Rational interpolation and extrapolation for SUMT

K. Vit

Department of Electrical Engineering, University of Salford, Salford M5 4WT

Two explicit formulae for rational interpolation are given and compared, with a special reference to a subroutine using only function values for a linear search in the SUMT (Sequential Unconstrained Minimisation Technique) transformation. Different forms of rational functions for extrapolation purposes are also considered.

(Received May 1973)

1. Rational interpolation

By a constrained minimisation will be understood the following problem:

Find a point x^* which locally minimises a scalar function $f(x)$ of a vector argument $x \in R^n$ and satisfies the system of inequalities

$$g_i(x^*) \geqslant 0 \,. \tag{1}$$

It is assumed that such a point exists and $f(x^*)$ is finite. Any point satisfying (1) will be called feasible.

The SUMT method transforms the difficult problem of a constrained minimisation into a sequence of simpler unconstrained problems which, under some conditions, converges to the solution of the original problem. To be more specific, let us consider the interior point algorithm with the inverse function for the penalty term (Fiacco, McCormick, 1968). This is where the rational interpolation in a linear search is likely to be most efficient due to the preservation of the differentiability order of the transformed function and because the region of definition of the inverse barrier function includes also unfeasible points which may be therefore used for locating the minimum inside the feasible region. This is not possible for a polynomial approximation.

Let r be a small positive number and define the transformation

$$F(x, r) = f(x) + r \sum_i \frac{1}{g_i(x)} \,. \tag{2}$$

Then,

$$\min_{g_i(x) \geqslant 0} f(x) = \lim_{r \to 0^+} \min F(x, r)$$

where the last minimisation is unconstrained for a small neighbourhood of the minimising point.

In practice, one chooses a sequence $r_k > r_{k+1} > 0$ and minimises numerically $F(x, r_k)$ to obtain x_k^*. For most procedures this requires the use of an one-dimensional minimisation subroutine, i.e. a linear search along a direction d has to be repeatedly carried out to find a scalar w^* minimising $F(x_0 + wd, r)$ subject to $g_i(x_0 + w^*d) \geqslant 0$.

When $y_j = F(x_j, r)$, $j = 0, 1, 2, \ldots$ have been computed from (2) for different points $x_j = x_0 + w_j d$, $w_0 = 0$, then, given a suitable function $y(w)$, one can use the pairs (w_j, y_j) to estimate both w^* and the tentative value y^* of $y(w^*)$ which is useful for an accuracy check. The simplest and most often used function is a polynomial $y(w) = \sum_{m=0}^{p} a_m w^m$. As well as being rather inefficient for this particular application due to a perverse behaviour of F near the constraints, it requires $g_i(x_j) \geqslant 0$, i.e., only feasible points may be used. Kowalik and Osborne (1968), used a golden-section algorithm and Fox (1971), suggested the following rational form of $y(.)$.

4-point rational approximation

$$y(w) = a_0 + a_1 w + \frac{b}{w - c} \,. \tag{3}$$

To determine four coefficients in (3) one could employ any of the standard algorithms for rational interpolation (Larkin, 1967). The first algorithm due to Stoer (1961) has been programmed in FORTRAN and it was found that it would require, for the present problem, 75 multiplications and 30 additions. This was thought to be a prohibitive computational burden and therefore a direct algebraic solution of the system

$$y(w_j) = y_j \qquad j = 0, 1, 2, 3$$

was carried out. After a lengthy manipulation the corresponding expressions were obtained as given by (4). Some of the

$$c = -w_1 w_2 w_3 \frac{(w_2 - w_3)y_1 + (w_3 - w_1)y_2 + (w_1 - w_2)y_3}{\begin{array}{c}[w_1 w_2(w_1 - w_2) + w_1 w_3(w_3 - w_1) + / \\ w_2 w_3(w_2 - w_3)]y_0 - w_2 w_3(w_2 - w_3)y_1 - / \\ w_1 w_3(w_3 - w_1)y_2 - w_1 w_2(w_1 - w_2)y_3\end{array}} \tag{4a}$$

$$b = -\frac{(w_1 - w_2)y_0 - w_1 y_2 + w_2 y_1}{w_1 w_2(w_1 - w_2)} c(w_1 - c)(w_2 - c) \tag{4b}$$

$$a_1 = \frac{y_1 - y_0}{w_1} - \frac{b}{c(w_1 - c)} \tag{4c}$$

$$w^* = c + \text{sign}(b) \sqrt{\frac{b}{a_1}} \tag{4d}$$

$$y^* = y_0 + a_1 w^* + \frac{w^*}{c(w^* - c)} b \tag{4e}$$

coefficients in equations (4) were precomputed for a few commonly used values of triples w_j, $j = 1, 2, 3$ and stored as constants.

A sign check has to be made before taking the square root in (4d) and, if the number is negative, some emergency action has to be taken (e.g. restart of the linear search from the best feasible point with w_j's reduced).

3-point rational approximation

$$y(w) = u(w) + z(w) \tag{5a}$$

$$u(w) = a_{01} + a_1 w \tag{5b}$$

$$z(w) = a_{02} + \frac{b}{w - c} \,. \tag{5c}$$

The two functions in (2) are approximated independently with $u_j = f(x_0 + w_j d)$ and

$$z_j = r \sum_i \frac{1}{g_i(x_0 + w_j d)} \,, j = 0, 1, 2, w_0 = 0 \,.$$

The function $u(w)$ in (5b) was chosen linear since a quadratic polynomial would lead to a cubic equation in w^*. One possible way of determining a_{01} and a_1 in (5b) is a least-squares fit of $u(w)$ to three points $u_j = u(w_j)$. The approach used here, however, is to fit the three points with a quadratic polynomial and then replace this by its tangent at some point $x_0 + \bar{w}d$ This procedure gives rise to equations (6): $\bar{w}$ is chosen as follows.

$$c = w_1 w_2 \frac{z_2 - z_1}{w_2(z_0 - z_1) - w_1(z_0 - z_2)} \tag{6a}$$

$$b = \frac{c(w_1 - c)}{w_1}(z_1 - z_0) \tag{6b}$$

$$a_1 = \frac{(w_1 - w_2)(w_1 + w_2 - 2\bar{w})u_0 + / \; w_2(w_2 - 2\bar{w})u_1 - w_1(w_1 - 2\bar{w})u_2}{w_1 w_2 (w_2 - w_1)} \tag{6c}$$

$$w^* = c + \operatorname{sign}(b)\sqrt{\frac{b}{a_1}} \tag{6d}$$

$$y^* = u_0 + z_0 + a_1 w^* + \frac{w^*}{c(w^* - c)} b \tag{6e}$$

To nullify the effect of replacing the quadratic polynomial by its tangent one would like to carry out the linearisation about w^*, i.e. to have $\bar{w} = w^*(\bar{w})$. From the special nature of the problem, the iterative solution of equations (6c-d) written as

$$\bar{w}_{l+1} = w^*(\bar{w}_l) \tag{7}$$

may be expected to converge fast for small values of r. The following scheme has been found satisfactory

1. set $\bar{w}_0 = 0$,
2. compute $\bar{w}_1$ and $\bar{w}_2$ using (7),
3. estimate $w^* = \bar{w}_1^2 2/(2\bar{w}_1 - \bar{w}_2)$ by Δ^2-method.

This method gave results superior to the least-squares fit and it was programmed with 23 multiplications, 18 additions and 5 divisions.

2. Comparison of the 4-point and 3-point algorithms

Different minimisation procedures are often compared on the basis of a number of function evaluations necessary to achieve the same accuracy. The 4-point algorithm requires one function evaluation more but it may be expected to approximate the minimised function better. However, for small r's a poor accuracy of the 4-point algorithm was caused by numerical difficulties when extracting information about the constraints from $F(x, r)$ due to the second term in (2) being small. The 3-point algorithm which approximates both parts of (2) separately performed better. To demonstrate this, a simple example was solved on the KDF9 computer (11 significant digits).

Example 1

Let $f(x) = 1 + 10x$ and $g(x) = 1{\cdot}0001 + x$, $r = {}_{10} - 7$, $d \triangleq 1$. Then the function $F(x, r)$ attains its local minimum value $F(x^*, r) = -8{\cdot}999$ at $x^* = -1{\cdot}0$. The results given on **Table 1** were obtained using the derived expressions with $\bar{w} = 0$.

Example 2

The following problem was solved by Kowalik *et al.* (1968), who used a golden-section algorithm for their linear search. Minimise

$$f(x) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to

$$g_1(x) = x_1$$

$$g_2(x) = x_2$$

$$g_3(x) = 3 - x_1 - x_2 - 2x_3$$

The minimum value $f(x^*) = 1/9$ is reached at

$$x^* = [4/3, 7/9, 4/9] .$$

Their results are compared here with three versions of Davies, Swann and Campey minimisation procedure (Swann, 1964) refined by Hoshino (1971) with orthogonal directions generated by the algorithm due to Palmer (1969). The versions differed only by the functional approximation for the linear search. A quadratic interpolation was used normally inside the feasible region, a rational approximation only after a constraint was violated. The results are summarised in **Table 2.** The 3-point

Table 1(*a*) 4-Point algorithm

i	w_i	x_i	y_i
0	0·0	1·0	$0{\cdot}11000000050_{10} + 2$
1	−1·0	0·0	$0{\cdot}10000001000_{10} + 1$
2	1·0	2·0	$0{\cdot}21000000033_{10} + 2$
3	2·0	3·0	$0{\cdot}31000000025_{10} + 2$

$c = -0{\cdot}98839907193$
$x^* = -0{\cdot}98830016286$
$y^* = -0{\cdot}88820125368_{10} + 1 \quad y(x^*) = -0{\cdot}88829931539_{10} + 1$

Table 1(*b*) 3-Point algorithm

i	w_i	x_i	u_i	z_i
0	0·0	1·0	11·0	$0{\cdot}49997500125_{10} - 7$
1	−1·0	0·0	10·0	$0{\cdot}99990001000_{10} - 7$
2	1·0	2·0	21·0	$0{\cdot}33332222259_{10} - 7$

$c = -0{\cdot}10001000000_{10} + 1$
$x^* = -0{\cdot}99999999999$
$y^* = -0{\cdot}89989999999_{10} + 1 \quad y(x^*) = -0{\cdot}89990000000_{10} + 1$

Table 2

		Number of function evaluations			
r	$F(x^*, r)$	*GS*	*P*	*4R*	*3R*
1	7·4168	210	107	128	107
$_{10} - 2$	0·2605	330	95	102	93
$_{10} - 4$	0·1210	510	153	151	179
$_{10} - 6$	0·1128	560	378	286	95
$_{10} - 8$	0·1112		327	*	172
$_{10} - 9$	0·1111		*	*	155

GS golden-section (from Kowalik *et al.*, 1968)
P parabolic approximation
4R 4-point rational approximation
3R 3-point rational approximation
* procedure failed to reach minimum

Table 3

r	$_{10} - 3$	$_{10} - 4$	$_{10} - 5$
$x_1^*(r)$	1·353223	1·340245	1·339885
$x_2^*(r)$	0·762537	0·773118	0·774010
$x_3^*(r)$	0·409701	0·432827	0·439643
$h(r)$	0·145908	0·121024	0·114161
N/D	2/0	1/1	0/2
$x_1^*(0)$	1·339855	1·339848	1·339855
$x_2^*(0)$	0·774107	0·774108	0·774107
$x_3^*(0)$	0·440450	0·440552	0·440463
$h(0)$	0·113349	0·113261	0·113395

rational approximation performs the best as measured by the number of function calls.

The separate computation of both parts of $F(x, r)$ has yet another advantage. Consider a situation where a search is made in a direction d for which $f(x_0 + wd)$ is independent of w. Due to the penalty term in (2), however, $F(x_0 + wd, r)$ may be a decreasing function of w and the search may continue infinitely unless some special precaution is taken. This may happen in the Example 2 if the same procedure is used for maximisation of violated constraints in order to find a starting feasible point. For if g_1 and g_3 are positive, the following function is to be minimised

$$F(x, r) = -x_2 + r\left(\frac{1}{x_1} + \frac{1}{3 - x_1 - x_2 - 2x_3}\right)$$

and provided that the search is made in the direction x_3, then, as $x_3 \to -\infty$ F continues to decrease without g_1 and g_3 being violated. By checking f rather than F in (2) against stationary values the danger of 'getting stuck' on such functions is eliminated.

3. Rational extrapolation

Let

$$\min_{g_i(x) \geqslant 0} F(x, r_k) = F[x^*(r_k), r_k] = h(r_k) \ .$$

Under some conditions, $h(r)$ is known to be a differentiable function near $r = 0$ and, using a suitable functional expansion of $h(\cdot)$, one can estimate $h(0)$ from a few known values of $h(r_k)$, $r_k > 0$. Similarly, the point x^* minimising $f(x)$ subject to $g_i(x) \geqslant 0$ can be estimated by applying this limiting process co-ordinate-wise to a few points $x^*(r_k)$. The accuracy of this estimation $|h(0) - f[x^*(0)]|$ may be used as a stopping criterion for the optimisation process. Fiacco and McCormick (1966, 1968) used successfully the Taylor expansion in $\sqrt{r}$ but further generalisation to the Laurent series expansion, leading to a rational approximation for $h(\cdot)$ and $x^*(.)$, is possible. In spite of lack of a rigorous justification for such a step (Oliver, 1971), Bulirsch and Stoer (1966, 1967) reported a higher numerical efficiency of the rational extrapolation when applied to a similar problem of solution of ordinary differential equations and numerical quadrature.

A series of computer tests was made with the SUMT transformation and, using the 2nd algorithm of Stoer, (1961), all possible rational functions in $\sqrt{r}$ (including polynomials) were fitted to up to five successive data from $F[x^*(r_k), r_k]$ of Example 2 to obtain the limiting values. A surprising consistency of results for all forms of the approximating functions was found. As a sample of these tests, partial results of extrapolation of the function from example 2 are presented on **Table 3.** N and D denote the orders of the polynomials in the numerator and denominator of the rational function, respectively, fitted to $N + D + 1$ successive points.

Some understanding of this behaviour may be obtained from the following simple case. Consider the polynomial $\tilde{y}(\sqrt{r})$ and the rational function $\bar{y}(\sqrt{r})$ fitted to two points

$$y_i = \tilde{y}(\sqrt{r_i}) = \bar{y}(\sqrt{r_i}), \quad i = 1, 2$$

define $\rho = r_2/r_1 \neq 1$. Then

$$\frac{\bar{y}(0)}{\tilde{y}(0)} = \frac{(1 - \rho)^2}{1 - \dfrac{(y_1^2 + y_2^2)}{y_1 y_2}\rho + \rho^2} \ .$$

For small r_i, one should expect the difference $\Delta = y_2 - y_1$ to be small enough to allow for the approximation

$$y_1^2 + y_2^2 = 2y_1 y_2$$

giving $\bar{y}(0) \approx \tilde{y}(0)$.

To conclude, it seems that, because of its relative simplicity, the low-order polynomial for extrapolation purposes in the SUMT transformation using inverse barrier function is the best choice.

References

Bulirsch, R., and Stoer, J. (1966). Numerical treatment of ordinary differential equations by extrapolation method, *Num. Math.*, Vol. 8, pp. 1-13.

Bulirsch, R., and Stoer, J. (1967). Numerical quadrature by extrapolation, *Num. Math.*, Vol. 9, pp. 271-278.

Fiacco, A. V., and McCormick, G. P. (1966). Extension of SUMT for nonlinear programming: Equality constraints and extrapolation, *Management Sci.*, Vol. 12, pp. 816-828.

Fiacco, A. V., and McCormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons.

Fox, R. L. (1971). *Optimization Methods for Engineering Design*, Addison-Wesley Publishing Co., p. 149.

Hoshino, S. (1971). On Davies, Swann and Campey minimisation process, *The Computer Journal*, Vol. 14, No. 4, p. 426.

Kowalik, J., and Osborne, M. R. (1968). *Methods for Unconstrained Optimization Problems*, American Elsevier Publishing Co.

Larkin, F. M. (1967). Some techniques for rational interpolation, *The Computer Journal*, Vol. 10, No. 2, pp. 178-187.

Oliver, J. (1971). The efficiency of extrapolation methods for numerical integration, *Num. Math.*, Vol. 17, pp. 17-32.

Palmer, J. R. (1969). An improved procedure for orthogonalising the search vectors in Rosenbrock's and Swann's direct search optimisation methods, *The Computer Journal*, Vol. 12, No. 1, pp. 69-71.

Stoer, J. (1961). Über zwei Algorithmen zur Interpolation mit rationalen Funktionen, *Num. Math.*, Vol. 3, pp. 285-304.

Swann, W. H. (1964). *Report on the development of a new direct search method of optimisation*, I.C.I. Ltd., Central Instrument Laboratory, Research Note 64/3.