```
max := b[nb];
if max < b[1] then max := 4;
begin array rw, iw, rx, ix[1 : max];
  integer nless1, wtj, j, i, r, gpstep, wg, g, kend, k, jj, kbase,
  rless1;
  real z, wrz, rwrz, iwrz, rwj, iwj, wgz, re, im, rwi, iwi,
  rsum, isum, twopi;
  nless1 := nn - 1;
  twopi := 6·283185307180;
  comment this constant is the value of 2π correct to
  12 decimal places;
  z := twopi/n; rless1 := 0;
  if n < 0 then twopi := - twopi;
  for r := 1 step 1 until nb do
  begin kbase := b[r]; gpstep := wtk;
    wtk := wtk ÷ kbase;
    wrz := twopi/kbase;
    rwrz := cos (wrz); iwrz := sin (wrz);
    for g := 0 step gpstep until nless1 do
    begin if g = 0 then
      begin rwj := 1·0;
        iwj := 0·0
      end
      else
      begin wg := rev (g, b, nb, rless1) × wtk;
        wgz := wg × z;
        rwj := rw[1] := cos (wgz);
        iwj := iw[1] := sin (wgz)
      end;
      for j := 2 step 1 until kbase do
      begin re := rwrz × rwj - iwrz × iwj;
        im := rwrz × iwj + iwrz × rwj;
        rwj := rw[j] := re;
        iwj := iw[j] := im
      end;
      kend := g + wtk - 1;
      for k := g step 1 until kend do
      begin jj := k + origin;
        for j := 1 step 1 until kbase do
        begin if jj > nless1 then jj := jj - nn;
          rx[j] := rea[jj]; ix[j] := ima[jj];
          jj := jj + wtk
        end;
        jj := k + origin;
        for i := 1 step 1 until kbase do
        begin if jj > nless1 then jj := jj - nn;
          rwi := rw[i]; iwi := iw[i];
          rsum := rx[kbase]; isum := ix[kbase];
          for j := kbase - 1 step -1 until 1 do
          if i ≠ 1 ∨ g ≠ 0 then
          begin re := rsum × rwi - isum × iwi;
            im := rsum × iwi + isum × rwi;
            rsum := re + rx[j];
            isum := im + ix[j]
          end
          else
          begin rsum := rsum + rx[j];
            isum := isum + ix[j]
          end;
          rea[jj] := rsum; ima[jj] := isum;
          jj := jj + wtk
        end i
      end k
    end g; rless1 := r
  end r;
  if n < 0 then nless1 := - nless1;
  jkperm (rea, ima, nless1, origin, b, nb)
end
end complexfourier
```

Contributions for the Algorithms Supplement should be sent to

P. Hammersley
University Mathematical Laboratory
Corn Exchange Street
Cambridge

Discussion and Correspondence

Modification of the complex method of constrained optimization

By J. A. Guin*

On the basis of some recent computational experience using the complex method of Box (1965), it has been found that the following modifications in the method increase the chances of reaching the optimum.

(1) Box has suggested that a projected trial point be moved in halfway toward the centroid of the remaining points until a new point better than the rejected one is found. If by chance all points on the line from the centroid to the projected point are worse than the original point, application of this rule causes the projected point eventually to coincide with the centroid. When this happens no further progress is possible. Considering this situation, it is recommended that if the projection factor α is found to have been reduced below a certain quantity (we have found $\alpha = 10^{-5}$ to be a satisfactory criterion) without obtaining a better function value for the projected trial point, then this trial point should be replaced to its original unprojected position and the second worst point rejected instead. This procedure tends to keep the complex moving unless the centroid is indeed near the optimum.

(2) The rule of setting an independent variable to 0·000001 inside its limit sometimes causes the method to obtain a false optimum if all points of the complex fall into this hyperplane. This happens especially when the optimum is near, but not upon the constraint. To alleviate this situation, it is recommended that the above rule be abandoned and that only the

* *Department of Chemical Engineering, The University of Texas, Austin, Texas.*

rule for moving halfway toward the centroid be retained to deal with constraint violation.

(3) Obviously the method fails when the centroid falls into a non-feasible region as often occurs when searching non-convex spaces. We have found that it is desirable to continue moving toward at least a local optimum, and to effect this the method can be modified as follows. The centroid of the remaining points should be checked for feasibility before rejecting a point. If the centroid is found to be unfeasible, then the procedure is to discard all except the best point of the complex and to construct a new complex according to $x_i = x_0 + r_i(x_c - x_0)$. Here x_0 is the best point of the old complex, x_c is the unfeasible centroid, and r_i is a random number over the interval (0, 1). The construction of the new complex is restricted to a more favourable subspace of the original region and it will continue movement in the direction of an optimum. In constructing the new complex, rule (2) is invoked for constraint violation.

In actual practice the three modifications listed above were found to allow the complex method to reach the optimum in situations where it would have ordinarily terminated.

Correspondence

To the Editor,
The Computer Journal.

Gödel's theorem

Sir,

I would like to mention a correction to a statement of mine in a review of Arbib's book (this *Journal*, Vol. 8, p. 88).

It has been held by some people that Gödel's theorem shows that a man's reasoning transcends that of any Turing machine. I denied this in my review and suggested that the catch might be that both machine and man might not have enough internal states to complete Gödel's construction in all cases. This conjecture is false, as pointed out by Alan Tritter, but I still do not believe that a mathematical theorem can prove a metaphysical statement. I think I have given a proper discussion of the problem in my article "Human and Machine Logic", *British Journal of Philosophy and Science*, Vol. 18 (1967), pp. 144–147.

Yours sincerely,

I. J. Good,
Professor of Statistics

Virginia Polytechnic Institute,
College of Arts and Sciences,
Blacksburg, Virginia 24061.
24 October 1967.

To the Editor,
The Computer Journal.

Solution of linear differential equations

Sir,

A recent paper (Davison, 1967) proposes a step-by-step method for solving the set of simultaneous first-order linear time-invariant equations

$$\dot{x} = Ax + Bu(t). \qquad (1)$$

For an nth order formula, with step size h, the truncation error is $O(h^{n+1})$, n even, or $O(h^{n+2})$, n odd. An alternative approach gives equations of the same form, but with truncation error $O(h^{2n+1})$. These equations are (taking $t = 0$ to $t = h$ as a typical step)

$$x(h) = \left[\sum_{j=0}^{n} (-)^j K(n,j)(Ah)^j\right]^{-1} \times \left[\left(\sum_{j=0}^{n} K(n,j)(Ah)^j\right)x(0) + \sum_{k=0}^{n-1}\left(\sum_{j=0}^{n-1-k} K(n,j+k+1)(Ah)^j\right)Bu^{(k)}(0)h^{k+1} + \sum_{k=0}^{n-1}\left(\sum_{j=0}^{n-1-k} (-)^{j+k}K(n,j+k+1)(Ah)^j\right)Bu^{(k)}(h)h^{k+1} + C_n\right] \qquad (2)$$

where
$$K(n,j) = \frac{n!(2n-j)!}{(2n)!(n-j)!j!}$$

and
$$C_n = (-)^n \frac{(n!)^2}{(2n)!(2n+1)!} h^{2n+1}x^{(2n+1)}(\tfrac{1}{2}h) + \ldots$$
$$= (-)^n \frac{(n!)^2}{(2n)!(2n+1)!} \delta^{2n+1}x(\tfrac{1}{2}h) - \ldots$$

represents the truncation error.

The well-known trapezoidal formula, with a truncation error $O(h^3)$, is given by $n = 1$; $n = 2$ gives

$$x(h) = (I - \tfrac{1}{2}Ah + \frac{1}{12}A^2h^2)^{-1} \times \left[\left(I + \tfrac{1}{2}Ah + \frac{1}{12}A^2h^2\right)x(0) + \left(\tfrac{1}{2}I + \frac{1}{12}Ah\right)Bu(0)h + \left(\tfrac{1}{2}I - \frac{1}{12}Ah\right)Bu(h)h + \frac{1}{12}B\dot{u}(0)h^2 - \frac{1}{12}B\dot{u}(h)h^2 + \frac{1}{720}\delta^5x(\tfrac{1}{2}h) - \ldots\right]. \qquad (3)$$

Davison's proposed formula involves polynomials in A up to A^3 and derivatives of u up to the third order; the leading term of the expression corresponding to C_n is $\frac{1}{120}\delta^5x(\tfrac{1}{2}h) - \frac{1}{48}A^4h^4\delta x(\tfrac{1}{2}h)$. Thus equation (3) is to be preferred, since it is easier to calculate and has a comparable truncation error.

Equation (2) may be proved thus: consider

$$C_n = \sum_{j=0}^{n} (-)^j K(n,j)h^j[f^{(j)}(h) - (-)^j f^{(j)}(0)] \qquad (4)$$

put

$$f^{(j)}(h) - (-)^j f^{(j)}(0) = D^j[\exp(\tfrac{1}{2}hD) - (-)^j \exp(-\tfrac{1}{2}hD)]f \qquad (5)$$

where D represents the differential operator at time $t = \frac{1}{2}h$. The resultant expression can be put in the form

$$C_n = (-)^n\{n!/(2n)!\}\pi^{1/2}(hD)^{n+1/2}I_{n+1/2}(\tfrac{1}{2}hD)f \qquad (6)$$

where $I_{n+1/2}$ is a modified spherical Bessel function of the first kind; on expansion

$$C_n = (-)^n\frac{n!}{(2n)!}(hD)^{2n+1}\sum_{j=0}^{\infty}\frac{(n+j)!}{j!(2n+2j+1)!}(\tfrac{1}{2}hD)^{2j}f. \qquad (7)$$

Results equivalent to this have been derived previously, at least for $n = 1, 2, 3$; see Section 8.11 of Buckingham (1957).