineers instinctively reach for under appropriate circumstances. If this happens often, then once it is programmed, it should be available for subsequent use by anyone. Computer languages created for algebraic computational use include a feature called *subprogram capability.* The algorithm encoded is given a name and an argument list. In Fortran such a program can be a function subprogram or a subroutine subprogram. If the spring rate equation were to be coded as a Fortran subroutine with the name SPRNGK, then the coding could be

```
SUBROUTINE SPRNGK(DWIRE,DCOIL,G,EN,XK)
XK=DWIRE**4*G/8./DCOIL**3/EN
RETURN
END
```

and any program in which DWIRE, DCOIL, G, and EN have been defined can obtain XK by

```
⋮
CALL SPRNGK(DWIRE,DCOIL,G,EN,XK)
⋮
```

and XK is now defined in the calling program. This simplicity is welcome as the tasks become more complicated, such as finding the stress at the inner fiber of a curved beam of a tee cross section or locating the neutral axis of the cross section.

Such routine answers to computational chores can be added to a subroutine library to which the computer and the users have access. Usage by one designer of programs written by another person depends on documentation, error messaging, and tests. At this point and for our purposes, we will treat this as detail and retain the larger picture. A library of analysis subroutines can be created which the designer can manipulate in an executive manner simply by calling appropriate routines. Such subroutines are called *design subroutines* because through an inverse-analysis strategy they can be made to yield design decisions. Within them is the essence of the reality of the physical world. When decisions are made which completely describe a helical compression spring intended for static use, the computer can be used to examine important features. From decisions on

Material:	1085 music wire
Wire size:	0.071 in
Ends:	Ground and squared
Turns:	21.7 total
OD:	0.685 in
Length:	4.476 in

a large number of attributes can be viewed:

Ultimate strength estimate:	288 kpsi
Shearing yield strength:	125 kpsi
Spring rate:	8.01 lbf/in
Solid length:	1.47 in
Deflection to closure:	3.01 in
Working force:	7.8 lbf
Working deflection:	0.98 in
Working length:	3.5 in
Force at closure:	24.1 lbf
Spring index:	8.65
Shear stress at closure:	111 kpsi
Working shear stress:	36.1 kpsi
Static factor of safety:	3.5
Factor of safety at closure:	1.12
Critical frequency:	135 Hz
Buckling load:	9.6 lbf

Scanning these items, the designer can detect the need for a constraint to prevent buckling and observe the low factor of safety guarding against permanent set due to closure. This too can be assessed, and the computer can assist routinely. The static factor of safety at closure is given by [see Ref. [5.3], Eqs. (10.3), (10.17) and $S_y = 0.75S_{ut}$]

$$\eta_s = \frac{S_{sy}}{\tau_s} = \frac{0.577(0.75)A}{d^m} \, \frac{\pi d^3}{[1 + (0.5d/D)]8F_sD}$$

Substituting for force to closure

$$F_s = ky_s = \frac{d^4G}{8D^3N}\,[\ell_o - (N+1)d]$$

into the η_s equation yields

$$\eta_s = \frac{0.577(0.75)A\pi D^2 N}{[1 + (0.5d/D)]d^{1+m}\, G[\ell_f - (N+1)d]}$$

for ground and squared ends. Tolerances on d, D, N, and ℓ_o give rise to variation in η_s, that tabulated above being a median value. The worst-case stacking of tolerance occurs when all deviations from the midrange values are such that

$$\Delta\eta_s = \left|\frac{\partial\eta_s}{\partial d}\Delta d\right| + \left|\frac{\partial\eta_s}{\partial D}\Delta D\right| + \left|\frac{\partial\eta_s}{\partial N}\Delta N\right| + \left|\frac{\partial\eta_s}{\partial \ell_o}\Delta \ell_o\right|$$

In the case at hand with $A = 196$ kpsi (Ref. [5.3], Table 10-2), $m = 0.146$, $d = 0.071$ in, $D = 0.614$ in, $N = 19.7$, $\ell_o = 4.476$ in, and $G = 11.5 \times 10^6$ psi; $\eta_s = 1.121$:

$$\frac{\partial\eta_s}{\partial d} \doteq -11.03 \text{ in}^{-1} \qquad \frac{\partial\eta_s}{\partial D} \doteq 3.78 \text{ in}^{-1} \qquad \frac{\partial\eta_s}{\partial N} \doteq 0.083 \qquad \frac{\partial\eta_s}{\partial \ell_o} \doteq -0.372 \text{ in}^{-1}$$

where these values are obtained by taking partial derivatives in the η_s equation by numerical means. If the bilateral tolerances are

$$d = 0.071 \pm 0.001 \text{ in}$$

$$D = 0.614 \pm 0.010 \text{ in}$$

$$N = 19.7 \pm \tfrac{1}{4} \text{ turn}$$

$$\ell_o = 4.476 \pm 0.097 \text{ in}$$

we have

$$\Delta\eta_s = |-11.03(0.001)| + |3.78(0.010)| + |0.083(\tfrac{1}{4})| + |-0.372(0.097)|$$

$$= 0.011 + 0.038 + 0.021 + 0.036 = 0.106$$

The relative contribution of the various tolerances can be observed. The smallest possible value of η_s is

$$\eta_s(\min) = \eta_s - \Delta\eta_s = 1.121 - 0.106 = 1.02$$

This is the worst-case stacking of tolerances.

To make a statistical statement as to the probability of observing a value of η_s of a particular magnitude, we need an estimate of the variance of η_s.

$$\sigma^2_{\eta_s} = \left(\frac{\partial\eta_s}{\partial d}\right)^2 \sigma_d^2 + \left(\frac{\partial\eta_s}{\partial D}\right)^2 \sigma_D^2 + \left(\frac{\partial\eta_s}{\partial N}\right)^2 \sigma_N^2 + \left(\frac{\partial\eta_s}{\partial \ell_o}\right)^2 \sigma_{\ell_o}^2$$

We can estimate the individual variances on the basis that the tolerance width represents six standard deviations as shown in Chap. 2:

$$\sigma_d = \frac{2(0.001)}{6} = 0.000\,333 \text{ in}$$

$$\sigma_D = \frac{2(0.010)}{6} = 0.003\,33 \text{ in}$$

$$\sigma_N = \frac{2(\tfrac{1}{4})}{6} = 0.0833$$

$$\sigma_{\ell_o} = \frac{2(0.097)}{6} = 0.032 \text{ in}$$

The estimate of the variance and standard deviation of η_s is

$$\sigma_{\eta_s}^2 = (-11.03)^2(0.000\,333)^2 + (3.78)^2(0.003\,33)^2 + (0.083)^2(0.083)^2 + (0.372)^2(0.032)^2 = 0.000\,361$$

$$\sigma_{\eta_s} = \sqrt{0.000\,361} = 0.019$$

For a gaussian distribution of η_s there are 3 chances in 1000 of observing a deviation from the mean of 3(0.019) = 0.057, or about 1½ chances in 1000 of observing an instance of η_s less than 1.121 − 0.057 = 1.064. These kinds of analysis chores are easily built into a computer-adequacy display program. This is the kind of quantitative information designers need before they commit themselves.

5.4 MATHEMATICAL TASKS

In problems which are coded for computer assistance, a number of recurring mathematical tasks are encountered which can also be discharged by the computer as they are encountered. The procedure is to identify the pertinent algorithm and then code it as appropriate to your computer. Recurring tasks can be coded as subprograms which represent convenient building blocks for use in solving larger problems.

One frequently encountered task is that of finding a root or zero place of a function of a single independent variable. An effective algorithm for this task is the successive-substitution procedure with assured convergence. The algorithm is as follows (Ref. [5.7], p. 168):

Step 1. Express the problem in the form $f(x) = 0$. Establish the largest successive difference allowable in root estimates ϵ.

Step 2. Rewrite in the form $x = F(x)$, thereby defining $F(x)$.

Step 3. Establish the convergence parameter $k = 1/[1 - F'(x)]$ or the finite-difference equivalent.

Step 4. Write the iteration equation [see Ref. [5.7], Eq. (3.25)], that is,

$$x_{i+1} = [(1-k)x + kF(x)]_i$$

and begin with root estimate x_0.

Step 5. If $|x_{i+1} - x_i| < \epsilon$, stop; otherwise go to step 4.

A simple example whose root is known is to find the root of ln x:

Step 1. $f(x) = \ln x = 0$

Step 2. Solve for x by adding and subtracting x, to establish $F(x)$:

$$x - x + \ln x = 0$$

$$x = x - \ln x = F(x)$$

Step 3. Establish

$$k = \frac{1}{1 - F'(x)} = \frac{1}{1 - (1 - 1/x)} = x$$

Step 4. Write the iteration equation:

$$x_{i+1} = [(1 - x)x + x(x - \ln x)]_i$$
$$= [x(1 - \ln x)]_i$$

With $x_0 = 2$, the following successive approximations are obtained:

2.000 000 000

0.613 705 639

0.913 341 207

0.996 131 704

0.999 992 508

1.000 000 000

In 5 iterations, 10 correct digits have been obtained. For a programmable hand-held calculator using reversed Polish notation, the problem-specific coding could be

```
[A] STO1 lnx CHS 1 + RCL1 × R/S
```

As an example of a problem with unknown answer, consider a 2 × ¼ in tube of 1035 cold drawn steel ($S_y = 67$ kpsi) that is 48 in long and must support a column load with an eccentricity of ⅛ in, as depicted in Fig. 5.5. For a design factor of 4 on the load, what allowable load is predicted by the secant column equation [Ref. [5.3], Eq. (3.54)]? The equation is

$$\frac{nP}{A} = S_y \Bigg/ \left[1 + \frac{ec}{r^2}\sec\left(\frac{\ell}{r}\sqrt{\frac{nP}{A}\frac{1}{4E}}\right)\right]$$

where $A = 1.374$ in^2, $r = 0.625$ in, $e = 0.125$ in, $c = 1$ in, $\ell = 48$ in, $E = 30 \times 10^6$ psi, and $S_y = 67\ 000$ psi. The secant equation is of the form $nP/A = F(nP/A)$. Choosing $\Delta = 0.001\ nP/A$, we can construct a finite-difference approximation to $F'(nP/A)$ for use in estimating the convergence parameter:

$$k = 1 \Bigg/ \left[1 - \frac{F(1.001nP/A) - F(nP/A)}{0.001nP/A}\right]$$

Using the iteration equation,

$$\left(\frac{nP}{A}\right)_{i+1} = \left[(1 - k)\frac{nP}{A} + kF\left(\frac{nP}{A}\right)\right]_i$$

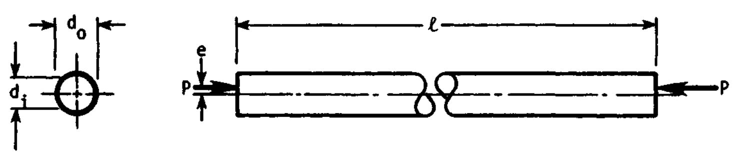

FIGURE 5.5 An eccentrically loaded hollow column.

Recalculating k every time and beginning with $(nP/A)_0 = 20\ 000$, the successive approximations are

$$\begin{array}{c} 20\ 000 \\ 34\ 004 \\ 32\ 548 \\ 32\ 518 \\ 32\ 518 \end{array}$$

It follows that

$$P = \frac{32\ 518A}{n} = \frac{32\ 518(1.374)}{4} = 11\ 170 \text{ lbf}$$

The range over which convergence is prompt is shown by using three different estimates, that is, $(nP/A)_0 = 1, 10\ 000$, and $50\ 000$:

1	10 000	50 000
39 967	36 353	38 322
33 367	32 726	33 015
32 528	32 519	32 521
32 518	32 518	32 518

Such an effective algorithm as successive substitution deserves coding on all kinds of computers. The designer should be able to perform the algorithm manually if required.

For finding the zero place of a function of more than one variable, a somewhat different formulation is useful. If a function of x is expanded about the point x_0 in the neighborhood of the root as a Taylor series (Ref. [5.5], p. 579), we obtain

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2!} f''(x_0)(x - x_0)^2 + \cdots$$

If x is the root, then $f(x) = 0$, and if the series is truncated after two terms, solution for x is a better estimate of the root than is x_0. Denoting $x - x_0$ as Δx, then

$$\Delta x = -\frac{f(x_0)}{f'(x_0)}$$

and the better estimate of the root is $x_1 = x_0 + \Delta x$. Using this pair of equations iteratively will result in finding the root. For example, if the involute of ϕ is 0.01, what is the value of ϕ? Recalling (Ref. [5.6], p. 266), that inv $\phi = \tan \phi - \phi$, we write

$$f(\phi) = \tan \phi - \phi - 0.01 = 0$$

$$f'(\phi) = \sec^2\phi - 1 = \frac{1}{\cos^2\phi} - 1$$

$$\Delta\phi = -\frac{f(\phi)}{f'(\phi)} = -\frac{\tan\phi - \phi - 0.01}{(1/\cos^2\phi) - 1}$$

$$\phi_{i+1} = \phi_i + \Delta\phi_i$$

For an initial estimate of $\phi_0 = 0.35$, we obtain

0.350 000 000

0.312 261 514

0.306 874 838

0.306 772 584

0.306 772 547

0.306 772 547

observing convergence to be rapid. For details, see any textbook on numerical methods, such as Carnahan et al. (Ref. [5.4], p. 171).

For the problem of two functions of two independent variables, namely, $f_1(x, y)$ and $f_2(x, y)$, the Taylor-series expansions are (Ref. [5.5], p. 580):

$$f_1(x, y) = f_1(x_0, y_0) + \frac{\partial f_1(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f_1(x_0, y_0)}{\partial y}(y - y_0) + \cdots$$

$$f_2(x, y) = f_2(x_0, y_0) + \frac{\partial f_2(x_0, y_0)}{\partial x}(x - x_0) + \frac{\partial f_2(x_0, y_0)}{\partial y}(y - y_0) + \cdots$$

If x and y represent the roots of $f_1(x, y) = 0$ and $f_2(x, y) = 0$, and identifying $(x - x_0) = \Delta x$ and $(y - y_0) = \Delta y$, then the preceding equations can be written as

$$\frac{\partial f_1}{\partial x}\Delta x + \frac{\partial f_1}{\partial y}\Delta y = -f_1 = r_1$$

$$\frac{\partial f_2}{\partial x}\Delta x + \frac{\partial f_2}{\partial y}\Delta y = -f_2 = r_2$$

where r_1 and r_2 are called *residuals.* Solving the preceding equations simultaneously for Δx and Δy, we obtain

$$\Delta x = \frac{1}{A}\left(r_1\frac{\partial f_2}{\partial y} - r_2\frac{\partial f_1}{\partial y}\right)$$

$$\Delta y = \frac{1}{A}\left(r_2\frac{\partial f_1}{\partial x} - r_1\frac{\partial f_2}{\partial x}\right)$$

where

$$A = \frac{\partial f_1}{\partial x}\frac{\partial f_2}{\partial y} - \frac{\partial f_2}{\partial x}\frac{\partial f_1}{\partial y}$$

Better estimates of x and y than x_0 and y_0 are

$$x_{i+1} = x_i + \Delta x_i \qquad y_{i+1} = y_i + \Delta y_i$$

The solution algorithm is

Step 1. Decide the value of ϵ that $|\Delta x|$ and $|\Delta y|$ must not exceed. Then write equations in the form

$$f_1(x, y) = 0 \qquad f_2(x, y) = 0$$

Step 2. Calculate the residuals using starting estimates $x = x_0$ and $y = y_0$ for first evaluation:

$$r_1 = -f_1(x, y) \qquad r_2 = -f_2(x, y)$$

Step 3. Evaluate the Jacobian:

$$A = \frac{\partial f_1}{\partial x}\frac{\partial f_2}{\partial y} - \frac{\partial f_2}{\partial x}\frac{\partial f_1}{\partial y}$$

Step 4.

$$\Delta x = \frac{1}{A}\left(r_1\frac{\partial f_2}{\partial y} - r_2\frac{\partial f_1}{\partial y}\right)$$

$$\Delta y = \frac{1}{A}\left(r_2\frac{\partial f_1}{\partial x} - r_1\frac{\partial f_2}{\partial x}\right)$$

Step 5. Estimate

$$x \leftarrow x + \Delta x \qquad y \leftarrow y + \Delta y$$

Step 6. If $|\Delta x| < \epsilon$ and $|\Delta y| < \epsilon$, stop; otherwise go to step 2 with the new estimates of x and y.

As an example of the use of the Newton-Raphson method, consider a position analysis of the four-bar linkage depicted in Fig. 5.6, wherein $\rho_1 = 2$ in, $\rho_2 = 1$ in, $\rho_3 = 2.5$ in, and $\rho_4 = 3$ in. For a crank angle of $\theta_2 = 90°$, what are the abscissa angles θ_3 and θ_4? The vector equation

$$\boldsymbol{\rho}_2 + \boldsymbol{\rho}_3 = \boldsymbol{\rho}_1 + \boldsymbol{\rho}_4$$

gives rise to the pair of scalar equations

$$\rho_3 \cos\theta_3 = \rho_4 \cos\theta_4 - \rho_2 \cos\theta_2 + \rho_1$$

$$\rho_3 \sin\theta_3 = \rho_4 \sin\theta_4 - \rho_2 \sin\theta_2$$

Rewrite the equations in the form $f_1(\theta_3, \theta_4) = 0$ and $f_2(\theta_3, \theta_4) = 0$:

$$f_1(\theta_3, \theta_4) = \rho_4 \cos\theta_4 - \rho_2 \cos\theta_2 + \rho_1 - \rho_3 \cos\theta_3$$

$$f_2(\theta_3, \theta_4) = \rho_4 \sin\theta_4 - \rho_2 \sin\theta_2 - \rho_3 \sin\theta_3$$

The residuals are

$$r_1 = -f_1 = -\rho_4 \cos\theta_4 + \rho_2 \cos\theta_2 - \rho_1 + \rho_3 \cos\theta_3$$

$$r_2 = -f_2 = -\rho_4 \sin\theta_4 + \rho_2 \sin\theta_2 + \rho_3 \sin\theta_3$$

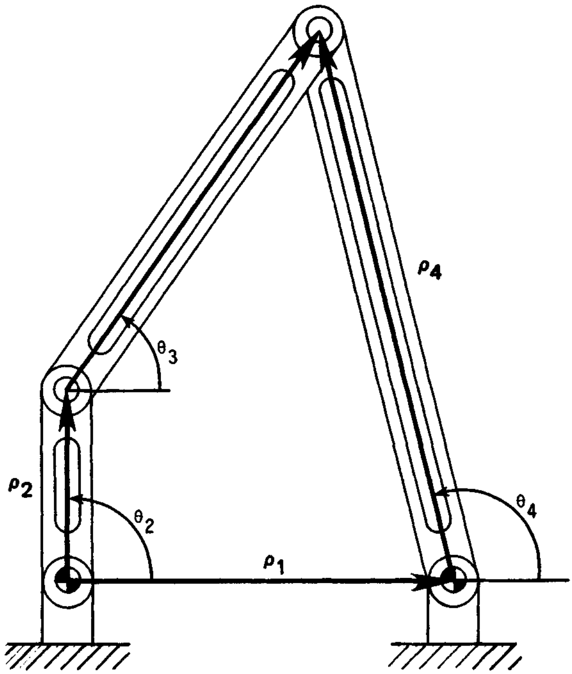

FIGURE 5.6 A vector model of a four-bar linkage with link 1 grounded, link 2 as a crank, link 3 as a coupler, and link 4 as a follower.

The value of A is determined:

$$\frac{\partial f_1}{\partial \theta_3} = \rho_3 \sin \theta_3 \qquad \frac{\partial f_1}{\partial \theta_4} = -\rho_4 \sin \theta_4$$

$$\frac{\partial f_2}{\partial \theta_3} = -\rho_3 \cos \theta_3 \qquad \frac{\partial f_2}{\partial \theta_4} = \rho_4 \cos \theta_4$$

$$A = (\rho_3 \sin \theta_3)(\rho_4 \cos \theta_4) - (-\rho_3 \cos \theta_3)(-\rho_4 \sin \theta_4)$$

$$= \rho_3\rho_4 \sin (\theta_3 - \theta_4)$$

Find $\Delta\theta_3$ and $\Delta\theta_4$:

$$\Delta\theta_3 = \frac{1}{A} (r_1\rho_4 \cos \theta_4 + r_2\rho_4 \sin \theta_4)$$

$$= \frac{\rho_4}{A} (r_1 \cos \theta_4 + r_2 \sin \theta_4)$$

$$\Delta\theta_4 = \frac{1}{A} [r_2\rho_3 \sin \theta_3 - r_1(-\rho_3 \cos \theta_3)]$$

$$= \frac{\rho_3}{A} (r_2 \sin \theta_3 + r_1 \cos \theta_3)$$

Improve the estimate of θ_3 and θ_4:

$$\theta_3 \leftarrow \theta_3 + \Delta\theta_3 \qquad \theta_4 \leftarrow \theta_4 + \Delta\theta_4$$

For initial values $(\theta_3)_0 = 0.8$ and $(\theta_4)_0 = 1.6$, we obtain in four iterations

θ_3	θ_4
0.800 000	1.600 000
0.911 666	1.723 684
0.904 495	1.722 968
0.904 519	1.722 977
0.904 519	1.722 977

converging on $\theta_3 = 0.904\,519$ rad or 51.83° and $\theta_4 = 1.722\,977$ rad or 98.72°. For initial values $(\theta_3)_0 = 5$ and $(\theta_4)_0 = 4$, we obtain in five iterations

θ_3	θ_4
5.000 000	4.000 000
4.477 696	3.743 041
4.440 784	3.626 671
4.451 343	3.632 940
4.451 371	3.632 893
4.451 371	3.632 893

identifying θ_3 as 255.0° and θ_4 as 208.1°, which represents the configuration where the coupler crosses the grounded link.

Such a solution algorithm for simultaneous equations can be generalized to n equations. The previous kinematic problem can be coded for a hand-held calculator in approximately a hundred steps.

Another recurring task is that of integration. A powerful numerical tool is Simpson's first rule:

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}\,[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f''''(\xi) \qquad x_0 \le \xi \le x_2$$

The error term is exact for some (generally unavailable) value of ξ. If the number of repetitions of this rule made in an interval a, b is n, then the number of panels is $N = 2n$. Richardson (Ref. [5.4], p. 78) showed that if an integration is performed in interval a, b with N_2 panels and then repeated with $N_1 = N_2/2$ panels (using every other ordinate), then the value of the integral is given by

$$I = I_{N_2} + \frac{I_{N_2} - I_{N_1}}{15} = I_{N_2} + R_{N_2}$$

where the last term is called *Richardson's error estimate.* The number of panels N_2 must be divisible by 4. The approximate relation between the number of panels and the error is

$$N_j \doteq N_i \left|\frac{E_i}{E_j}\right|^{1/4} = N_i \left|\frac{I_{N_2} - I_{N_1}}{15E_j}\right|^{1/4} = N_i \left|\frac{R_i}{E_j}\right|^{1/4}$$

Solved Problem. Estimate the value of $\int_0^\pi \sin x\,dx$ to five significant digits to the right of the decimal point.

Step 1. Perform the integration with 2 panels, obtaining $I_2 = 2.0944$.

Step 2. Perform the integration with 4 panels, obtaining $I_4 = 2.00456$.

Step 3. Estimate the error in I_4 as

$$R_i = \frac{I_4 - I_2}{15} = \frac{2.004\,56 - 2.0944}{15} = -0.005\,989$$

Step 4. Estimate the number of panels necessary to attain requisite accuracy:

$$N_j \doteq N_i \left|\frac{R_i}{E_j}\right|^{1/4} = 4\left|\frac{0.005\,989}{0.5 \times 10^{-5}}\right|^{1/4}$$

$$= 23.5$$

say, 24 panels.

Step 5. Integrate using 24 panels, obtaining $I_{24} = 2.000\,003\,269$.

This result is high, as indicated by the sign of Richardson's correction and by an estimated amount of $-0.000\,003\,290$. Note that the objective has been achieved. An improved estimate of the value of the integral might be $I_{24} + R_{24} = 1.999\,999\,979$, which rounded to five significant digits to the right of the decimal point is still 2.000 00.

Solved Problem. An electric motor has a torque-rpm characteristic of $36(1 - n/1800)$ ft · lbf and a moment of inertia of 1 slug · ft². Estimate within a tenth of a second the time to come up to a speed of 1600 rpm from rest in the absence of load.

The expression for the time estimate is in the form of an integral:

$$t = \int \frac{I}{T}\,d\omega = \frac{2\pi}{60}\int_0^{1600} \frac{dn}{36(1 - n/1800)}$$

Integrating with four panels using Simpson's rule, the Richardson correction is $R_4 = -0.094\,561\,754$. The estimated number of panels to assess the starting time to within a tenth of a second ($E = 0.05$ s) is

$$N_j \doteq N_i \left|\frac{R_i}{E_j}\right|^{1/4} = 4\left|\frac{0.094\,561\,754}{0.05}\right|^{1/4} = 4.69$$

This is rounded to the next larger integer divisible by 4. Using eight panels:

$$I = I_8 + R_8 = 11.571\,809\,98 - 0.023\,455\,747$$

$$= 11.548\,354\,23$$

The result shows the objective achieved with a result of 11.5 s. Simpson's first rule should be coded and available to any user of a computer.

In computer-aided engineering, a number of routine mathematical tasks are encountered, and these should be available to the programmer in an executive fashion, discharged, if possible, by a one-line call statement (in Fortran).

5.5 STATISTICAL TASKS

There are innumerable statistical tasks to be performed incidental to engineering calculations; for example:

- Descriptive statistics such as means, medians, variances, and ranks have to be developed from data.
- Probabilities of observations from binomial, hypergeometric, Poisson, normal, lognormal, exponential, and Weibullian distributions need to be found.
- Inferential statistics must be developed for distributional parameters such as means, variances, and proportions.
- Data need to be fitted to distributional curves using least-squares lines, polynomials, or distributional functions.
- Goodness-of-fit tests for conformity must be made.

There exist programs for large computational machines which can be imitated or approximated on smaller machines. An important thing to be remembered concerning statistical computations conducted with paper and pencil as compared to those conducted with a computer is that computer programs are executed out of sight and supervision of a human and there is no experienced eye monitoring intermediate results and exhibiting a healthy skepticism when the occasion warrants.

It is so easy to calculate a correlation coefficient, be impressed with its nearness to unity, have it indicate statistical significance of fit, and be wrong without a warning signal. The correlation coefficient has meaning *only* if the data fall randomly about the regression curve. When using the computer, it is important to inspect a graphic presentation or to conduct a run test prior to testing for significance of fit. This test detects randomness or the lack of it in the case of dichotomous events (heads or tails, larger or smaller than the mean, above or below a regression curve). We can observe successes (above), failures (below), and runs (sequences of successes or failures). If n_1 is the number of successes, n_2 is the number of failures, and $n_1 \geq 10$ and $n_2 \geq 10$, then the sampling distribution of the number of runs n is approximately gaussian, with (Ref. [5.17], p. 414)

$$\mu_n = \frac{2n_1n_2}{n_1 + n_2} + 1$$

$$\sigma_n = \left[\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1 + n_2)^2(n_1 + n_2 - 1)}\right]^{1/2}$$

The null hypothesis that the sample is random can be based on the statistic

$$z = \begin{cases} \dfrac{n + \frac{1}{2} - \mu_n}{\sigma_n} & n > \mu_n \\[2ex] \dfrac{n - \frac{1}{2} - \mu_n}{\sigma_n} & n < \mu_n \end{cases}$$

where the ½ improves the gaussian continuous fit to the discrete PDF of n.

If a regression line were determined from data, using a for above the line and b for below the line, then as we move along the abscissa we would observe

$$\underline{aa}\ \underline{b}\ \underline{a}\ \underline{b}\ \underline{a}\ \underline{b}\ \underline{a}\ \underline{b}\ \underline{aa}\ \underline{b}\ \underline{a}\ \underline{b}\ \underline{aa}\ \underline{bb}\ \underline{aa}\ \underline{b}\ \underline{aa}\ \underline{bb}\ \underline{a}\ \underline{b}\ \underline{a}\ \underline{bb}\ \underline{a}\ \underline{b}\ \underline{a}\ \underline{b}\ \underline{a}$$

detecting $n = 27$ runs, $n_1 = 19$ above, $n_2 = 16$ below. The mean number of runs expected μ_n and the standard deviation expected σ_n are

$$\mu_n = \frac{2(19)16}{19+16} + 1 = 18.37$$

$$\sigma_n = \left[\frac{2(19)16[2(19)16 - 19 - 16]}{(19+16)^2(19+16-1)}\right]^{1/2} = 2.89$$

Since the number of runs, that is, 27, is greater than the mean of 18.37,

$$z = \frac{27 + \frac{1}{2} - 18.37}{2.89} = 3.16$$

If the null hypothesis is H_0: runs random, then under H_0, $z = 3.16$, and z (tabulated two-tailed) = 1.96, and we can reject H_0 at 0.95 confidence level and embrace the alternative that the runs are not random. Similarly, if a straight line is fitted to parabolic data, the number of runs might be 3 or 4 or 5. If it is as much as 9, then

$$z = \frac{9 - \frac{1}{2} - 18.37}{2.89} = -3.42$$

and we can still reject randomness at the 0.95 confidence level. Inasmuch as no one is looking, it is important that we build in sentinels such as this.

If the differences between y and $\hat{y}$ are ranked in the order of their corresponding abscissas and placed in the column vector *DY*, then the number of runs can be detected with the following Fortran coding:

```
      NUMBER=1
      DO 100 I=2,N
      A=DY(I)/ABS(DY(I))
      B=DY(I-1)/ABS(DY(I-1))
      IF(A/B.LT.0.)NUMBER=NUMBER+1
100   CONTINUE
```

where the integer NUMBER has a magnitude equal to the number of runs.

5.6 OPTIMIZATION TASKS

The structure of the design-decision problem and that of the optimization problem are similar. Many ideas and techniques of the latter are applicable to the former. The optimization problem can be posed as

$$\text{Maximize (or minimize) } M(x_1, x_2, \ldots, x_n)$$

subject to

$$g_1(x_1, x_2, \ldots, x_n) = 0$$

$$g_2(x_1, x_2, \ldots, x_n) = 0$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$g_m(x_1, x_2, \ldots, x_n) = 0$$

and

$$z_1 \leq f_1(x_1, x_2, \ldots, x_n) \leq Z_1$$
$$z_2 \leq f_2(x_1, x_2, \ldots, x_n) \leq Z_2$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$z_\lambda \leq f_\lambda(x_1, x_2, \ldots, x_n) \leq Z_\lambda$$

The functions $g_i\{x_n\} = 0$ are called *equality* or *functional constraints.* The functions $z_i \leq f_i\{x_n\} \leq Z_i$ are called *inequality* or *regional constraints.* The set $\{x_n\}$ is called the *decision set.* In terms of the ideas in Sec. 5.2, the *specification set* consists of

$$P_1, P_2, \ldots, P_k$$

which for a helical-coil compression spring can consist of (1) material and its condition, that is, A, m, G, E, (2) wire size d, (3) end treatment, that is, Q, Q', (4) total number of turns T, (5) coil outside diameter OD, and (6) free length ℓ_o. The adequacy assessment can be performed by a Fortran subroutine:

```
ADEQ(A,m,G,E,d,Q,Q',T,OD,d,lo,J)
```

where J is returned as $\neq 0$ if inadequate and as 0 if adequate. If the a priori decisions are (1) material and condition, (2) end treatment, (3) total turns, (4) coil outside diameter, and (5) free length, and the wire diameter is chosen as the sole decision variable, then the tasks are

1. Choose d. (This completes the decision set.)
2. Call CONVERT. (Change the decision set into the equivalent specification set.)
3. Call ADEQ. (Establish the adequacy of the decision set.)
4. Call FOM. (Evaluate the figure of merit if the decision set is adequate.)

The choice of d is provided either manually (interactively) or by an appropriate optimization algorithm which makes successive choices of d which have superior merit. The program CONVERT might be problem-specific and need to be created for each type of design problem. The program ADEQ is durable and once programmed can be used. The program FOM is durable as long as the merit criterion (say spring cost) is unchanged.

Figure 5.7 shows the interrelationships of programs FOM, CONVERT, ADEQ, OPT, and the executive program for the helical-spring example.

Optimization programs have to be chosen with care because highly constrained problems can defeat classical strategies. The issue is further complicated by the mixture of discrete and continuous variables. In a spring design, the wire size, end treatment, and material parameters are discrete, whereas the other variables are usually continuous. The user is solving a problem to which the answer is not known and cannot be sure of having attained the global extreme of the figure-of-merit function. Since multidimensional optimization strategies involve some gradient sensitivity, it is judicious to ensure that discrete variables in a problem are among the a priori decisions and the decision set and not buried within. Sometimes an exhaustive search over discrete variables, although computationally inelegant, will attain a global maximum with the least expenditure of computer plus engineering costs. For example, in a spring design with the wire size d as the sole decision variable, marching through the discrete preferred (or available) wire diameters will solve the problem efficiently:

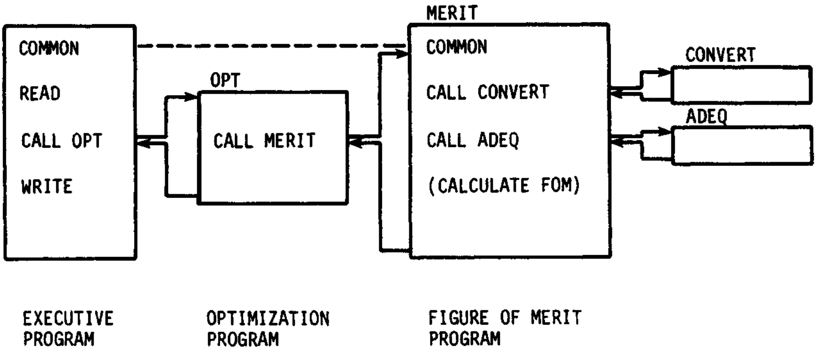

FIGURE 5.7 Organization and subordination of programs for the helical-coil compression spring example.

Washburn and Moen gage: No. 40, 39, 38, 37, . . .
Decimal inch preferred: 0.004, 0.005, 0.006, 0.008, . . .
Decimal millimeter preferred: 0.1, 0.12, 0.16, 0.20, 0.25, . . .

Suppose two decision variables remain after four a priori decisions, and these are wire diameter d and free length ℓ_o (d is discrete and ℓ_o is continuous). The steps might be

1. Enter d and bounds on ℓ_o.
2. For an available wire size d, use a do-loop to show 11 springs of different lengths, displaying d, ℓ_o, FOM, and NG.
3. If there exists a feasible range, enter the ℓ_o bounds which define the range.
4. Use a golden-section search strategy (Ref. [5.8]) on ℓ_o to find the maximum figure of merit.
5. Display the specification set and figure of merit.
6. Repeat for all possible wire sizes.
7. Select best of field as global optimum.

This procedure is best made interactive and can be presented to a user without requiring a knowledge of programming.

For the static-service helical-coil compression spring using one decision variable d and two decision variables d and ℓ_o, the specification sets are

	Case 1. One decision variable d	Case 2. Two decision variables d and ℓ_o
Material and condition	Cold-drawn spring wire 1066	Cold-drawn 1066
Wire size	W&M no. 11 (0.1205 in)	W&M no. 11 (0.1205 in)
End condition	Squared and ground	Squared and ground
Total turns	14.4	13.4
OD	1.091 in	1.091 in
Free length	3.25 in	3.19 in
(Wire volume)	(0.502 in^3)	(0.467 in^3)

Note that both springs are optimal for the conditions, but case 2 is a superior spring in that less material, and therefore less cost, is involved.

Reference [5.1] is a comprehensive introduction to the optimization problem with many examples.

5.7 SIMULATION

Statistics is the science of empiricism. Armed with rationales from probability theory, statistics developed methods for gathering, analyzing, and summarizing data and formulating inferences to learn of systematic relationships, together with an estimate of the chance of being incorrect. Drawing balls from an urn to learn about its contents is a statistical experiment, the balls simulating a universe (a distribution) and those withdrawn constituting a sample. To simulate is to mimic some or all of the behavior of one system with another, with equipment, or with a computer using random numbers.

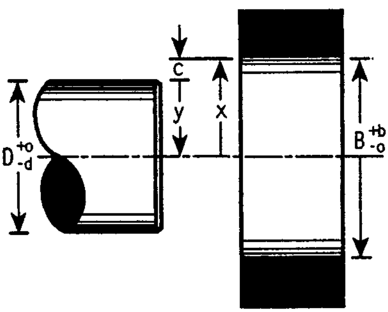

FIGURE 5.8 Dimensions and unilateral tolerances on a journal and bushing. In a specific bearing assembly the bushing radius x, the journal radius y, and the radial clearance c are signed variates related as shown.

Random numbers from the uniform distribution $U[0, 1]$ can be selected using a machine-specific subprogram supplied by the computer manufacturer. These numbers can be transformed into another distribution of interest using software. Through selection of random numbers and calculations performed with them, data can be gathered and answers to useful questions obtained. Consider the unilateral tolerances on a journal and a bushing as depicted in Fig. 5.8. If the journal is generated in a turning or grinding operation, each successive part created on automatic machinery is slightly larger as a result of tool wear and dulling, which increases tool forces and workpiece deflection. The distribution of journal diameter produced between setups and thoroughly mixed is uniform. The reamed bushing bores are uniform for similar reasons. The radii $\mathbf{x}$ and $\mathbf{y}$ and the radial clearance $\mathbf{c}$ are random variables related through the equation $\mathbf{x} - \mathbf{c} - \mathbf{y} = \mathbf{0}$ or

$$\mathbf{c} = \mathbf{x} - \mathbf{y}$$

What is the distribution of the radial clearance $\mathbf{c}$? What is its cumulative distribution function $F(c)$? It takes more than elementary statistics to go straight to the answers. A simulation can be run to obtain robust answers without knowing or using the statistical knowledge.

Example 2. The bearing formed by the journal and bushing of Fig. 5.8 has $D = 2.002$ in, $d = 0.002$ in, $B = 2.004$ in, and $b = 0.003$ in. Estimate the probability that the random assembly radial clearance c is less than or equal to 0.0015 in.

Solution. Given that your computer has a uniform random number generator $u \sim U[0, 1]$ subroutine named RANDU, set up a do-loop which calls RANDU twice, returning u_1 and u_2, and create instances of

$$\vdots$$

$$x = (B + bu_1)/2$$

$$y = (D - d + du_2)/2$$

$$c = x - y$$

$$\vdots$$

Note whether c is less than or equal to 0.0015 in. If the do-loop is executed n times, then the probability of encountering radial clearances less than or equal to 0.0015 is $p = n_1/n$, where n_1 is the number of instances of $c \leq 0.0015$ in. In one million trials, $p = 0.083\,395$. This was accomplished without any special statistical knowledge.

How close is the simulation answer to the correct value? This question will be answered later. Example 2 illustrates the power of the computer simulation process. The price of not knowing the closed-form solution is that the simulation must be repeated to answer another question, say $p(c \leq 0.0017)$. If the program is adjusted to run the simulation to an interactively specified number of trials beginning with the same seed(s) (the same list of random numbers as far as it is needed), one can gather data for the plot of Fig. 5.9. Note the poor estimation associated with a small number of trials, and the eventual approach to an asymptote. The cumulative distribution function can be well approximated for various values of radial clearance c and a polynomial fitted to the data.

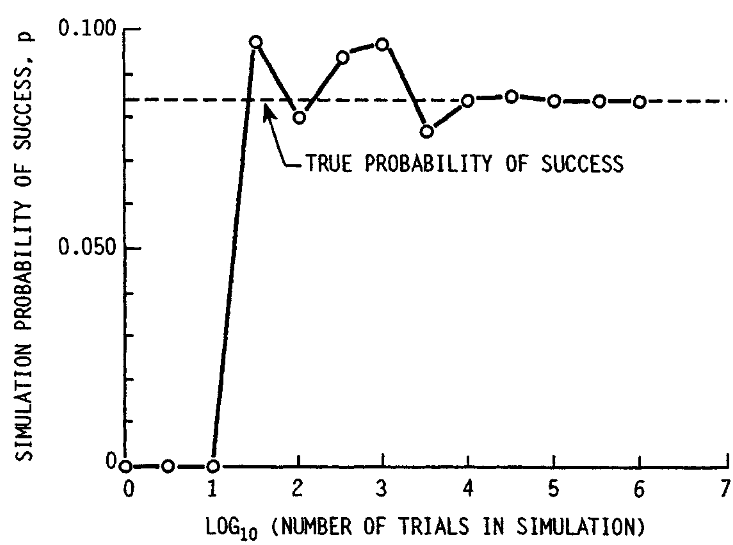

FIGURE 5.9 Simulation convergence in Example 2.

Example 3. For the assembly of Example 2, find an expression for the cumulative distribution function $F(c)$ in the range $0.001 \le c \le 0.002$.

Solution. For simulations of one million trials, the following data were obtained:

c, in	$F(c)$
0.0010	0
0.0012	0.013 308
0.0014	0.053 201
0.0016	0.120 140
0.0018	0.213 820
0.0020	0.333 989

A least-squares quadratic fit of the form $F(c) = a_0 + a_1c + a_2c^2$ gives

$$F(c) = 0.335\,084 - 669.672c + 334577c^2$$

Using this equation to predict the answer to Example 2 yields

$$F(0.0015) = 0.335\,084 - 669.672(0.0015) + 334\,577(0.0015)^2$$
$$= 0.083\,374$$

It is useful to have subroutines to generate random numbers from other useful distributions using RANDU. For a uniform distribution in the interval a, b,

```
SUBROUTINE UNIF1(IX,IY,A,B,U)
CALL RANDU(IX,IY,R)
U=A+(B-A)*R
RETURN
END
```

For a mean of $\bar{x}$, coded XBAR, and a standard deviation of s_x, coded SX, a uniform random number U is returned by

```
SUBROUTINE UNIF2(IX,IY,XBAR,SX,U)
CALL RANDU(IX,IY,R)
A=XBAR-SQRT(3.)*SX
B=XBAR+SQRT(3.)*SX
U=A+(B-A)*R
RETURN
END
```

For a mean of $\bar{x}$, coded XBAR, and a standard deviation s_x, coded SX, a normally distributed random number G is returned by

```
      SUBROUTINE GAUSS(IX,IY,XBAR,SX,G)
      SUM=0.
      DO 100 I=1,12
      CALL RANDU(IX,IY,U)
      SUM=SUM+U
100   CONTINUE
      G=XBAR+(SUM-6.)*SX
      RETURN
      END
```

For a mean of $\bar{x}$, coded XBAR, and a standard deviation of s_x, coded SX, a lognormally distributed random number XLOG is returned by

```
SUBROUTINE LOGNOR(IX,IY,XBAR,SX,XLOG)
CX=SX/XBAR
YBAR=ALOG(XBAR)-ALOG(SQRT(1.+CX**2))
SY=SQRT(ALOG(1.+CX**2))
CALL GAUSS(IX,IY,YBAR,SY,G)
XLOG=EXP(G)
RETURN
END
```

For Weibull parameters x_0, θ, and b, a Weibull-distributed random number W is returned by

```
SUBROUTINE WEIBUL(IX,IY,XO,THETA,B,W)
CALL RANDU(IX,IY,U)
W=XO+(THETA-XO)*(ALOG(1./U))**(1./B)
RETURN
END
```

For distributions with survival equations that can be explicitly solved for R or F, solve for variate x and substitute random numbers $u \sim U[0, 1]$ for either R or F, and x will be random in that distribution.

Example 4. Interfere a Weibull-distributed strength $\mathbf{S} \sim W[40, 50, 3.3]$ kpsi with a Weibull-distributed stress $\sigma \sim W[30, 40, 2]$ kpsi by simulation.

Solution. The body of a Fortran program using the Weibull random number generator WEIBUL is

```
          ⋮
    NSUCC=0
    DO 100 I=1,N
    CALL WEIBUL(IX,IY,XO1,THETA1,B1,W1)
    CALL WEIBUL(IX,IY,XO2,THETA2,B2,W2)
    IF(W1.GE.W2)NSUCC=NSUCC+1
100 CONTINUE
    R=FLOAT(NSUCC)/FLOAT(N)
          ⋮
```

For one million trials, the reliability estimate was 0.956 815.

The question of the accuracy of the reliability estimate is addressed as follows. An interference reliability simulation is really the construction of a column vector $\{x\}$ composed of zeros and ones. The sum of the elements in $\{x\}$ is np, where n is the number of entries (trials) and p is the probability of success. The mean of the elements in $\{x\}$ is

$$\bar{x} = \frac{\Sigma x}{n} = \frac{np}{n} = p \tag{5.1}$$

so $\bar{x}$ is an estimator of the probability of success p. The column vector of the squares of the elements in $\{x\}$ is identical to the elements in $\{x\}$. The sum of the squares of the elements in $\{x\}$ is also np. The standard deviation of x is

$$\sigma_x = \sqrt{\frac{\Sigma x^2 - (\Sigma x)^2/n}{n-1}} = \sqrt{\frac{np - n^2p^2/n}{n}} = \sqrt{p(1-p)} \tag{5.2}$$

The standard deviation of the mean $\bar{x}$ is

$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{n}} = \frac{\sqrt{p(1-p)}}{\sqrt{n}} = \sqrt{\frac{p(1-p)}{\sqrt{n}}} \tag{5.3}$$

In a simulation involving n_1 trials, the bilateral tolerance at the α confidence level (two-tailed) $z_\alpha \sigma_{\bar{x}}$ is denoted as error e_1:

$$e_1 = z_\alpha(\sigma_{\bar{x}})_1 = z_\alpha \sqrt{\frac{p_1(1-p_1)}{n_1}} \tag{5.4}$$

The number of trials n_2 necessary to attain an error $e_2 = ½(10^{-m})$ associated with m significant digits to the right of the decimal point is

$$e_2 = ½(10^{-m}) = z_\alpha \sqrt{\frac{p_2(1-p_2)}{n_2}} \tag{5.5}$$

Arguing that $p_1(1-p_1) \doteq p_2(1-p_2)$ allows Eqs. (5.4) and (5.5) to be combined as

$$e_1\sqrt{n_1} = ½(10^{-m})\sqrt{n_2}$$

or

$$n_2 = n_1\left[\frac{e_1}{½(10^{-m})}\right]^2 = 4n_1(10^{2m})z_\alpha^2(\sigma_{\bar{x}})_1^2 \tag{5.6}$$

Using Eq. (5.3),

$$n_2 = 4(10^{2m})z_\alpha^2\hat{p}(1-\hat{p}) \tag{5.7}$$

Solving Eq. (5.7) for m gives

$$m = ½ \log \frac{n_2}{4z_\alpha^2\hat{p}(1-\hat{p})} \tag{5.8}$$

In Example 4, $p = 0.956\,815$ in 10^6 trials. The digits 0.956 815 were in the computer arithmetic register. Some of the left-hand digits are correct. Using Eq. (5.8),

$$m = ½ \log \frac{10^6}{4(1.96)^2(0.956\,815)(1-0.956\,815)} = 3.1$$

At the 0.95 confidence level, the left two digits are correct; the third may be correct or rounded.

Alternatively, one may use Eq. (5.5) to find the bilateral tolerance:

$$e_2 = 1.96\sqrt{\frac{0.956\,815(1-0.956\,815)}{10^6}} = 0.000\,398$$

and display $p = 0.956\,815 \pm 0.000\,398$ at the 0.95 confidence level.

There is an advantage to writing a computer code to start a string of random numbers from a seed or seeds, then proceeding with the simulation in steps.

- Using a few thousand trials, find n_1 and p_1, and from Eq. (5.4) estimate e_1.
- Use Eq. (5.6) to estimate n_2. The number of additional trials needed to reach the accuracy goal is $n_2 - n_1$. Conduct the additional trials.

- Using n_2 and p_2, find m using Eq. (5.8) or find e_2 from Eq. (5.5). Use Eq. (5.7) and n_2 results as a check.

The net effect of this plan is that no more trials than necessary are performed. In small problems this is a minor consideration, but if a single trial has consequential cost, economy of effort is important.

A Fortran program for solving Example 4 using incremental trials, rather than ever-larger simulations until the goal is reached, follows.

```
c Program wsim.f
c Simulation program for Weibull-Weibull interference C. Mischke Nov 93
    1 print*,'Simulation program Weibull-Weibull interference,Mischke'
      print*,' '
      print*,'Enter RANDU seeds ix, iy (odd, five digits or more)'
      read*,ix,iy
c Initialize counters so simulation proceeds in steps (economically)
c under the control of the user.
      sum1=0.
      sum2=0.
      nused=0
c Enter distribution parameters
      print*,'Enter strength Weibull parameters x0, theta, b'
      read*,x01,theta1,b1
      print*,'Enter stress Weibull parameters x0,theta,b'
      read*,x02,theta2,b2
      print*,'Enter z-variable and corresponding confidence level'
      print*,'for bilateral tolerance on result'
      read*,z,alpha
c Conduct simulation
    2 print*,'Enter number of trials in similation n'
      read*,nmore
      print*,'Enter bilateral tolerance allowable on simulation result'
      read*,error
      do 100 i=1,nmore
      call weibul(ix,iy,x01,theta1,b1,w1)
      call weibul(ix,iy,x02,theta2,b2,w2)
      if(w1.ge.w2)x=1
      if(w1.lt.w2)x=0.
      sum1=sum1+x
      sum2=sum2+x*x
  100 continue
      nused=nmore+nused
      p=sum1/float(nused)
      sigmap=sqrt((sum2-sum1**2/float(nused))/float(nused-1))
      sigmamu=sigmap/sqrt(float(nused))
      tol=z*sigmamu
      print*,'For',nused,' trials probability of success p is',p
      print*,'sigmap=',sigmap,' sigmapbar=',sigmamu,' bilat. tol.=',tol
      print*,'Largest allowable bilateral error is',error
      xn2= float(nused)*tol**2/error**2
      if(nused.gt.xn2)go to 10
      print*,'Total trials needed =',xn2,' or ',xn2-float(nused),' more'
      go to 11
   10 print*,'Simulation complete in',nused,' trials'
      correct=alog10(1./2./tol)
      print*,'Correct digits to right of decimal in p is',correct
      print*,'at',alpha,' confidence level.'
      ncorrect=correct
      np=p*10.**ncorrect
      p=float(np)/10.**ncorrect
      print*,'Correct digits are',p,' at',alpha,' confidence level'
   11 print*,' '
      print*,'For a new problem,      enter 1'
      print*,'To continue simulation, enter 2'
      print*,'To quit                 enter 3'
      read*,index
      go to(1,2,3),index
    3 call exit
      end
```

REFERENCES

5.1 G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization, Methods and Applications,* John Wiley & Sons, New York, 1983.

5.2 *Design Handbook,* Associated Spring-Barnes Group, Bristol, Conn., 1981.

5.3 J. E. Shigley and C. R. Mischke, *Mechanical Engineering Design,* 5th ed., McGraw-Hill, New York, 1989.

5.4 B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods,* John Wiley & Sons, New York, 1969, p. 168.

5.5 R. C. Weast and S. M. Shelby, eds., *Handbook of Tables for Mathematics,* 3d ed., The Chemical Rubber Company, 1967, p. 579.

5.6 J. E. Shigley and J. J. Uicker, *Theory of Machines and Mechanisms,* McGraw-Hill, New York, 1980, p. 266.

5.7 W. H. Beyer, *Handbook of Tables for Probability and Statistics,* 2d ed., The Chemical Rubber Company, 1968, p. 414.

5.8 C. R. Mischke, *Mathematical Model Building,* 2d rev. ed., Iowa State University Press, Ames, 1980, chap. 3.