NAG Library Chapter Introduction

C05 – Roots of One or More Transcendental Equations

Contents

1 Scope of the Chapter

This chapter is concerned with the calculation of real zeros of continuous real functions of one or more variables. (Complex equations must be expressed in terms of the equivalent larger system of real equations.)

2 Background to the Problems

The chapter divides naturally into two parts.

2.1 A Single Equation

The first deals with the real zeros of a real function of a single variable $f(x)$.

There are three routines with simple calling sequences. The first assumes that you can determine an initial interval $[a, b]$ within which the desired zero lies, (that is, where $f(a) \times f(b) < 0$), and outside which all other zeros lie. The routine then systematically subdivides the interval to produce a final interval containing the zero. This final interval has a length bounded by your specified error requirements; the end of the interval where the function has smallest magnitude is returned as the zero. This routine is guaranteed to converge to a **simple** zero of the function. (Here we define a simple zero as a zero corresponding to a sign-change of the function; none of the available routines are capable of making any finer distinction.) However, as with the other routines described below, a non-simple zero might be determined and it is left to you to check for this. The algorithm used is due to Brent (1973).

The two other routines are both designed for the case where you are unable to specify an interval containing the simple zero. The first routine starts from an initial point and performs a search for an interval containing a simple zero. If such an interval is computed then the method described above is used next to determine the zero accurately. The second method uses a ‘continuation’ method based on a secant iteration. A sequence of subproblems is solved; the first of these is trivial and the last is the actual problem of finding a zero of $f(x)$. The intermediate problems employ the solutions of earlier problems to provide initial guesses for the secant iterations used to calculate their solutions.

Three other routines are also supplied. They employ reverse communication and use the same core algorithms as the routines described above.

Finally, a routine is provided that uses the iterative method described in Barry *et al.* (1995) to return values from the real branches of Lambert’s W function (sometimes known as the ‘product log’ or ‘Omega’ function), which is the inverse function of

$$f(w) = we^w \qquad \text{for} \qquad w \in C;$$

that is, if Lambert’s W function $W(x) = a$ for $x, a \in C$, then a is a zero of the function $F(w) = we^w - x$. In this chapter we restrict $x, a \in R$.

2.2 Systems of Equations

The routines in the second part of this chapter are designed to solve a set of nonlinear equations in n unknowns

$$f_i(x) = 0, \qquad i = 1, 2, \ldots, n, \qquad x = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}, \tag{1}$$

where T stands for transpose.

It is assumed that the functions are continuous and differentiable so that the matrix of first partial derivatives of the functions, the **Jacobian** matrix $J_{ij}(x) = \left(\frac{\partial f_i}{\partial x_j}\right)$ evaluated at the point x, exists, though it may not be possible to calculate it directly.

The functions f_i must be independent, otherwise there will be an infinity of solutions and the methods will fail. However, even when the functions are independent the solutions may not be unique. Since the methods are iterative, an initial guess at the solution has to be supplied, and the solution located will usually be the one closest to this initial guess.

3 Recommendations on Choice and Use of Available Routines

3.1 Thread Safe Routines

Some of the routines in this chapter come as pairs, with each routine in the pair having exactly the same functionality, except that one of them has additional parameters in order to make it safe for use in multithreaded applications. The routine that is safe for use in multithreaded applicatons has an 'A' as the last character in the name, in place of the usual 'F'.

In this chapter there are three such pairs (C05PBF/C05PBA, C05PCF/C05PCA and C05PDF/C05PDA).

All 'F' routines not scheduled for withdrawal from the Library and where there is no 'A' version of that routine are threadsafe provided that the implementation as a whole is considered threadsafe (refer to the Users' Note for your implementation).

3.2 Zeros of Functions of One Variable

The routines can be divided into two classes. There are three routines (C05AVF, C05AXF and C05AZF) all written in reverse communication form and three (C05ADF, C05AGF and C05AJF) written in direct communication form. The direct communication routines are designed for inexperienced users and, in particular, for solving problems where the function $f(x)$ whose zero is to be calculated, can be coded as a user-supplied (sub)program. These routines find the zero by using the same core algorithms as the reverse communication routines. Experienced users are recommended to use the reverse communication routines directly as they permit you more control of the calculation. Indeed, if the zero-finding process is embedded in a much larger program then the reverse communication routines should always be used.

The recommendation as to which routine should be used depends mainly on whether you can supply an interval $[a,b]$ containing the zero; that is, where $f(a) \times f(b) < 0$. If the interval can be supplied, then C05ADF (or, in reverse communication, C05AZF) should be used, in general. This recommendation should be qualified in the case when the only interval which can be supplied is very long relative to your error requirements **and** you can also supply a good approximation to the zero. In this case C05AJF (or, in reverse communication, C05AXF) **may** prove more efficient (though these latter routines will not provide the error bound available from C05AZF).

If an interval containing the zero cannot be supplied then you must choose between C05AGF (or, in reverse communication, C05AVF followed by C05AZF) and C05AJF (or, in reverse communication, C05AXF). C05AGF first determines an interval containing the zero, and then proceeds as in C05ADF; it is particularly recommended when you do not have a good initial approximation to the zero. If a good initial approximation to the zero is available then C05AJF is to be preferred. Since neither of these latter routines has guaranteed convergence to the zero, you are recommended to experiment with both in case of difficulty.

3.3 Solution of Sets of Nonlinear Equations

The solution of a set of nonlinear equations

$$f_i(x_1, x_2, \ldots, x_n) = 0, \qquad i = 1, 2, \ldots, n \tag{2}$$

can be regarded as a special case of the problem of finding a minimum of a sum of squares

$$s(x) = \sum_{i=1}^{m} [f_i(x_1, x_2, \ldots, x_n)]^2, \qquad (m \geq n). \tag{3}$$

So the routines in Chapter E04 are relevant as well as the special nonlinear equations routines.

The routines for solving a set of nonlinear equations can also be divided into classes. There are four routines (C05NBF, C05NCF, C05PBF/C05PBA and C05PCF/C05PCA) all written in direct communication form and two (C05NDF and C05PDF/C05PDA) written in reverse communication form. The direct communication routines are designed for inexperienced users and, in particular, these routines require the f_i (and possibly their derivatives) to be calculated in user-supplied subroutines. These should be set up carefully so the Library routines can work as efficiently as possible. Experienced users are recommended to use the reverse communication routines as they permit you more control of the calculation. Indeed, if the zero-finding process is embedded in a much larger program then the reverse communication routines should always be used.

The main decision you have to make is whether to supply the derivatives $\frac{\partial f_i}{\partial x_j}$. It is advisable to do so if possible, since the results obtained by algorithms which use derivatives are generally more reliable than those obtained by algorithms which do not use derivatives.

C05PBF/C05PBA and C05PCF/C05PCA (or, in reverse communication, C05PDF/C05PDA) require you to provide the derivatives, whilst C05NBF and C05NCF (or, in reverse communication, C05NDF) do not. C05NBF and C05PBF/C05PBA are easy-to-use routines; greater flexibility may be obtained using C05NCF and C05PCF/C05PCA (or, in reverse communication, C05NDF and C05PDF/C05PDA), but these have longer parameter lists. C05ZAF is provided for use in conjunction with C05PBF/C05PBA and C05PCF/C05PCA to check the user-supplied derivatives for consistency with the functions themselves. You are strongly advised to make use of this routine whenever C05PBF/C05PBA or C05PCF/C05PCA is used.

Firstly, the calculation of the functions and their derivatives should be ordered so that **cancellation errors** are avoided. This is particularly important in a routine that uses these quantities to build up estimates of higher derivatives.

Secondly, **scaling** of the variables has a considerable effect on the efficiency of a routine. The problem should be designed so that the elements of x are of similar magnitude. The same comment applies to the functions, i.e., all the f_i should be of comparable size.

The accuracy is usually determined by the accuracy parameters of the routines, but the following points may be useful.

(i) Greater accuracy in the solution may be requested by choosing smaller input values for the accuracy parameters. However, if unreasonable accuracy is demanded, rounding errors may become important and cause a failure.

(ii) Some idea of the accuracies of the x_i may be obtained by monitoring the progress of the routine to see how many figures remain unchanged during the last few iterations.

(iii) An approximation to the error in the solution x is given by e where e is the solution to the set of linear equations

$$J(x)e = -f(x)$$

where $f(x) = (f_1(x), f_2(x), \ldots, f_n(x))^{\mathrm{T}}$.

Note that the QR decomposition of J is available from C05NCF and C05PCF/C05PCA (or, in reverse communication, C05NDF and C05PDF/C05PDA) so that

$$Re = -Q^{\mathrm{T}} f$$

and $Q^{\mathrm{T}} f$ is also provided by these routines.

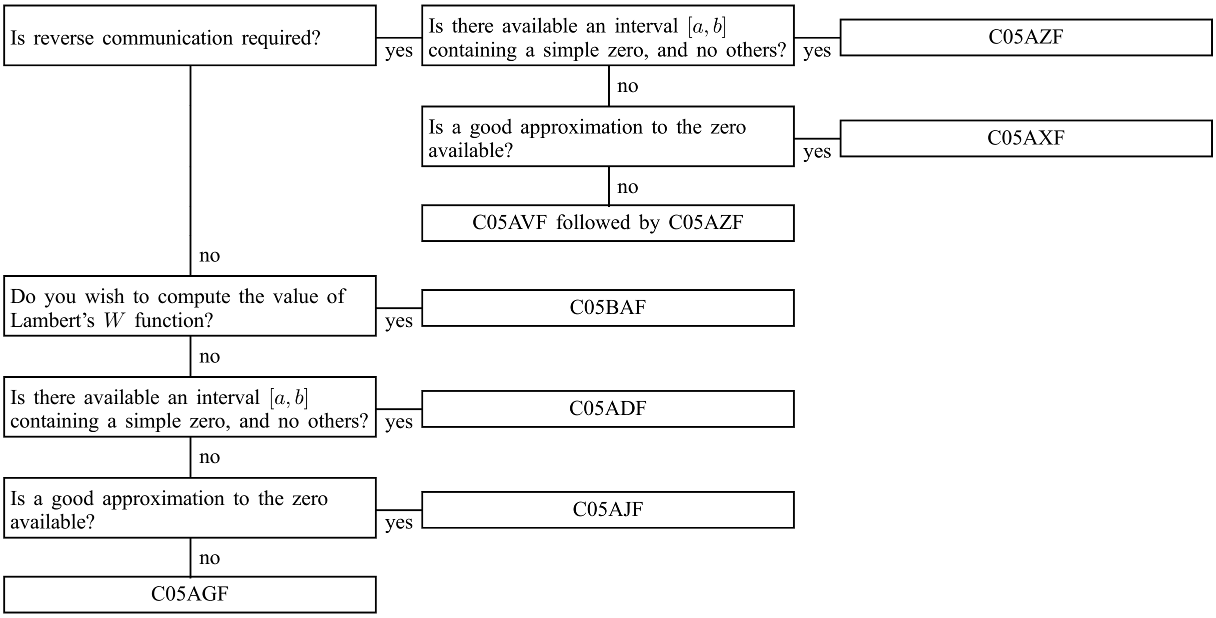
(iv) If the functions $f_i(x)$ are changed by small amounts ϵ_i, for $i = 1, 2, \ldots, n$, then the corresponding change in the solution x is given approximately by σ, where σ is the solution of the set of linear equations

$$J(x)\sigma = -\epsilon.$$

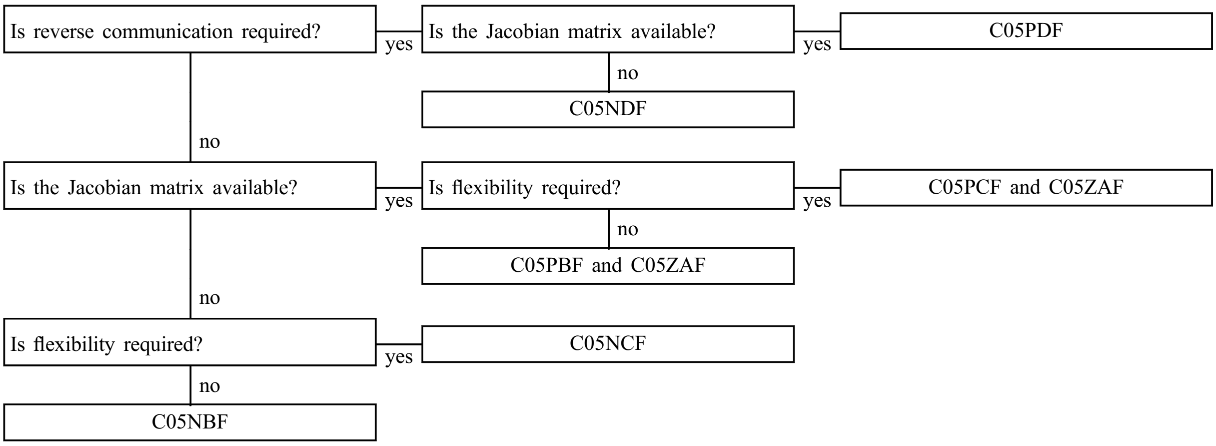
Thus one can estimate the sensitivity of x to any uncertainties in the specification of $f_i(x)$, for $i = 1, 2, \ldots, n$. As noted above, the sophisticated routines C05NCF and C05PCF/C05PCA (or, in reverse communication, C05NDF and C05PDF/C05PDA) provide the QR decomposition of J.

4 Decision Trees

Tree 1: Functions of One Variable



Tree 2: Functions of several variables



5 Index

6 Routines Withdrawn or Scheduled for Withdrawal

None.

7 References

Barry D J, Culligan–Hensley P J, and Barry S J (1995) Real Values of the W-function *ACM Trans. Math. Software* **21 (2)** 161–171

Brent R P (1973) *Algorithms for Minimization Without Derivatives* Prentice–Hall

Gill P E and Murray W (1976) Algorithms for the solution of the nonlinear least-squares problem *Report NAC 71* National Physical Laboratory

Moré J J, Garbow B S and Hillstrom K E (1980) User guide for MINPACK-1 *Technical Report ANL-80-74* Argonne National Laboratory

Ortega J M and Rheinboldt W C (1970) *Iterative Solution of Nonlinear Equations in Several Variables* Academic Press

Rabinowitz P (1970) *Numerical Methods for Nonlinear Algebraic Equations* Gordon and Breach

---