1 PURPOSE

To find the minimum of a function F(x) of n variables x subject to both equality constraints $c_i(\underline{x}) = 0$ i = 1,2,...,k and inequality constraints $c_i(\underline{x}) \geq 0$ i = k + 1,...,m (k = 0 or k = m is allowed but k must be ≤ n). Derivatives of all functions with respect to x must be provided, both the vector $(\partial F/\partial x_1, \partial F/\partial x_2, ..., \partial F/\partial x_n)$ and the matrix whose ith |column is $(\partial c_i/\partial x_1, \partial c_i/\partial x_2, ..., \partial c_i/\partial x_n)$ for i = 1,2,...,m. These functions and derivatives must be specified in a user subroutine VF01B (see section 4). An initial estimate of the solution must be provided which need not be feasible. The subroutine allows advantage to be taken of the possibility that some of the constraints are linear, and also of certain other types of information about the problem, if available. If all the constraints are linear, the use of VF01A is not most efficient, and one of the LA or VE routines should be used. The method is a penalty function – Lagrangian method (see section 8) and VF01A calls VA09A to carry out the associated unconstrained minimizations.

2 ARGUMENT LIST

CALL VF01A(N,M,K,X,EPS,AKMIN,DFN,MAXFN,IPR1,IPR2,IW,MODE)

N An INTEGER set to the number of variables n (N ≥ 2).

M An INTEGER set to the total number of constraints m (M ≥ 1).

K An INTEGER set to the total number of equality constraints k.

X A REAL array of N elements in which the initial estimate of the solution must be set. VF01A returns the solution x in X.

EPS A REAL array of N elements, in which the tolerances for the unconstrained minimizations must be set. EPS(I) should be set so that EPS(I)/X(I) ≈ AKMIN, roughly speaking.

AKMIN A REAL number in which the relative error tolerance required in the constraint residuals must be set. VF01A will exit when $max\{|c_i(x)|/scaling\ factor\ for\ c_i\} \leq$ AKMIN for the active constraints {i} (see section 7).

DFN A REAL number in which the likely reduction in F(x) must be set. This is done in the same way as for VA09A – see the VA09A specification sheet.

MAXFN An INTEGER in which the maximum number of calls of VF01B on any one unconstrained minimization must be set. Roughly speaking 2 or 3 times MAXFN calls of VF01B are likely to be made altogether.

IPR1 An INTEGER controlling the frequency of printing from VF01A. Printing occurs every IPR1 iterations, except for details of increases to the σ_i which are always printed. No printing at all occurs (except for error diagnostics) if IPR1 = 0. All printing controlled by IPR1 is suitably annotated.

IPR2 An INTEGER controlling the frequency of printing from VA09A. IPR2 should be set as described in the VA09A specification sheet.

IW An INTEGER giving the amount of storage available in COMMON/VF01L/W(.). Set to 2500 unless wishing to change the restrictions (see section 5).

MODE An INTEGER controlling the mode of operation of VF01A. If any positive definite estimate is available of the hessian matrix of the penalty function, set |MODE| = 2 or 3, otherwise set |MODE| = 1 (see VA09A specification sheet). If estimates of the σ_i and θ_i parameters are available (see section 8) set MODE < 0, otherwise set MODE > 0. A normal setting for a one-off job with no information available is MODE = 1.

3 THE NAMED COMMON AREAS

Certain named COMMON areas must be declared and set on entry to VF01A.

COMMON/VF01E/C(150)	Set scale factors (0) for the constraints in C(1), C(2),...,C(M). Choose the magnitude of these scale factors to give an indication of the magnitude of the constraints evaluated about the initial approximation x. If any constraints are violated by an amount greater in modulus than that which is set, then the setting is increased accordingly. These scale factors are transferred to C(M+1),C(M+2),...,C(2M) by VF01A.
COMMON/VF01F/GC(25,50)	Set the derivatives of any linear constraints on entry rather than in VF01B. This is the most efficient and the numbers are not disturbed. The manner of setting is described in section 4.
COMMON/VF01G/T(150)	If MODE 0 is used, then set the parameters $\theta_1,\theta_2,...,\theta_m$ in T(1),T(2),...,T(M) and the parameters $\sigma_1,\sigma_2,...,\sigma_m$ in T(M+1),T(M+2),...,T(2M). The meaning of these parameters may be found in the report TP552 – see section 8.
COMMONVF01I/G2P(325)	If |MODE| = 2 or 3 set the estimated hessian matrix of the penalty function in G2P(1),..., G2P(N*(N+1)/2). The manner of setting is that described in the specification sheet of VA09A under the heading MODE.

4 THE USER SUBROUTINE VF01B

The user must declare a subroutine headed SUBROUTINE VF01B(N,M,X)

```
REAL X(1)
COMMON/VF01C/F
COMMON/VF01D/G(50)
COMMON/VF01E/C(150)
COMMON/VF01F/GC(25,50)
```

This subroutine must take x, the vector supplied in X(1),...,X(N) and set F(x) in F: $c_1(x)$,...,$c_m(x)$ in C(1),...,C(M); $(\partial F/\partial x_1,...,\partial F/\partial x_n)_x$ in G(1),...,G(N); and set $(\partial c_i/\partial x_1,...,\partial c_i/\partial x_n)_x$ in GC(1,I),...,GC(N,I) for all I = 1,2,...,M.

[Excepting the linear constraints which should be set on entry, as the numbers are constant]. Some time may also be saved if required by also including COMMON/VF01G/T(150) and by not evaluating GC(1,I),...,GC(N,I) when C(I) ≥ T(I) for any I K. Note that the optimum values F(x), $(\partial F/\partial x_1,...,\partial F/\partial x_n)$, etc. are left in these named COMMON areas on exit from VF01A. Note also that in the double precision version the user subroutine name is VF01BD and there is a D appended to the named COMMON areas.

5 REDEFINING NAMED COMMON AREAS

Local storage for VF01A is through named COMMON areas. These have been set on the assumption that N ≤ 25 and M ≤ 50. If it is desired to remove either or both of these restrictions, then it is necessary to increase the storage available in some or all of these areas. This can be done by defining the named COMMON areas in the users MAIN with the increased storage settings, in which case the extra storage will be effective throughout the whole program. The complete list of named COMMON used by VF01A and the corresponding values of N and M are as follows.

COMMON/VF01C/F,M,K,IS,MK,NU	independent of N and M

D/G(50)	2N
E/C(150)	3M
F/GC(25,50)*	N,M
G/T(150)	3M
H/GP(50)	μ (μ = max(M,N))
I(G2P(325)	N*(N+1)2
J/V(50)	μ
K/WW(150)	3μ
L/W(2500)**	μ^2
M/ZZ(100)	2μ
N/LT(100)	2M

NOTES

* On increasing M, when N < 25, redefine GC with bounds (N,M) not (25,M). VF01A has been coded under this assumption, as it requires less storage. (VF01A treats GC as a single suffix array).

** For M very large, μ^2 storage locations may be prohibitively large. However it is very unlikely that this amount of storage will actually be needed by VF01A (no problem has yet been encountered for which more than $(2N)^2$ locations have been required). Hence in these circumstances, either declare W with $(2N)^2$ locations (or whatever can be spared), and set the integer IW to this number in the calling sequence of VF01A. If more locations are required, then VF01A will stop and print out the storage required.

6 GENERAL

Use of COMMON:	named COMMON only – see section 5.
Workspace:	5K words unless N or M is redefined, when it is not usually more than $\sim 4\frac{1}{2}N^2$ + NM + 0(max (M,N)) words.
Other routines:	Calls VE04, VF01B (user subroutine), VF01Z (private), VA09A, MC11A, MC11E. VA09A calls MC11B in addition.
Input/Output:	No input, all output on stream 6 (line printer), controlled by user through IPR1 and IPR2.
Restrictions:	N ≤ 25 and M ≤ 50 but can be lifted – see section 5.
Date of routine:	August 1973.

7 ACCURACY

It may be that VF01A is unable to achieve the accuracy requested in the parameter AKMIN. In this case a diagnostic is printed. To find the cause of this, first examine the print out of the VA09A iteration. If this is anomalous ($\nabla \phi \to 0$ for instance) suspect a mistake in the programming of VF01B particularly in obtaining derivatives. If VA09A seems O.K., then other possible causes are (i) there is no feasible point (in which case $\sigma_i \to \infty$ and $c_i \to const \neq 0$), (ii) EPS has been set too large relative to AKMIN, (iii) AKMIN has been set too small relative to the machine precision, (iv) the problem is too ill-conditioned.

8 METHOD

That described in R. Fletcher, “An ideal penalty function for constrained optimization”, C.S.S.2, December 1973. The penalty function for inequalities is

$$\phi(\underline{x},\underline{\theta},\underline{\sigma}) = F(\underline{x}) + \tfrac{1}{2} \sum_i \sigma_i \, (c_i(\underline{x}) - \theta_i)_-^2$$

and each iteration involves minimizing $\phi(\underline{x})$ for fixed $\underline{\theta},\underline{\sigma}$. After each iteration the $\underline{\theta}$ and $\underline{\sigma}$ parameters are varied so that the sequence of minima $x_{\theta,\sigma}$ tends to the solution of constrained problem. The value of the product $\theta_i \sigma_i$ tends to the i^{th}|Lagrange multiplier of the problem. Any information about Lagrange multipliers, or about the hessian of ϕ can usefully be incorporated.

Convergence is guaranteed (in exact arithmetic) and this implementation of the method can be expected to converge at a second order rate.