TEST PROBLEMS FOR PARTIALLY SEPARABLE OPTIMIZATION AND RESULTS FOR THE ROUTINE PSPMIN

by Ph.L. Toint

[10 February 1983]

Report Nr 83/4

Abstract. In this paper, we give the complete Fortran listing for the problems used in [2] to test the routine PSPMIN for solving bounded partially separable optimization problems, together with the detailed results obtained on these problems by PSPMIN.

DEPARTMENT OF MATHEMATICS
FACULTES UNIVERSITAIRES DE NAMUR
NAMUR (BELGIUM)

Table of Contents

List of Tables

1 Introduction

In a recent paper [2], Griewank and Toint presented some numerical results obtained by experimenting with PSPMIN, a routine for solving partially separable minimization problems, with possible bounds on the variables. More precisely, they considered problems of the type

$$\min f(x) = \sum_{i=1}^{m} f_i(x), \qquad (1)$$

subject to the constraints

$$a_i \leq x_i \leq b_i \quad (i=1, \ldots, n), \qquad (2)$$

where each "element function" $f_i(\cdot)$ has a Hessian matrix of low rank compared to the total number of variables of the problem. They justified their interest in this particular form by pointing out many practical frameworks where a minimization problem would naturally have this structure, as finite elements, network problems, econometry and others.

After discussing some theoretical advantages of the problem ((1)-(2)), they described how a particular Fortran routine, called PSPMIN, was implemented, and they gave some numerical results obtained with this routine on a collection of 154 test problems. The purpose of this report is to provide a full description of these problems, as well as tables containing the complete results of the 457 experimental runs of PSPMIN on this collection.

Section 2 briefly surveys the collection of test problems, while Section 3 will presents the complete results obtained with PSPMIN. Section 4 contains the complete Fortran listing needed to experiment with the test examples, as well as a driver program for PSPMIN.

2 A survey of the test problem collection

The test problem collection was build on the basis of 25 different functions, by considering various dimensions and starting points, as well as the presence or absence of bounds on the variables. Most of these functions already appeared in the literature, and were only extended to higher dimension if needed. Our main source are Himmelblau [3], the Argonne test set by Moré, Garbow and Hillstrom [5], the TESTPAK program by Buckley [1] and the Hock-Schittkowski collection [4].

For all these problems, the analytical gradients are available, and most of them feature variable dimension and number of element functions. These dimensions range from 2 up to 1000. 52 of these problems involve bounds on some or all their variables.

In addition to this collection, 7 small problems are given, that were used to test the correctness of PSPMIN in the early stages.

These simple functions are numbered 1 up to 7, while the 43 tests are numbered from 8 up to 50.

We now describe them succesively.

2.1 Problem 1

Dimension : 3

Nbr of elements : 2

$f_1(x) = x_1,$

$f_2(x) = 0.5(x_1 - x_2)^2 + x_1^2 .$

Source : -

Derivatives : analytical

Convex : yes

Lower bound : 0

Nullspaces : correct

Lower bounds : $x_i \geqq 0$ (i=1,2)

Upper bounds : no

Starting point : (10. 4. 10.) (unfeasible)

2.2 Problem 2

Dimension : 3

Nbr of elements : 3

$f_1(x) = x_1^2$

$f_i(x) = 0.5(x_{i-1} - x_i)^2 \quad (i=2,3)$

Source : -

Derivatives : analytical

Convex : yes

Lower bound : 0

Nullspaces : too small

Lower bounds : 0

Upper bounds : 0

Starting point : (-1. 4. -1.)

2.3 Problem 3

Dimension : 3

Nbr of elements : 3

$f_1(x) = x_1^2$

$f_i(x) = 0.5(x_{i-1} - x_i)^2 \quad (i=2,3)$

Source : -

Derivatives : analytical

Convex : yes

Lower bound : 0

Nullspaces : correct

Lower bounds : 0

Upper bounds : 0

Starting point : (-1. 4. -1.)

2.4 Problem 4

Dimension : 4

Nbr of elements : 2

$f_1(x) = x_1^2 + x_2^2 + 1$

$f_2(x) = 0.5[(x_1-x_2)^2 + (x_3-x_4)^2] + 1$

Source : -

Derivatives : not available

Convex : yes

Lower bound : 2.

Nullspaces : correct

Lower bounds : 0

Upper bounds : 0

Starting point : (-1. 4. -1. 4.)

2.5 Problem 5

Dimension : 3

Nbr of elements : 2

$f_1(x) = x_1 + 2x_2$

$f_2(x) = 3x_2 + 10x_3$

Source : -

Derivatives : not given

Convex : yes

Lower bound : no

Nullspaces : correct

Lower bounds : $x_i \geq 0$ (i=1,2,3)

Upper bounds : no

Starting point : (-4. -5. 100.) (unfeasible)

2.6 Problem 6

Dimension : 4

Nbr of elements : 2

$f_1(x) = x_1 + 0.5x_2^2$

$f_2(x) = 0.5[(x_1-x_3)^2 + (x_2-x_4)^2]$

Source : -

Derivatives : not available

Convex : yes

Lower bound : 0

Nullspaces : correct

Lower bounds : $x_i \geq 0$ (i=1,2,3,4)

Upper bounds : no

Starting point : (7.8 3.4 -2. -9.) (feasible)

2.7 Problem 7

Dimension : 4

Nbr of elements : 2

$f_1(x) = x_1 + \mathrm{sqrt}\{1 + x_2^2\}$

$f_2(x) = \mathrm{sqrt}\{1 + 0.5[(x_1-x_3)^2 + (x_2-x_4)^2]\}$

Source : -

Derivatives : not available

Convex : yes

Lower bound : 0

Nullspaces : correct

Lower bounds : $x_i \leq 0$ (i=1,2,3,4)

Upper bounds : no

Starting point : (7.8 3.4 -2. -9.) (feasible)

2.8 Problem 8 : TRIDIA

Dimension : variable (n = 10, 25, 50, 500, 1000)

Nbr of elements : depends on the dimension (n)

$f_1(x) = (2x_1 - 1)^2$,

$f_i(x) = i(2x_{i-1} - x_i)^2$

for i=2, ..., n.

Source : D. Shanno

Derivatives : available

Convex : yes

Lower bound : 0

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -1$ (i=1, ..., n)

2.9 Problem 9 : Shifted TRIDIA

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n)

$$f_1(x) = (2x_1 - 1)^2 + x_2^2,$$

$$f_i(x) = i(2x_{i-1} - x_i)^2 - (i-1)x_{i-1}^2 + ix_i^2$$

for i=2, ..., n-1, and

$$f_n(x) = n(2x_{n-1} - x_n)^2 - (n-1)x_{n-1}^2$$

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -1$ (i=1, ..., n)

2.10 Problem 10 : The ever famous Rosenbrock function

Dimension : variable (n = 2, 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$f_i(x) = 100(x_{i+1} - x_i^2)^2 + (x_{i+1} - 1)^2 \quad (i=1, \ldots, n-1)$

Source : Gill and Murray

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : none

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -1$ (i=1, ..., n)

2.11 Problem 11 : Linear Minimum Surface

Dimension : variable ($n = p^2$ = 25, 64, 121, 484)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$f_i(x) = \text{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$

for i = 1, ..., $(p-1)^2$, and where

s = int[i/(p-1)]p + mod(i,p-1).

Source : Griewank and Toint

Derivatives : available

Convex : yes

Lower bound : 1.

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t,$

$x_{(i-1)p+1} = 1 + 8t,$

$x_{i+p(p-1)} = 9 + 4t,$

$x_{ip} = 5 + 8t,$

for i=1, ..., p, and where

$t = (i - 1)/(p - 1).$

The other variables are the free variables of the problem, and are initialized to 0.

2.12 Problem 12 : Nonlinear Minimum Surface

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$f_i(x) = \text{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$

for $i = 1, \ldots, (p-1)^2$, and where

$s = \text{int}[i/(p-1)]p + \text{mod}(i,p-1)$.

Source : Griewank and Toint

Derivatives : available

Convex : yes

Lower bound : 1.

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t + 10(1+t)^2$,

$x_{(i-1)p+1} = 1 + 8t + 10(1-t)^2$,

$x_{i+p(p-1)} = 9 + 4t + 10t^2$,

$x_{ip} = 5 + 8t + 10(2-t)^2$,

for i=1, ..., p, and where

$t = (i - 1)/(p - 1)$.

The other variables are the free variables of the problem, and are initialized to 0.

2.13 Problem 13 : Shifted Linear Minimum Surface

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$$f_i(x) = q_i(x) - x_s^2/m$$

for i=1, ..., p-1 or if mod(i,p-1)=1,

$$f_i(x) = q_i(x) + x_{s+p+1}^2/m$$

for i=m-p+1, ..., m or if mod(i,p-1)=0, and

$$f_i(x) = q_i(x) - x_s^2/m + x_{s+p+1}^2/m$$

otherwise, where

$$q_i(x) = \mathrm{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$$

and

$$s = \mathrm{int}[i/(p-1)]p + \mathrm{mod}(i,p-1).$$

Source : Griewank and Toint

Derivatives : available

Convex : no

Lower bound : 1.

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$$x_i = 1 + 4t,$$

$$x_{(i-1)p+1} = 1 + 8t,$$

$$x_{i+p(p-1)} = 9 + 4t,$$

$$x_{ip} = 5 + 8t,$$

for i=1, ..., p, and where

$$t = (i - 1)/(p - 1).$$

The other variables are the free variables of the problem, and are initialized to 0.

2.14 Problem 14 : Shifted Nonlinear Minimum Surface

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$$f_i(x) = q_i(x) - x_s^2/m$$

for i=1, ..., p-1 or if mod(i,p-1)=1,

$$f_i(x) = q_i(x) + x_{s+p+1}^2/m$$

for i=m-p+1, ..., m or if mod(i,p-1)=0, and

$$f_i(x) = q_i(x) - x_s^2/m + x_{s+p+1}^2/m$$

otherwise, where

$$q_i(x) = \mathrm{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$$

and

$$s = \mathrm{int}[i/(p-1)]p + \mathrm{mod}(i,p-1).$$

Source : Griewank and Toint

Derivatives : available

Convex : no

Lower bound : 1.

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t + 10(1+t)^2,$

$x_{(i-1)p+1} = 1 + 8t + 10(1-t)^2,$

$x_{i+p(p-1)} = 9 + 4t + 10t^2,$

$x_{ip} = 5 + 8t + 10(2-t)^2,$

for i=1, ..., p, and where

$t = (i - 1)/(p - 1).$

The other variables are the free variables of the problem, and are initialized to 0.

2.15 Problem 15 : Nondiagonal extension of Rosenbrock function

Dimension : variable (n = 10, 20, 30, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$f_i(x) = 100(x_{i+1} - x_1^2)^2 + (x_{i+1} - 1)^2$

for i=1, ..., n-1.

Source : Shanno

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -1$ (i=1, ..., n)

2.16 Problem 16 : Boundary value problem

Dimension : variable (n = 10, 20, 30, 50, 100)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = [2x_i - x_{i-1} - x_{i+1} + h^2(x_i + ih + 1)^3/2]^2$$

for i = 2, ..., n-1, and where

$h = 1/(n-1)$.

Source : More, Garbow and Hillstrom

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : The variables x_1 and x_n are fixed to 0. for the complete calculation. The others are the free variables, and are initialized by

$x_i = ih(ih - 1)$ (i=2, ..., n-1).

2.17 Problem 17 : Broyden tridiagonal nonlinear system

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = [(3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1]^2$$

for i=2, ..., n-2.

Source : Broyden

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : The variables x_1 and x_n are fixed to 0. for the complete calculation. The others are the free variables, and are initialized to -1.

2.18 Problem 18 : Broyden Banded Function

Dimension : variable (n = 10, 50, 100, 250)

Nbr of elements : depends on the dimension (n)

$$f_i(x) = [x_i(2 + 5x_i^2) - q_i(x)]^2$$

for i=1, ..., n, and where

$s = \max\{1, i-5\}$,

$t = \min\{n, i+1\}$

and

$$q_i(x) = \sum_{j=s}^{i-1} x_j(1 + x_j) + \sum_{j=i+1}^{t} x_j(1 + x_j) .$$

Source : Moré, Garbow and Hillstrom

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -1$ (i=1, .., n)

2.19 Problem 19 : Extended Powell Singular Function

Dimension : variable (n = 3p+1 = 4, 10, 49, 100, 502)

Nbr of elements : depends on the dimension (m=p)

$$f_i(x) = (x_s + 10x_{s+1})^2 + 5(x_{s+2} - x_{s+3})^2 + (x_{s+1} - 2x_{s+2})^4 + 10(x_s - x_{s+3})^4,$$

for i=1, ..., m, and where

$s = 1 + 3(i-1)$.

Source : More, Garbow and Hillstrom (extended)

Derivatives : available

Convex : yes

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : The starting values are

$x_1 = 3,$

$x_{3i-1} = -1,$

$x_{3i} = 0,$

$x_{3i+1} = 1,$

for i = 1, ..., p.

2.20 Problem 20 : Wrong Extended Wood's Function

Dimension : variable (n = 2p = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (m=(n-2)/2)

$$f_i(x) = 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 + 90(x_{2i+2} - x_{2i+1}^2)^2 + (1 - x_{2i+1})^2 + 10(x_{2i} + x_{2i+2} -2)^2 + 10(x_{2i} - x_{2i+2})^2$$

Source : a mistake in writing the famous Wood's function

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -3$ if $\mathrm{mod}(i,2) = 1$, $x_i = -1$ if $\mathrm{mod}(i,2) = 0$.

2.21 Problem 21 : Gaussian-like Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = (a + x_{i+2}^2)(2 - \exp[-s/t])$$

for i=1, ..., n-2, and where

$$s = (x_i - x_{i+1})^2$$

and

$$t = 0.1 + x_{i+2}^2.$$

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : correct

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 3$ (i=1, ..., n).

2.22 Problem 22 : Diagonal quadratic

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = 100(x_{i+1}^2 + x_{i+2}^2) + x_i^2$$

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 3$ (i=1, ..., n).

2.23 Problem 23 : Spiked Linear Minimum Surface

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$$f_i(x) = \text{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$$

for $i = 1, \ldots, (p-1)^2$, and where

$$s = \text{int}[i/(p-1)]p + \text{mod}(i,p-1).$$

Source : -

Derivatives : available

Convex : yes

Lower bound : 1

Nullspaces : correct

Lower bounds : $x_r \geq 2.5$ where

$r = 0.5p^2 + p + 1.$

Upper bounds : $x_r \leq 4.$

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t,$

$x_{(i-1)p+1} = 1 + 8t,$

$x_{i+p(p-1)} = 9 + 4t,$

$x_{ip} = 5 + 8t,$

for i=1, ..., p, and where

$t = (i - 1)/(p - 1).$

The other variables are the free variables of the problem, and are initialized to 0.

2.24 Problem 24 : Plane Constrained Linear Minimum Surface

Dimension : variable ($n=p^2$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$f_i(x) = \mathrm{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$

for $i = 1, \ldots, (p-1)^2$, and where

$s = \mathrm{int}[i/(p-1)]p + \mathrm{mod}(i,p-1)$.

Source : -

Derivatives : available

Convex : yes

Lower bound : 1

Nullspaces : correct

Lower bounds : $x_i \geq 2.5$ where $i=r$, $r-p-1$, $r-p$, $r-p+1$, $r-1$, r, $r+1$, $r+p-1$, $r+p$, $r+p+1$, with

$r = 0.5p^2 + p + 1$.

Upper bounds : $x_i \leq 4$ for the same i as in the lower bounds

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t$,

$x_{(i-1)p+1} = 1 + 8t$,

$x_{i+p(p-1)} = 9 + 4t$,

$x_{ip} = 5 + 8t$,

for $i=1, \ldots, p$, and where

$t = (i - 1)/(p - 1)$.

The other variables are the free variables of the problem, and are initialized to 0.

2.25 Problem 25 : Spiked Nonlinear Minimum Surface

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$f_i(x) = \mathrm{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$

for $i = 1, \ldots, (p-1)^2$, and where

$s = \mathrm{int}[i/(p-1)]p + \mathrm{mod}(i,p-1)$.

Source : -

Derivatives : available

Convex : yes

Lower bound : 1

Nullspaces : correct

Lower bounds : $x_r \geqq 2.5$ where

$r = 0.5p^2 + p + 1$.

Upper bounds : $x_r \leqq 4$.

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t + 10(1+t)^2$,

$x_{(i-1)p+1} = 1 + 8t + 10(1-t)^2$,

$x_{i+p(p-1)} = 9 + 4t + 10t^2$,

$x_{ip} = 5 + 8t + 10(2-t)^2$,

for $i=1, \ldots, p$, and where

$t = (i - 1)/(p - 1)$.

The other variables are the free variables of the problem, and are initialized to 0.

2.26 Problem 26 : Plane Constrained Nonlinear Minimum Surface

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$f_i(x) = \text{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$

for $i = 1, \ldots, (p-1)^2$, and where

$s = \text{int}[i/(p-1)]p + \text{mod}(i,p-1)$.

Source : -

Derivatives : available

Convex : yes

Lower bound : 1

Nullspaces : correct

Lower bounds : $x_i \geq 2.5$ where i=r, r-p-1, r-p, r-p+1, r-1, r, r+1, r+p-1, r+p, r+p+1, with

$r = 0.5p^2 + p + 1$.

Upper bounds : $x_i \leq 4$ for the same i as in the lower bounds

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t + 10(1+t)^2$,

$$x_{(i-1)p+1} = 1 + 8t + 10(1-t)^2,$$

$$x_{i+p(p-1)} = 9 + 4t + 10t^2,$$

$$x_{ip} = 5 + 8t + 10(2-t)^2,$$

for i=1, ..., p, and where

$$t = (i - 1)/(p - 1).$$

The other variables are the free variables of the problem, and are initialized to 0.

2.27 Problem 27 : Extended Wood's Bounded Problem

Dimension : variable (n = 2p = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (m=(n-2)/2)

$$\begin{aligned} f_i(x) = 100(x_{2i} - x_{2i-1}^2)^2 &+ (1 - x_{2i-1})^2 \\ &+ 90(x_{2i+2} - x_{2i+1}^2)^2 + (1 - x_{2i+1})^2 \\ &+ 10.1(x_{2i} - 1)^2 + 10.1(x_{2i+2} - 1)^2 \\ &+ 19.8(x_{2i} - 1)(x_{2i+2} - 1) \end{aligned}$$

for i=1, ..., m.

Source : Himmelblau (8)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : $x_i \geq -10$ (i=1, ..., n)

Upper bounds : $x_i \leq 0.5$ (i=1, ..., n)

Starting point : x_i = -3 if mod(i,2) = 1, x_i = -1 if mod(i,2) = 0.

2.28 Problem 28 : Paviani's Bounded Problem

Dimension : variable (n = 7p+3 = 10, 52, 500, 997)

Nbr of elements : depends on the dimension (p)

$$f_i(x) = \sum_{j=s}^{t} [\log^2(x_j - 2) + \log^2(10 - x_j)] - [\prod_{i=s}^{t} x_i]^{0.2}$$

for i=1, ..., p, and where

$s = 7(i-1)+1$

and

$t = 7(i-1)+10.$

Source : Himmelblau (17)

Derivatives : available

Convex : yes

Lower bound : no

Nullspaces : no

Lower bounds : $x_i \geq 2.001$ (i=1, ..., n)

Upper bounds : $x_i \leq 9.999$ (i=1, ..., n)

Starting point : $x_i = 9$ (i=1, ..., n) (feasible)

2.29 Problem 29 : Extended McCormick's Bounded Problem

Dimension : variable (n = 2, 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = \sin(x_i + x_{i+1}) + (x_{i+1} - x_i)^2 - 1.5x_i + 2.5x_{i+1} + 1$$

for i=1, ..., n-1.

Source : Schittkowski (5) (extended)

Derivatives : available

Convex : no

Lower bound : -5.

Nullspaces : no

Lower bounds : $x_i \geq -1.5$ (i=1, ..., n)

Upper bounds : $x_i \leq 3$ (i=1, ..., n)

Starting point : $x_i = 0$ (i=1, ..., n)

2.30 Problem 30 : Extended Wood Function

Dimension : variable (n = 2p = 4, 10, 50, 500, 1000)

Nbr of elements : depends on the dimension ((n-2)/2)

$$\begin{aligned} f_i(x) = {} & 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \\ & + 90(x_{2i+2} - x_{2i+1}^2)^2 + (1 - x_{2i+1})^2 \\ & + 10.1(x_{2i} - 1)^2 + 10.1(x_{2i+2} - 1)^2 \\ & + 19.8(x_{2i} - 1)(x_{2i+2} - 1) \end{aligned}$$

for i=1, ..., m.

Source : Testpack (Woods)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -3$ if mod(i,2) = 1, $x_i = -1$ if mod(i,2) = 0.

2.31 Problem 31 : Extended ENGVL1 Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = (x_i^2 + x_{i+1}^2)^2 - 4x_i + 3$$

for i=1, ..., n-1.

Source : Testpack (10) (modified)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 2$ (i=1, ..., n)

2.32 Problem 32 : Extended CRGLVY Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (m=(n-2)/2)

$$f_i(x) = (x_{2i} - \exp[x_{2i-1}])^4 + 100(x_{2i} - x_{2i+1})^6 + (\sin[x_{2i+1} - x_{2i+2}]/\cos[x_{2i+1} - x_{2i+2}])^4 + x_{2i-1}^8 + (x_{2i+2} - 1)^2$$

for i=1, ..., m.

Source : Testpack (18) (extended)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 2$ (i=1, ..., n)

2.33 Problem 33 : Extended Freudenstein and Roth Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = [x_i + x_{i+1}((5 - x_{i+1})x_{i+1} - 2) - 13]^2$$
$$+ [x_i + x_{i+1}((x_{i+1} + 1)x_{i+1} - 14) - 29]^2$$

for i=1, ..., n-1.

Source : Testpack (24) (extended)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : x_i = -2 (i=1, ..., n)

2.34 Problem 34 : Extended Powell Badly Scaled Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = (10000x_i x_{i+1} - 1)^2$$
$$+ (\exp[-x_i] + \exp[-x_{i+1}] - 1.0001)^2$$

for i=1, ..., n-1.

Source : Testpack (22) (extended)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 1$ (i=1, ..., n)

2.35 Problem 35 : Extended SCHMVT problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$f_i(x) = 3 - 1/s - \sin(t) - \exp[-r^2]$

for i=1, ..., n-2, and where

$s = 1 + (x_i - x_{i+1})^2,$

$t = (x_{i+1} pi + x_{i+2})/2$

and

$r = (x_i + x_{i+2})/x_{i+1} - 2.$

Source : Testpack (14) (extended)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 0.5$ (i=1, ..., n)

2.36 Problem 36 : Cube Problem

Dimension : variable (n = 2, 10, 50)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = 100(x_{i+1} - x_i^3)^2 + (1 - x_i)^2$$

for i=1, ..., n-1.

Source : Testpack (5) (extended)

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = -2$ (i=1, ..., n)

2.37 Problem 37 : Bounded Broyden Tridiagonal Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = [(3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1]^2$$

for i=2, ..., n-2.

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : yes

Lower bounds : $x_i \geq 0.65$ (i=1, ...,n)

Upper bounds : $x_i \leq 0.71$ (i=1, ...,n)

Starting point : The variables x_1 and x_n are fixed to 0. for the complete calculation. The others are the free variables, and are initialized to -1.

2.38 Problem 38 : Bounded CRGLVY Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (m=(n-2)/2)

$$f_i(x) = (x_{2i} - \exp[x_{2i-1}])^4 + 100(x_{2i} - x_{2i+1})^6 + (\sin[x_{2i+1} - x_{2i+2}]/\cos[x_{2i+1} - x_{2i+2}])^4 + x_{2i-1}^8 + (x_{2i+2} - 1)^2$$

for i=1, ..., m.

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : $x_i \leq 0$ (i=1, ..., n)

Starting point : $x_i = 2$ (i=1, ..., n)

2.39 Problem 39 : Bounded ENGLV1 Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = (x_i^2 + x_{i+1}^2)^2 - 4x_i + 3$$

for i=1, ..., n-1.

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : $x_i \geq 0.5$ (i=1, ..., n)

Upper bounds : $x_i \geqq 0.63$ (i=1, ..., n)

Starting point : $x_i = 2$ (i=1, ..., n)

2.40 Problem 40 : Bounded Freudenstein and Roth Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = [x_i + x_{i+1}((5 - x_{i+1})x_{i+1} - 2) - 13]^2 + [x_i + x_{i+1}((x_{i+1} + 1)x_{i+1} - 14) - 29]^2$$

for i=1, ..., n-1.

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : $x_i \leqq 3$ (i=1, ..., n)

Starting point : $x_i = -2$ (i=1, ..., n)

2.41 Problem 41 : Bounded SCHMVT Problem

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = 3 - 1/s - \sin(t) - \exp[-r^2]$$

for i=1, ..., n-2, and where

$$s = 1 + (x_i - x_{i+1})^2,$$

$$t = (x_{i+1}pi + x_{i+2})/2$$

and

$$r = (x_i + x_{i+2})/x_{i+1} - 2.$$

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : no

Lower bounds : no

Upper bounds : x_i =< 0.75 if mod(i,2) = 1, x_i =< 10.0 if mod(i,2) = 0.

Starting point : x_i = 0.5 (i=1, ..., n)

2.42 Problem 42 : Bounded Rosenbrock Problem

Dimension : variable (n = 10, 50, 100)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = 100(x_{i+1} - x_i^2)^2 + (x_{i+1} - 1)^2$$

for i=1, ..., n-1.

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : none

Lower bounds : no

Upper bounds : $x_i \leq 0.5$ (i=1, ...,n)

Starting point : $x_i = -1$ (i=1, ..., n)

2.43 Problem 43 : Bounded TRIDIA

Dimension : variable (n = 10, 25, 50, 500, 1000)

Nbr of elements : depends on the dimension (n)

$f_1(x) = (2x_1 - 1)^2$,

$f_i(x) = i(2x_{i-1} - x_i)^2$

for i=2, ..., n.

Source : -

Derivatives : available

Convex : yes

Lower bound : 0

Nullspaces : correct

Lower bounds : no

Upper bounds : $x_1 \leq 0.1$, $x_i \leq 0.05$ (i=2, ..., n)

Starting point : $x_i = -1$ (i=1, ..., n)

2.44 Problem 44 : Plane Constrained Shifted Linear Minimum Surface Problem

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$$f_i(x) = q_i(x) - x_s^2/m$$

for i=1, ..., p-1 or if mod(i,p-1)=1,

$$f_i(x) = q_i(x) + x_{s+p+1}^2/m$$

for i=m-p+1, ..., m or if mod(i,p-1)=0, and

$$f_i(x) = q_i(x) - x_s^2/m + x_{s+p+1}^2/m$$

otherwise, where

$$q_i(x) = \mathrm{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$$

and

$$s = \mathrm{int}[i/(p-1)]p + \mathrm{mod}(i,p-1).$$

Source : -

Derivatives : available

Convex : yes

Lower bound : 1

Nullspaces : correct

Lower bounds : $x_i \geq 2.5$ where i=r, r-p-1, r-p, r-p+1, r-1, r, r+1, r+p-1, r+p, r+p+1, with

$$r = 0.5p^2 + p + 1.$$

Upper bounds : $x_i \leq 4$ for the same i as in the lower bounds

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$x_i = 1 + 4t,$

$x_{(i-1)p+1} = 1 + 8t,$

$x_{i+p(p-1)} = 9 + 4t,$

$x_{ip} = 5 + 8t,$

for i=1, ..., p, and where

$t = (i - 1)/(p - 1).$

The other variables are the free variables of the problem, and are initialized to 0.

2.45 Problem 45 : Plane Constrained Shifted Nonlinear Minimum Surface Problem

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$f_i(x) = q_i(x) - x_s^2/m$

for i=1, ..., p-1 or if mod(i,p-1)=1,

$f_i(x) = q_i(x) + x_{s+p+1}^2/m$

for i=m-p+1, ..., m or if mod(i,p-1)=0, and

$f_i(x) = q_i(x) - x_s^2/m + x_{s+p+1}^2/m$

otherwise, where

$q_i(x) = \text{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$

and

$s = \text{int}[i/(p-1)]p + \text{mod}(i,p-1).$

Source : -

Derivatives : available

Convex : yes

Lower bound : 1

Nullspaces : correct

Lower bounds : $x_i \geq 2.5$ where i=r, r-p-1, r-p, r-p+1, r-1, r, r+1, r+p-1, r+p, r+p+1, with

$$r = 0.5p^2 + p + 1.$$

Upper bounds : $x_i \leq 4$ for the same i as in the lower bounds

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$$x_i = 1 + 4t + 10(1+t)^2,$$

$$x_{(i-1)p+1} = 1 + 8t + 10(1-t)^2,$$

$$x_{i+p(p-1)} = 9 + 4t + 10t^2,$$

$$x_{ip} = 5 + 8t + 10(2-t)^2,$$

for i=1, ..., p, and where

$$t = (i - 1)/(p - 1).$$

The other variables are the free variables of the problem, and are initialized to 0.

2.46 Problem 46 : Probabilistic Penalty Function Problem

Dimension : variable (n = 10, 50, 100)

Nbr of elements : depends on the dimension (n)

$$f_i(x) = (x_i + x_{i+1})\exp\{-x_i x_{i+1}\}$$

for i=1, ..., n-1, and

$$f_n(x) = 100(\mathrm{SUM}_{i=1}^{n} x_i - 1)^2$$

Source : -

Derivatives : available

Convex : no

Lower bound : yes

Nullspaces : no

Lower bounds : no

Upper bounds : no

Starting point : $x_i = 0.5$ (i=1, ..., n)

2.47 Problem 47 : PSPDOC Example

Dimension : 4

Nbr of elements : 2

$$f_i(x) = \mathrm{sqrt}\{1 + x_i^2 + (x_{i+1} - x_{i+2})^2\}$$

for i=1 ,2.

Source : -

Derivatives : available

Convex : yes

Lower bound : 0.0

Nullspaces : correct

Lower bounds : no

Upper bounds : yes

Starting point : $x_i = 3$ (i=1, 2, 3, 4)

2.48 Problem 48 : Extended Rosenbrock with close starting point

Dimension : variable (n = 2, 10, 50, 100)

Nbr of elements : depends on the dimension (n-1)

$$f_i(x) = 100(x_{i+1} - x_i^2)^2 + (x_{i+1} - 1)^2$$

for i=1, ..., n-1.

Source : Gill and Murray

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : none

Lower bounds : no

Upper bounds : no

Starting point : $x_i = i/(n+1)$ (i=1, ..., n)

2.49 Problem 49 : Broyden Tridiagonal with wrong nullspace

Dimension : variable (n = 10, 50, 500, 1000)

Nbr of elements : depends on the dimension (n-2)

$$f_i(x) = [(3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1]^2$$

for i=2, ..., n-2.

Source : -

Derivatives : available

Convex : no

Lower bound : 0

Nullspaces : too small

Lower bounds : no

Upper bounds : no

Starting point : The variables x_1 and x_n are fixed to 0. for the complete calculation. The others are the free variables, and are initialized to -1.

2.50 Problem 50 : Nonlinear Minimum Surface Problem with Wrong Nullspace

Dimension : variable ($n = p^2 = 64, 484$)

Nbr of elements : depends on the dimension ($m=(p-1)^2$)

$$f_i(x) = \text{sqrt}\{1+m[(x_s-x_{s+p+1})^2+(x_{s+1}-x_{s+p})^2]/2\}$$

for $i = 1, \ldots, (p-1)^2$, and where

$s = \text{int}[i/(p-1)]p + \text{mod}(i,p-1).$

Source : Griewank and Toint

Derivatives : available

Convex : yes

Lower bound : 1.

Nullspaces : too small

Lower bounds : no

Upper bounds : no

Starting point : the variables on the boundaries of the unit square are kept fixed during the calculation to the values :

$$x_i = 1 + 4t + 10(1+t)^2,$$

$$x_{(i-1)p+1} = 1 + 8t + 10(1-t)^2,$$

$$x_{i+p(p-1)} = 9 + 4t + 10t^2,$$

$$x_{ip} = 5 + 8t + 10(2-t)^2,$$

for i=1, ..., p, and where

$$t = (i - 1)/(p - 1).$$

The other variables are the free variables of the problem, and are initialized to 0.

3 Complete results for the routine PSPMIN

In this section, we present the complete results obtained with the routine PSPMIN, whose summary has been discussed in [2].

The symbols "*", placed after the iteration count of a particular problem, mean that the noise on the function value was juged too high by PSPMIN, who stopped nevertheless very close to the minimum. This is caused by high optimal function values. In these cases, minimization was therefore considered as successful.

The abbreviations "no conv.", "nr" and ls.fail." mean respectively no convergence in 500 iterations, problem not run and linesearch failure, due to inaccuracy in the estimated gradients.

test problem	n	n'	ns	alg1	alg2	alg3
Tridia [8]	10	10	10	3	6	2
	25	25	25	3	4	3
	50	50	50	3	6	3
	500	500	500	4	8	3
	1000	1000	1000	4.	8	3
Shifted Tridia [9]	10	10	10	10	11	2
	50	50	50	17	13	3
	500	500	500	22	16	4
	1000	1000	1000	22	15	3
Extended Rosenbrock [10]	2	2	1	18	22	1
	10	10	9	20	20	64
	50	50	49	20	15	255
	500	500	499	20	18	nr
	1000	1000	999	21	16	nr
LMS [11]	25	9	16	12	10	12
	64	36	49	13	11	13
	121	81	100	15	11	14
	484	400	441	17	18	18
NLMS [12]	64	36	49	14	12	14
	484	400	441	19	15	22
Shifted LMS [13]	64	36	49	27	28	15
	484	400	441	48	69	35
Shifted NLMS [14]	64	36	49	34	26	26
	484	400	441	98	74	56
Nondia [15]	10	10	9	20	23	1
	20	20	19	20	28	1
	30	30	29	20	27	1
	50	50	49	20	28	1
	500	500	499	21	25	2
	1000	1000	999	21	25	2
Boundary Value [16]	10	8	8	16	15	11
	20	18	18	19	16	15
	30	28	28	19	16	18
	50	48	48	24	18	22
	100	98	98	54	21	55

Table 1: Main iterations 1

test problem	n	n′	ns	alg1	alg2	alg3
Broyden Tridiagonal [17]	10	8	8	15	13	10
	50	48	48	14	12	11
	500	448	448	18	14	10
	1000	998	998	17	15	10
Broyden Banded [18]	10	10	10	29	23	24
	50	50	50	26	24	26
	100	100	100	25	25	25
	250	250	250	26	24	27
Powell Singular [19]	4	4	1	54	39	37
	10	10	3	56	40	75
	49	49	16	65	42	95
	100	100	33	79	46	70
	502	502	167	88	49	93
Wrong Wood [20]	10	10	4	30	31	23
	50	50	24	32	31	29
	500	500	249	43	34	36
	1000	1000	499	43	33	35
Gaussian [21]	10	10	8	15	9	1
	50	50	48	22	8	14
	500	500	498	19	12	19
	1000	1000	998	36	19	39
Diagonal Quadratic [22]	10	10	8	16	14	1
	50	50	48	17	11	1
	500	500	498	16	12	1
	1000	1000	998	16	12	1
Spiked LMS [23, bounds]	64	36	49	14	13	16
	484	400	441	17	18	19
Plane LMS [24, bounds]	64	36	49	15	13	16
	484	400	441	19	20	20
Spiked NLMS [25, bounds]	64	36	49	17	17	20
	484	400	441	26	24	27
Plane NLMS [26, bounds]	64	36	49	16	16	16
	484	400	441	21	21	23

Table 2: Main iterations 2

test problem	n	n'	ns	alg1	alg2	alg3
Bounded Wood [27, bounds]	10	10	4	10	8	2
	50	50	24	12	10	3
	500	500	249	11	9	3
	1000	1000	499	11	9	3
Paviani [28, bounds]	10	10	1	6	5	6
	52	52	7	6	5	6
	500	500	70	7	6	6
	997	997	140	7	6	6
McCormick [29, bounds]	2	2	1	6	5	4
	10	10	9	11	8	7
	50	50	49	12	9	7
	500	500	499	12	9	8
	1000	1000	999	12	9	8
Extended Wood [30]	4	4	1	39	36	92
	10	10	4	39	35	145
	50	50	24	42	39	184
	500	500	249	53	45	nr
	1000	1000	499	57	46	nr
Extended ENGVL1 [31]	10	10	9	11	9	11
	50	50	49	15	11	12
	500	500	499	14	11	12
	1000	1000	999	14	11	12
Extended CRGLVY [32]	10	10	4	71	62	54
	50	50	24	57	45	55
	500	500	249	62	89*	54
	1000	1000	499	76	no conv	65
Extended Freudenstein and Roth [33]	10	10	9	24	20	12
	50	50	49	27	21	11
	500	500	499	51	20	12
	1000	1000	999	22	20	12
Extended Powell Badly Scaled [34]	10	10	9	21	21	23
	50	50	49	21	21	27
	500	500	499	21	21	29
	1000	1000	999	21	21	31
Extended SCHMVT [35]	10	10	8	12	10	9
	50	50	48	12	9	9
	500	500	498	12	9	10
	1000	1000	998	13	9	10

Table 3: Main iterations 3

test problem	n	n′	ns	alg1	alg2	alg3
Extended	2	2	1	48	50	69
Cube	10	10	9	117	118	195
[36]	50	50	49	432	nr	nr
Bounded	10	8	8	8	5	7
3D Broyden	50	48	48	11	7	12
[37, bounds]	500	498	498	11	7	45
	1000	998	998	11	7	44*
Bounded	10	10	4	10	8	13
CRGLVY	50	50	24	11	10	20
[38, bounds]	500	500	249	11	9	81
	1000	1000	499	11	9	116
Bounded	10	10	9	8	3	4
ENGLV1	50	50	49	8	3	4
[39, bounds]	500	500	499	8	3	4
	1000	1000	999	8	3	4
Bounded	10	10	9	20	17	9
Freud. Roth	50	50	49	23	25*	9
[40, bounds]	500	500	499	24	25*	9
	1000	1000	999	20	40*	10
Bounded	10	10	8	9	8	7
SCHMVT	50	50	48	11	10	8
[41, bounds]	500	500	498	11	9	8
	1000	1000	998	11	9	8
Bounded	10	10	9	19	17	21
Rosenbrock	50	50	49	60	58	21
[42, bounds]	100	100	99	112	109	27

Table 4: Main iterations 4

test problem	n	n′	ns	alg1	alg2	alg3
Bounded TRIDIA [43, bounds]	10 50 500 1000	10 50 500 1000	9 49 499 999	3 3 3 3	5 6 6 7	2 3 3 3
Plane B SLMS [44, bounds]	64 484	36 400	49 441	16 75	18 58	17 53
Plane B SNLMS [45, bounds]	64 484	36 400	49 441	25 90	20 95	23 59
Prob. penalty [46]	10 50 100	10 50 100	10 50 100	6 11 11	ls.fail. ls.fail. ls.fail.	6 no conv. no conv.
PSPDOC [47]	4	4	2	5	7	6
Extended Rosenbrock # [48]	2 10 50 100	2 10 50 100	1 9 49 99	21 64 82 154	17 65 76 144	33 74 79 154
Extended 3D-Broyden [49] (wrong N)	10 50 500 1000	8 48 498 998	8 48 498 998	16 19 21 17	11 14 15 13	10 11 10 10
NLMS [50] (wrong N)	64 484	36 400	49 441	26 81	26 94	19 31

Table 5: Main iterations 5

test problem	n	n′	ns	alg1	alg2	alg3
Tridia [8]	10	10	10	5.00	17.90	4.00
	25	25	25	5.00	15.96	5.00
	50	50	50	5.00	20.98	5.00
	500	500	500	6.00	24.00	5.00
	1000	1000	1000	6.00	24.00	5.00
Shifted Tridia [9]	10	10	10	12.00	42.70	4.90
	50	50	50	19.00	48.72	5.98
	500	500	500	25.00	56.97	7.00
	1000	1000	1000	24.00	53.98	6.00
Extended Rosenbrock [10]	2	2	1	30.00	102.00	4.00
	10	10	9	29.00	68.00	78.00
	50	50	49	29.00	54.00	326.51
	500	500	499	27.00	62.00	nr
	1000	1000	999	28.00	56.00	nr
LMS [11]	25	9	16	14.00	34.50	15.75
	64	36	49	15.00	41.00	17.92
	121	81	100	18.00	42.32	20.96
	484	400	441	22.00	70.73	29.99
NLMS [12]	64	36	49	17.00	43.92	20.92
	484	400	441	25.00	64.39	32.99
Shifted LMS [13]	64	36	49	37.96	86.41	22.43
	484	400	441	78.49	314.22	50.81
Shifted NLMS [14]	64	36	49	43.96	119.00	35.43
	484	400	441	147.49	358.31	84.81
Nondia [15]	10	10	9	31.00	92.00	4.00
	20	20	19	31.00	115.00	4.00
	30	30	29	32.00	114.00	4.00
	50	50	49	32.00	115.00	4.00
	500	500	499	33.00	113.00	5.00
	1000	1000	999	33.00	113.00	5.00
Boundary Value [16]	10	8	8	19.00	55.75	14.25
	20	18	18	21.00	56.78	18.11
	30	28	28	21.00	56.14	21.07
	50	48	48	26.00	60.75	25.04
	100	98	98	56.00	68.43	58.02

Table 6: Function and gradient evaluations 1

test problem	n	n'	ns	alg1	alg2	alg3
Broyden Tridiagonal [17]	10	8	8	21.00	51.25	13.25
	50	48	48	21.00	45.50	14.04
	500	448	448	25.00	52.06	13.00
	1000	998	998	21.00	54.03	13.00
Broyden Banded [18]	10	10	10	50.00	189.20	30.40
	50	50	50	37.00	211.68	33.68
	100	100	100	34.00	224.68	32.84
	250	250	250	39.00	218.58	34.94
Powell Singular [19]	4	4	1	61.00	204.00	43.00
	10	10	3	64.00	210.00	75.00
	49	49	16	71.00	220.00	95.00
	100	100	33	93.00	240.00	92.00
	502	502	167	80.00	255.00	143.00
Wrong Wood [20]	10	10	4	36.00	164.00	29.00
	50	50	24	38.00	165.17	34.00
	500	500	249	48.50	189.33	41.80
	1000	1000	499	48.73	179.53	40.00
Gaussian [21]	10	10	8	34.00	35.13	4.00
	50	50	48	44.00	25.54	17.00
	500	500	498	26.00	83.94	19.00
	1000	1000	998	67.00	121.94	66.00
Diagonal Quadratic [22]	10	10	8	19.00	67.00	5.00
	50	50	48	19.00	59.00	5.00
	500	500	498	18.00	63.00	5.00
	1000	1000	998	18.00	63.00	5.00
Spiked LMS [23, bounds]	64	36	49	19.00	46.67	20.92
	484	400	441	22.00	70.71	28.99
Plane LMS [24, bounds]	64	36	49	19.59	46.90	23.10
	484	400	441	23.80	82.70	31.74
Spiked NLMS [25, bounds]	64	36	49	20.00	62.35	27.92
	484	400	441	36.00	94.66	36.99
Plane NLMS [26, bounds]	64	36	49	20.45	61.18	25.04
	484	400	441	27.76	80.70	32.73

Table 7: Function and gradient evaluations 2

test problem	n	n′	ns	alg1	alg2	alg3
Bounded Wood [27, bounds]	10	10	4	16.00	50.00	7.00
	50	50	24	17.00	59.13	8.00
	500	500	249	16.00	54.00	8.00
	1000	1000	499	17.00	54.00	8.00
Paviani [28, bounds]	10	10	1	9.00	68.00	18.00
	52	52	7	9.00	68.00	18.00
	500	500	70	10.00	79.00	18.00
	997	997	140	10.00	79.00	18.00
McCormick [29, bounds]	2	2	1	10.00	18.00	10.00
	10	10	9	12.00	27.00	10.00
	50	50	49	13.00	30.00	10.00
	500	500	499	13.00	30.00	11.00
	1000	1000	999	12.05	30.00	11.00
Extended Wood [30]	4	4	1	47.00	194.00	133.00
	10	10	4	45.00	189.00	180.00
	50	50	24	48.00	205.04	246.00
	500	500	249	59.00	238.58	nr
	1000	1000	499	63.00	243.10	nr
Extended ENGVL1 [31]	10	10	9	17.00	34.44	14.00
	50	50	49	21.00	39.14	15.00
	500	500	499	20.00	39.02	15.00
	1000	1000	999	19.05	39.01	15.00
Extended CRGLVY [32]	10	10	4	76.00	320.00	71.00
	50	50	24	65.00	240.00	86.00
	500	500	249	69.90	517.85	78.00
	1000	1000	499	80.65	no conv	89.00
Extended Freudenstein and Roth [33]	10	10	9	31.00	69.00	15.00
	50	50	49	33.00	72.00	15.00
	500	500	499	46.48	68.00	15.00
	1000	1000	999	31.03	68.00	15.00
Extended Powell Badly Scaled [34]	10	10	9	28.00	70.00	26.00
	50	50	49	28.00	70.00	32.00
	500	500	499	28.00	70.00	36.00
	1000	1000	999	28.00	70.00	40.00
Extended SCHMVT [35]	10	10	8	14.00	45.50	13.00
	50	50	48	13.00	42.40	13.00
	500	500	498	13.00	48.97	14.00
	1000	1000	998	14.00	48.61	14.00

Table 8: Function and gradient evaluations 3

test problem	n	n′	ns	alg1	alg2	alg3
Extended Cube [36]	2	2	1	72.00	200.00	112.00
	10	10	9	160.00	427.00	255.00
	50	50	49	539.78	nr	nr
Bounded 3D-Broyden [37, bounds]	10	8	8	5.50	13.25	7.75
	50	48	48	11.17	23.54	22.38
	500	498	498	11.44	24.86	72.45
	1000	998	998	11.96	24.93	78.70
Bounded CRGLVY [38, bounds]	10	10	4	14.00	51.75	22.00
	50	50	24	15.00	59.38	31.00
	500	500	249	15.00	54.03	120.94
	1000	1000	499	15.00	54.02	176.40
Bounded ENGLV1 [39, bounds]	10	10	9	8.33	11.67	6.67
	50	50	49	8.49	11.94	6.94
	500	500	499	8.05	11.99	6.99
	1000	1000	999	8.02	12.00	7.00
Bounded Freud. Roth [40, bounds]	10	10	9	28.00	61.00	12.00
	50	50	49	32.00	95.71	12.00
	500	500	499	44.00	116.04	12.00
	1000	1000	999	39.00	195.06	13.00
Bounded SCHMVT [41, bounds]	10	10	8	11.00	37.00	11.00
	50	50	48	14.00	46.00	12.00
	500	500	498	13.00	41.00	12.00
	1000	1000	998	13.00	41.00	12.00
Bounded Rosenbrock [42, bounds]	10	10	9	19.78	44.78	30.00
	50	50	49	49.86	117.37	33.00
	100	100	99	69.66	201.42	35.00

Table 9: Function and gradient evaluations 4

test problem	n	n'	ns	alg1	alg2	alg3
Bounded	10	10	9	3.60	11.30	3.80
TRIDIA	50	50	49	3.92	14.58	4.96
[43, bounds]	500	500	499	4.00	14.97	5.00
	1000	1000	999	4.00	16.99	5.00
Plane B SLMS	64	36	49	24.76	82.04	31.84
[44, bounds]	484	400	441	119.15	281.93	86.11
Plane B SNLMS	64	36	49	29.88	86.33	35.65
[45, bounds]	484	400	441	129.80	461.05	92.01
Prob. penalty	10	10	10	13.30	ls.fail.	11.40
[46]	50	50	50	10.02	ls.fail.	no conv.
	100	100	100	13.79	ls.fail.	no conv.
PSPDOC [47]	4	4	2	8.00	40.00	10.00
Extended	2	2	1	27.00	72.00	51.00
Rosenbrock #	10	10	9	83.00	233.00	99.00
[48]	50	50	49	121.00	321.00	142.00
	100	100	99	242.00	586.19	254.00
Extended	10	8	8	24.00	53.00	14.00
3D-Broyden	50	48	48	28.00	65.00	15.00
[49]	500	498	498	26.00	69.04	14.00
(wrong N)	1000	998	998	22.00	54.00	14.00
NLMS [50]	64	36	49	31.20	133.29	29.80
(wrong N)	484	400	441	113.73	578.16	48.60

Table 10: Function and gradient evaluations 5

test problem	n	n'	ns	alg1	alg2	alg3
Tridia [8]	10	10	10	20	32	15
	25	25	25	34	36	53
	50	50	50	41	32	40
	500	500	500	43	41	42
	1000	1000	1000	43	38	43
Shifted Tridia [9]	10	10	10	56	56	15
	50	50	50	113	81	38
	500	500	500	239	102	48
	1000	1000	1000	194	107	40
Extended Rosenbrock [10]	2	2	1	29	36	2
	10	10	9	123	121	389
	50	50	49	137	96	1873
	500	500	499	134	126	nr
	1000	1000	999	153	108	nr
LMS [11]	25	9	16	52	41	65
	64	36	49	118	93	124
	121	81	100	191	124	204
	484	400	441	365	395	561
NLMS [12]	64	36	49	136	127	173
	484	400	441	496	367	691
Shifted LMS [13]	64	36	49	212	184	148
	484	400	441	931	848	1211
Shifted NLMS [14]	64	36	49	396	187	412
	484	400	441	1589	1374	2515
Nondia [15]	10	10	9	34	39	2
	20	20	19	34	47	2
	30	30	29	32	46	2
	50	50	49	32	47	2
	500	500	499	34	43	4
	1000	1000	999	34	43	4
Boundary Value [16]	10	8	8	97	82	82
	20	18	18	252	195	223
	30	28	28	373	273	393
	50	48	48	764	488	867
	100	98	98	4360	1004	4923

Table 11: Conjugate gradient iterations 1

test problem	n	n'	ns	alg1	alg2	alg3
Broyden	10	8	8	71	58	52
Tridiagonal	50	48	48	73	56	65
[17]	500	448	448	88	59	53
	1000	998	998	80	74	53
Broyden	10	10	10	101	76	65
Banded	50	50	50	128	107	82
[18]	100	100	100	119	135	74
	250	250	250	143	125	81
Powell	4	4	1	193	139	135
Singular	10	10	3	410	285	485
[19]	49	49	16	1649	770	659
	100	100	33	2134	1184	921
	502	502	167	13994	5000	1053
Wrong Wood	10	10	4	118	139	87
[20]	50	50	24	174	157	148
	500	500	249	341	226	327
	1000	1000	499	335	203	297
Gaussian	10	10	8	72	59	9
[21]	50	50	48	116	45	80
	500	500	498	58	45	126
	1000	1000	998	119	62	267
Diagonal	10	10	8	28	25	1
Quadratic	50	50	48	34	18	1
[22]	500	500	498	30	20	1
	1000	1000	998	30	20	1
Spiked LMS	64	36	49	141	104	166
[23, bounds]	484	400	441	448	460	625
Plane LMS	64	36	49	91	96	132
[24, bounds]	484	400	441	360	492	696
Spiked NLMS	64	36	49	162	133	254
[25, bounds]	484	400	441	768	775	1274
Plane NLMS	64	36	49	119	105	175
[26, bounds]	484	400	441	647	955	1206

Table 12: Conjugate gradient iterations 2

test problem	n	n′	ns	alg1	alg2	alg3
Bounded Wood	10	10	4	15	12	6
[27, bounds]	50	50	24	20	16	8
	500	500	249	16	12	8
	1000	1000	499	16	12	8
Paviani	10	10	1	6	5	6
[28, bounds]	52	52	7	6	5	6
	500	500	70	7	6	6
	997	997	140	7	6	6
McCormick	2	2	1	5	7	18
[29, bounds]	10	10	9	28	16	21
	50	50	49	30	18	42
	500	500	499	32	19	47
	1000	1000	999	31	19	41
Extended	4	4	1	125	119	291
Wood	10	10	4	245	206	886
[30]	50	50	24	508	383	1705
	500	500	249	1158	864	nr
	1000	1000	499	1675	996	nr
Extended	10	10	9	28	19	30
ENGVL1	50	50	49	49	24	37
[31]	500	500	499	43	21	37
	1000	1000	999	43	21	33
Extended	10	10	4	420	407	358
CRGLVY	50	50	24	801	739	627
[32]	500	500	249	1049	1147	598
	1000	1000	499	1948	no conv	813
Extended	10	10	9	66	52	38
Freudenstein	50	50	49	85	64	37
and Roth	500	500	499	123	51	36
[33]	1000	1000	999	46	51	34
Extended	10	10	9	21	31	23
Powell	50	50	49	21	46	39
Badly Scaled	500	500	499	21	44	44
[34]	1000	1000	999	21	35	52
Extended	10	10	8	65	46	67
SCHMVT	50	50	48	76	44	73
[35]	500	500	498	78	45	93
	1000	1000	998	89	45	92

Table 13: Conjugate gradient iterations 3

test problem	n	n′	ns	alg1	alg2	alg3
Extended	2	2	1	77	82	10
Cube	10	10	9	524	583	900
[36]	50	50	49	2352	nr	nr
Bounded	10	8	8	15	9	12
3D-Broyden	50	48	48	68	32	51
[37, bounds]	500	498	498	69	31	210
	1000	998	998	65	28	113
Bounded	10	10	4	26	19	41
CRGLVY	50	50	24	28	26	79
[38, bounds]	500	500	249	26	19	463
	1000	1000	499	27	19	895
Bounded	10	10	9	22	3	12
ENGLV1	50	50	49	21	3	13
[39, bounds]	500	500	499	21	3	12
	1000	1000	999	21	3	12
Bounded	10	10	9	45	36	17
Freud. Roth	50	50	49	52	56	17
[40, bounds]	500	500	499	58	59	15
	1000	1000	999	44	89	16
Bounded	10	10	8	28	24	21
SCHMVT	50	50	48	34	30	24
[41, bounds]	500	500	498	32	24	21
	1000	1000	998	31	23	20
Bounded	10	10	9	56	53	69
Rosenbrock	50	50	49	207	215	63
[42, bounds]	100	100	99	415	428	67

Table 14: Conjugate gradient iterations 4

test problem	n	n′	ns	alg1	alg2	alg3
Bounded TRIDIA [43, bounds]	10	10	9	17	25	13
	50	50	49	38	31	39
	500	500	499	39	28	39
	1000	1000	999	40	31	39
Plane B SLMS [44, bounds]	64	36	49	108	101	140
	484	400	441	357	956	903
Plane B SNLMS [45, bounds]	64	36	49	166	102	187
	484	400	441	1215	982	1188
Prob. penalty [46]	10	10	10	12	ls.fail.	9
	50	50	50	21	ls.fail.	no conv.
	100	100	100	26	ls.fail.	no conv.
PSPDOC [47]	4	4	2	15	19	18
Extended Rosenbrock # [48]	2	2	1	34	25	56
	10	10	9	391	391	434
	50	50	49	487	494	525
	100	100	99	1050	887	1097
Extended 3D-Broyden [49] (wrong N)	10	8	8	79	45	14
	50	48	48	121	84	65
	500	498	498	131	78	53
	1000	998	998	105	62	53
NLMS [50] (wrong N)	64	36	49	182	171	141
	484	400	441	1180	1083	1304

Table 15: Conjugate gradient iterations 5

4 A Fortran code for the test problem collection

The following pages give the complete listing of the driver program (PSPI), as well as of the necessary routines (INIELF, RANGE, XLOWER, XUPPER and ELFNCT) needed to run PSPMIN on the test problem collection of the previous section.

The driver PSPI correspond to the algorithm alg1 and alg2 in [2]. A driver for alg3 can be obtained by assigning the value .TRUE. instead of .FALSE. to the logical variable HESDIF in the beginning of PSPI.

```
      PROGRAM PSPI
C
C
C     ********************************************
C     *                                          *
C     *    TEST OF MINIMIZATION ROUTINE PSPMIN   *
C     *                                          *
C     ********************************************
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL FKNOWN,TESTGX,RESTRT,HESDIF
C
      DIMENSION X(1000),WK(17000),ISTATE(3000),NVAR(1001)
      DIMENSION INVAR(7000)
C
      COMMON/TEST/ITEST
C
C
C   Set the length of the work space
C   --------------------------------
C
      LWK=17000
C
C   Define output parameters
C   ------------------------
C
C   1) Output device
C
      IPDEVC=5
C
C   2) Output frequency
C
      IPFREQ=0
C
C   3) Output amount
C
```

```
      IPWHAT=1
C
C   Set the maximum number of function calls
C   ---------------------------------------
C
      NGRMAX=1000
      ITMAX=500
C
C   Define tolerance for anti-zigzag device (Bertsekas)
C   ---------------------------------------------------
C
      EBOUND=1.0D-4
C
C   Initialize the logical variable that asks for testing of the
C   ------------------------------------------------------------
C       analytical gradient
C       -------------------
C
      TESTGX=.FALSE.
C
C   Set the variable that prescribe estimation of the element
C   ---------------------------------------------------------
C        Hessians by differences at the starting point
C       ----------------------------------------------
C
      HESDIF=.FALSE.
C
C   Set the restart parameter to false
C   ----------------------------------
C
      RESTRT=.FALSE.
C
C   Define stepsize for difference estimation of the first
C   ------------------------------------------------------
C        gradient
C        --------
C
      DIFGRD=1.0D-7
C
C   Define the desired precision on the minimization
C   ------------------------------------------------
C
      EPSIL=1.0D-7
C
C   Define the machine precision
C   ----------------------------
C
      EPSMCH=1.0D-18
C
C   Define the overal maximum steplength and the maximum step
C   ---------------------------------------------------------
C       at the first iteration
C       ----------------------
C
      STMAX=1.0D+10
      STINIT=1.D+5
C
C   Define input device
C   -------------------
C
```

```
      INDEV=5
C
C   Read in some parameters
C   ----------------------
C
C    1) Index of test problem considered
C
      IF(INDEV.EQ.5)WRITE(INDEV,200)
200   FORMAT(1H1///5X,'Test problem considered?')
      READ (INDEV,100)ITEST
100   FORMAT(10I)
C
C    2) Dimension
C
      IF(INDEV.EQ.5)WRITE(INDEV,201)
201   FORMAT(//5X,'Dimension?')
      READ (INDEV,100)N
C
C    3) Availability of analytical derivatives
C
      IF(INDEV.EQ.5)WRITE(INDEV,202)
202   FORMAT(//5X,'Are analytical derivatives available ? ',
     1'(y=1,n=0)')
      READ (INDEV,100)IDER
C
C   Initialize the process for the considered test problem
C   ------------------------------------------------------
C
      CALL INIELF(INVAR,NVAR,ISTATE,X,FLOWBD,
     1              FKNOWN,N,NS)
C
C   If the derivatives are not available, modify ISTATE
C   ---------------------------------------------------
C
      IF(IDER.NE.0)GO TO 300
        DO 301 I=N+1,N+NS
          ISTATE(I)=-1
301       CONTINUE
300   CONTINUE
C
C   Minimize
C   --------
C
      CALL PSPMIN(X,FX,EPSIL,INFO,IFLAG,EBOUND,NGRMAX,ITMAX,
     1            FKNOWN,RESTRT,FLOWBD,EPSMCH,
     1            DIFGRD,TESTGX,HESDIF,
     1            STMAX,STINIT,
     1            N,NS,INVAR,NVAR,ISTATE,
     1            IPDEVC,IPFREQ,IPWHAT,LWK,WK)
C
C   End of test
C   -----------
C
      STOP
      END
C
C
C
      SUBROUTINE INIELF(INVAR,NVAR,ISTATE,X,FLOWBD,
     1                  FKNOWN,N,NS)
```

```
C
C
C      **************************************************
C      *                                                *
C      *   Initialization of the IELF-th element function   *
C      *                                                *
C      **************************************************
C
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
       DIMENSION INVAR(1),NVAR(1),ISTATE(1),X(1),WK(1)
       LOGICAL FKNOWN
C
       COMMON/TEST/ITEST
C
C
C   Decide what function is used
C   ----------------------------
C
       GO TO (1001,1002,1002,1004,1005,1006,1007,1008,1008,1010,
     1       1011,1012,1011,1012,1015,1016,1017,1018,1019,1020,
     1       1021,1021,1011,1011,1012,1012,1027,1028,1029,1020,
     1       1031,1032,1033,1034,1035,1036,1017,1032,1031,1033,
     1       1035,1010,1008,1011,1012,1047,1048,1010,1017,1012
     1       )ITEST
C
C   First trial trivial problem (lower bound)
C   -----------------------------------------
C
1001   CONTINUE
       FKNOWN=.FALSE.
       N=3
       NS=2
       X(1)=10.0
       X(2)=4.0
       X(3)=10.0
       NVAR(1)=1
       NVAR(2)=2
       NVAR(3)=4
       INVAR(1)=1
       INVAR(2)=2
       INVAR(3)=3
       ISTATE(1)=1
       ISTATE(2)=-1
       ISTATE(3)=-1
       ISTATE(4)=1
       ISTATE(5)=1
       FLOWBD=0.0
       RETURN
C
C   Unconstrained trivial quadratic test problem
C   --------------------------------------------
C
1002   CONTINUE
       FKNOWN=.FALSE.
       N=3
       NS=3
       X(1)=-1.0
       X(2)=4.0
```

```
      X(3)=-1.0
      NVAR(1)=1
      NVAR(2)=2
      NVAR(3)=4
      NVAR(4)=6
      INVAR(1)=1
      INVAR(2)=1
      INVAR(3)=2
      INVAR(4)=2
      INVAR(5)=3
      ISTATE(1)=-1
      ISTATE(2)=-1
      ISTATE(3)=-1
      ISTATE(4)=1
      ISTATE(5)=1
      ISTATE(6)=1
      FLOWBD=0.0
      RETURN
C
C   Trivial test problem without analytical derivatives
C   ---------------------------------------------------
C
1004  CONTINUE
      FKNOWN=.FALSE.
      N=4
      NS=2
      X(1)=-1.0
      X(2)=4.0
      X(3)=-1.0
      X(4)=-4.0
      NVAR(1)=1
      NVAR(2)=3
      NVAR(3)=7
      INVAR(1)=1
      INVAR(2)=3
      INVAR(3)=1
      INVAR(4)=2
      INVAR(5)=3
      INVAR(6)=4
      ISTATE(1)=-1
      ISTATE(2)=-1
      ISTATE(3)=-1
      ISTATE(4)=-1
      ISTATE(5)=-1
      ISTATE(6)=-1
      FLOWBD=2.0
      RETURN
C
C   Trivial linear problem
C   ----------------------
C
1005  CONTINUE
      FKNOWN=.FALSE.
      N=3
      NS=2
      X(1)=-4.0
      X(2)=-5.0
      X(3)=100.0
      NVAR(1)=1
      NVAR(2)=3
```

```
      NVAR(3)=5
      INVAR(1)=1
      INVAR(2)=2
      INVAR(3)=2
      INVAR(4)=3
      ISTATE(1)=1
      ISTATE(2)=1
      ISTATE(3)=1
      ISTATE(4)=-1
      ISTATE(5)=-1
      FLOWBD=-1.0D+30
      RETURN
C
C   Mixed trivial test problem
C   -------------------------
1006  CONTINUE
      FKNOWN=.FALSE.
      N=4
      NS=2
      X(1)=7.8
      X(2)=3.4
      X(3)=-2.0
      X(4)=-9.0
      NVAR(1)=1
      NVAR(2)=3
      NVAR(3)=7
      INVAR(1)=1
      INVAR(2)=2
      INVAR(3)=1
      INVAR(4)=2
      INVAR(5)=3
      INVAR(6)=4
      ISTATE(1)=1
      ISTATE(2)=-1
      ISTATE(3)=-1
      ISTATE(4)=-1
      ISTATE(5)=-1
      ISTATE(6)=-1
      FLOWBD=0.0
      RETURN
C
C   Non quadratic easy test problem
C   ------------------------------
C
1007  CONTINUE
      FKNOWN=.FALSE.
      N=4
      NS=2
      X(1)=7.8
      X(2)=3.4
      X(3)=-2.0
      X(4)=-9.0
      NVAR(1)=1
      NVAR(2)=3
      NVAR(3)=7
      INVAR(1)=1
      INVAR(2)=2
      INVAR(3)=1
      INVAR(4)=2
      INVAR(5)=3
```

```
      INVAR(6)=4
      ISTATE(1)=1
      ISTATE(2)=-1
      ISTATE(3)=-1
      ISTATE(4)=-1
      ISTATE(5)=-1
      ISTATE(6)=-1
      FLOWBD=0.0
      RETURN
C
C   TRIDIA, STRID and HSTRID
C   ------------------------
C
1008  CONTINUE
      FKNOWN=.FALSE.
      NS=N
      FLOWBD=0.0
      L=1
      NVAR(1)=1
      INVAR(1)=1
      ISTATE(N+1)=1
      DO 80 I=2,NS
        NVAR(I)=I+I-2
        L=L+1
        INVAR(L)=I-1
        L=L+1
        INVAR(L)=I
        IN=N+I
        ISTATE(IN)=1
80      CONTINUE
      NVAR(N+1)=N+N
      ITEMP=-1
      IF(ITEST.EQ.43)ITEMP=1
      DO 81 I=1,N
        X(I)=-1.0
        ISTATE(I)=ITEMP
81      CONTINUE
      RETURN
C
C   Extended Rosenbrock
C   -------------------
C
1010  CONTINUE
      FKNOWN=.FALSE.
      NS=N-1
      FLOWBD=0.0
      DO 90 I=1,NS
        IP=I+I-1
        NVAR(I)=IP
        INVAR(IP)=I
        INVAR(IP+1)=I+1
        ISTATE(N+I)=1
90      CONTINUE
      NVAR(NS+1)=NS+NS+1
      ITEMP=-1
      IF(ITEST.EQ.42)ITEMP=1
      DN=N+1.0
      DO 91 I=1,N
        ISTATE(I)=ITEMP
        IF(ITEST.EQ.10.OR.ITEST.EQ.42)X(I)=-1.0
```

```
          IF(ITEST.EQ.49)X(I)=DFLOAT(I)/DN
91        CONTINUE
        RETURN
C
C   Linear Minimum Surface
C   ---------------------
C
1011    CONTINUE
        FKNOWN=.FALSE.
        DN=N+0.1
        MLMS=DSQRT(DN)
        MO=MLMS-1
        NS=MO*MO
        FLOWBD=1.0
        DO 100 I=1,N
          X(I)=0.0
          ISTATE(I)=-1
100       CONTINUE
        DO 101 I=1,MLMS
          ISTATE(I)=0
          TEMP=(I-1.0)/MO
          X(I)=1.0+4.0*TEMP
          L=MLMS*(I-1)+1
          ISTATE(L)=0
          X(L)=1.0+8.0*TEMP
          L=I+MLMS*MO
          ISTATE(L)=0
          X(L)=9.0+4.0*TEMP
          L=I*MLMS
          ISTATE(L)=0
          X(L)=5.0+8.0*TEMP
101       CONTINUE
        NVAR(1)=1
        ISTATE(N+1)=1
        DO 103 I=2,NS
          NVAR(I)=4*(I-1)+1
          ISTATE(N+I)=1
103       CONTINUE
        NVAR(NS+1)=4*NS+1
        L=1
        K=0
        DO 104 J=1,MO
          DO 105 I=1,MO
            K=K+1
            INVAR(L)=K
            INVAR(L+1)=K+1
            INVAR(L+2)=K+MLMS
            INVAR(L+3)=K+MLMS+1
            L=L+4
105         CONTINUE
          K=K+1
104       CONTINUE
        IF(ITEST.EQ.11.OR.ITEST.EQ.13)RETURN
        GO TO 1023
C
C   Nonlinear Minimum Surface
C   ------------------------
C
1012    CONTINUE
        FKNOWN=.FALSE.
```

```
      DN=N+0.1
      MLMS=DSQRT(DN)
      MO=MLMS-1
      NS=MO*MO
      FLOWBD=1.0
      DO 120 I=1,N
        X(I)=0.0
        ISTATE(I)=-1
120     CONTINUE
      DO 121 I=1,MLMS
        ISTATE(I)=0
        TEMP=(I-1.0)/MO
        X(I)=1.0+4.0*TEMP+10.0*(TEMP+1.0)**2
        L=MLMS*(I-1)+1
        ISTATE(L)=0
        X(L)=1.0+8.0*TEMP+10.0*(1.0-TEMP)**2
        L=I+MLMS*MO
        ISTATE(L)=0
        X(L)=9.0+4.0*TEMP+10.0*TEMP**2
        L=I*MLMS
        ISTATE(L)=0
        X(L)=5.0+8.0*TEMP+10.0*(2.0-TEMP)**2
121     CONTINUE
      NVAR(1)=1
      ISTATE(N+1)=1
      DO 123 I=2,NS
        NVAR(I)=4*(I-1)+1
        ISTATE(N+I)=1
123     CONTINUE
      NVAR(NS+1)=4*NS+1
      L=1
      K=0
      DO 124 J=1,MO
        DO 125 I=1,MO
          K=K+1
          INVAR(L)=K
          INVAR(L+1)=K+1
          INVAR(L+2)=K+MLMS
          INVAR(L+3)=K+MLMS+1
          L=L+4
125       CONTINUE
        K=K+1
124     CONTINUE
      IF(ITEST.EQ.12.OR.ITEST.EQ.14)RETURN
      GO TO 1023
C
C   NONDIAGONAL ROSENBROCK FUNCTION :NDROS
C
1015  CONTINUE
      FKNOWN=.FALSE.
      NS=N-1
      FLOWBD=0.0
      DO 150 I=1,NS
        IP=I+I-1
        NVAR(I)=IP
        INVAR(IP)=1
        INVAR(IP+1)=I+1
        ISTATE(N+I)=1
150     CONTINUE
      NVAR(NS+1)=NS+NS+1
```

```
      DO 151 I=1,N
        ISTATE(I)=-1
        X(I)=-1.0
151     CONTINUE
      RETURN
C
C   Discrete boundary value problem
C   ------------------------------
C
1016  CONTINUE
      FKNOWN=.FALSE.
      NS=N-2
      IF(NS.LE.0)STOP 'NS le 0!'
      FLOWBD=0.0
      DO 160 I=1,NS
        IP=3*I-2
        NVAR(I)=IP
        INVAR(IP)=I
        INVAR(IP+1)=I+1
        INVAR(IP+2)=I+2
        ISTATE(N+I)=1
160     CONTINUE
      NVAR(NS+1)=3*NS+1
      H=1.0/(N-1.0)
      DO 161 I=2,N-1
        ISTATE(I)=-1
        X(I)=I*H*(I*H-1.0)
161     CONTINUE
      X(1)=0.0
      ISTATE(1)=0
      X(N)=0.0
      ISTATE(N)=0
      RETURN
C
C   Broyden Tridiagonal Nonlinear System
C   ------------------------------------
C
1017  CONTINUE
      FKNOWN=.FALSE.
      FLOWBD=0.0
      NS=N-2
      IF(NS.LE.0)STOP 'NS le 0!'
      DO 170 I=1,NS
        IP=3*I-2
        NVAR(I)=IP
        INVAR(IP)=I
        INVAR(IP+1)=I+1
        INVAR(IP+2)=I+2
        ISTATE(N+I)=1
170     CONTINUE
      NVAR(NS+1)=3*NS+1
      ITEMP=-1
      XTEMP=-1.0
      IF(ITEST.NE.37)GO TO 172
        ITEMP=1
172   CONTINUE
      DO 171 I=2,N-1
        X(I)=XTEMP
        ISTATE(I)=ITEMP
171     CONTINUE
```

```
      X(1)=0.0
      ISTATE(1)=0
      X(N)=0.0
      ISTATE(N)=0
      RETURN
C
C   Broyden Banded Function
C   -----------------------
C
1018  CONTINUE
      FKNOWN=.FALSE.
      FLOWBD=0.0
      NS=N
      IP=1
      DO 180 I=1,NS
        NVAR(I)=IP
        ILOW=MAX0(1,I-5)
        IUP=MIN0(N,I+1)
        DO 182 J=ILOW,IUP
          INVAR(IP)=J
          IP=IP+1
182       CONTINUE
        ISTATE(N+I)=1
180     CONTINUE
      NVAR(NS+1)=IP
      DO 181 I=1,N
        ISTATE(I)=-1
        X(I)=-1.0
181     CONTINUE
      RETURN
C
C   Extended Powell Singular Function
C   ---------------------------------
C
1019  CONTINUE
      FKNOWN=.FALSE.
      FLOWBD=0.0
      NS=(N-1)/3
      NVAR(1)=1
      INVAR(1)=1
      INVAR(2)=2
      INVAR(3)=3
      INVAR(4)=4
      ISTATE(N+1)=1
      IF(NS.EQ.1)GO TO 191
      JJ=4
      DO 190 I=2,NS
        IP=4*I-3
        NVAR(I)=IP
        INVAR(IP)=JJ
        INVAR(IP+1)=JJ+1
        INVAR(IP+2)=JJ+2
        INVAR(IP+3)=JJ+3
        JJ=JJ+3
        ISTATE(N+I)=1
190     CONTINUE
191   CONTINUE
      NVAR(NS+1)=4*NS+1
      X(1)=3.0
      ISTATE(1)=-1
```

```
      DO 192 I=2,N,3
        X(I)=-1.0
        X(I+1)=0.0
        X(I+2)=1.0
        ISTATE(I)=-1
        ISTATE(I+1)=-1
        ISTATE(I+2)=-1
192     CONTINUE
      RETURN
C
C   Extended Wood Function
C   ----------------------
C
1020  CONTINUE
      FKNOWN=.FALSE.
      FLOWBD=0.0
      NS=(N-2)/2
      DO 200 I=1,NS
        IP=4*I-3
        NVAR(I)=IP
        INVAR(IP)=I+I-1
        INVAR(IP+1)=I+I
        INVAR(IP+2)=I+I+1
        INVAR(IP+3)=I+I+2
        ISTATE(N+I)=1
200     CONTINUE
      NVAR(NS+1)=4*NS+1
      DO 201 I=1,N,2
        X(I)=-3.0
        X(I+1)=-1.0
        ISTATE(I)=-1
        ISTATE(I+1)=-1
201     CONTINUE
      RETURN
C
C   Gaussian-like Problem, Diagonal Quadratic
C   ----------------------------------------
C
1021  CONTINUE
      FKNOWN=.FALSE.
      FLOWBD=-1.D30
      NS=N-2
      IP=1
      DO 210 I=1,NS
        NVAR(I)=IP
        INVAR(IP)=I
        INVAR(IP+1)=I+1
        INVAR(IP+2)=I+2
        IP=IP+3
        ISTATE(N+I)=1
210     CONTINUE
      NVAR(NS+1)=IP
      DO 211 I=1,N
        X(I)=3.0
        ISTATE(I)=-1
211     CONTINUE
      RETURN
C
C   Constraint sets for bounded minimum surface problems
C   ----------------------------------------------------
```

```
C
1023    CONTINUE
        NMID=MLMS/2
        NMID=NMID*MLMS+NMID+1
        ISTATE(NMID)=1
        IF(ITEST.EQ.23.OR.ITEST.EQ.25)RETURN
        ITEMP=NMID-1
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID+1
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID-MLMS
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID-MLMS-1
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID-MLMS+1
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID+MLMS
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID+MLMS-1
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        ITEMP=NMID+MLMS+1
        IF(ISTATE(ITEMP).NE.0)ISTATE(ITEMP)=1
        RETURN
C
C   Extended Wood's Bounded Problem
C   -------------------------------
C
1027    CONTINUE
        FKNOWN=.FALSE.
        FLOWBD=0.0
        NS=(N-2)/2
        DO 270 I=1,NS
          IP=4*I-3
          NVAR(I)=IP
          INVAR(IP)=I+I-1
          INVAR(IP+1)=I+I
          INVAR(IP+2)=I+I+1
          INVAR(IP+3)=I+I+2
          IN=N+I
          ISTATE(IN)=1
270       CONTINUE
        NVAR(NS+1)=4*NS+1
        TEMP=-3.0
        IF(ITEST.EQ.27)TEMP=10.0
        DO 271 I=1,N,2
          X(I)=TEMP
          X(I+1)=-1.0
          ISTATE(I)=1
          ISTATE(I+1)=1
271       CONTINUE
        RETURN
C
C   Paviani's Bounded Problem
C   -------------------------
C
1028    CONTINUE
        FKNOWN=.FALSE.
        FLOWBD=-1.0D+34
        NS=(N-3)/7
        JJ=0
```

```
      DO 280 I=1,NS
        IP=10*I-9
        NVAR(I)=IP
        INVAR(IP)=JJ+1
        IP=IP+1
        INVAR(IP)=JJ+2
        IP=IP+1
        INVAR(IP)=JJ+3
        IP=IP+1
        INVAR(IP)=JJ+4
        IP=IP+1
        INVAR(IP)=JJ+5
        IP=IP+1
        INVAR(IP)=JJ+6
        IP=IP+1
        INVAR(IP)=JJ+7
        IP=IP+1
        INVAR(IP)=JJ+8
        IP=IP+1
        INVAR(IP)=JJ+9
        IP=IP+1
        INVAR(IP)=JJ+10
        JJ=JJ+7
        IN=N+I
        ISTATE(IN)=1
280     CONTINUE
      ITEMP=NS+1
      NVAR(ITEMP)=10*NS+1
      DO 281 I=1,N
        X(I)=9.0
        ISTATE(I)=1
281     CONTINUE
      RETURN
C
C   McCormick's Bounded Problem
C   --------------------------
C
1029  CONTINUE
      FKNOWN=.FALSE.
      FLOWBD=-5.0
      NS=N-1
      DO 290 I=1,NS
        IP=2*I-1
        NVAR(I)=IP
        INVAR(IP)=I
        IP=IP+1
        INVAR(IP)=I+1
        IN=N+I
        ISTATE(IN)=1
290     CONTINUE
      NVAR(N)=2*NS+1
      DO 291 I=1,N
        X(I)=0.0
        ISTATE(I)=1
291     CONTINUE
      RETURN
C
C   Extended ENGVL1 Problem
C   ----------------------
C
```

```
1031    CONTINUE
        FKNOWN=.FALSE.
        NS=N-1
        FLOWBD=0.0
        DO 310 I=1,NS
          IP=I+I-1
          NVAR(I)=IP
          INVAR(IP)=I
          INVAR(IP+1)=I+1
          ISTATE(N+I)=1
310       CONTINUE
        NVAR(NS+1)=NS+NS+1
        ITEMP=-1
        IF(ITEST.EQ.39)ITEMP=1
        DO 311 I=1,N
          ISTATE(I)=ITEMP
          X(I)=1.0
311       CONTINUE
        RETURN
C
C   Extended CRGLVY
C   ---------------
C
1032    CONTINUE
        FKNOWN=.FALSE.
        FLOWBD=0.0
        NS=(N-2)/2
        DO 320 I=1,NS
          IP=4*I-3
          NVAR(I)=IP
          INVAR(IP)=I+I-1
          INVAR(IP+1)=I+I
          INVAR(IP+2)=I+I+1
          INVAR(IP+3)=I+I+2
          ISTATE(N+I)=1
320       CONTINUE
        NVAR(NS+1)=4*NS+1
        ITEMP=-1
        IF(ITEST.EQ.38)ITEMP=1
        DO 321 I=1,N,2
          X(I)=2.0
          X(I+1)=2.0
          ISTATE(I)=ITEMP
          ISTATE(I+1)=ITEMP
321       CONTINUE
        RETURN
C
C   Extended Freudenstein and Roth Problem
C   --------------------------------------
C
1033    CONTINUE
        FKNOWN=.FALSE.
        NS=N-1
        FLOWBD=0.0
        DO 330 I=1,NS
          IP=I+I-1
          NVAR(I)=IP
          INVAR(IP)=I
          INVAR(IP+1)=I+1
          ISTATE(N+I)=1
```

```
330       CONTINUE
        NVAR(NS+1)=NS+NS+1
        ITEMP=-1
        IF(ITEST.EQ.40)ITEMP=1
        DO 331 I=1,N
          ISTATE(I)=ITEMP
          X(I)=-2.0
331       CONTINUE
        RETURN
C
C   Extended Powell Badly Scaled Problem
C   ------------------------------------
C
1034    CONTINUE
        FKNOWN=.FALSE.
        NS=N-1
        FLOWBD=0.0
        DO 340 I=1,NS
          IP=I+I-1
          NVAR(I)=IP
          INVAR(IP)=I
          INVAR(IP+1)=I+1
          ISTATE(N+I)=1
340       CONTINUE
        NVAR(NS+1)=NS+NS+1
        DO 341 I=1,N
          ISTATE(I)=-1
          X(I)=1.0
341       CONTINUE
        RETURN
C
C   Extended SCHMVT Problem
C   -----------------------
C
1035    CONTINUE
        FKNOWN=.FALSE.
        NS=N-2
        FLOWBD=0.0
        DO 350 I=1,NS
          IP=3*I-2
          NVAR(I)=IP
          INVAR(IP)=I
          IP=IP+1
          INVAR(IP)=I+1
          IP=IP+1
          INVAR(IP)=I+2
          ISTATE(N+I)=1
350       CONTINUE
        NVAR(NS+1)=3*NS+1
        ITEMP=-1
        IF(ITEST.EQ.41)ITEMP=1
        DO 351 I=1,N
          ISTATE(I)=ITEMP
          X(I)=0.5
351       CONTINUE
        RETURN
C
C   Extended Cube Problem
C   ---------------------
C
```

```
1036    CONTINUE
        FKNOWN=.FALSE.
        NS=N-1
        FLOWBD=0.0
        DO 360 I=1,NS
          IP=I+I-1
          NVAR(I)=IP
          INVAR(IP)=I
          INVAR(IP+1)=I+1
          ISTATE(N+I)=1
360       CONTINUE
        NVAR(NS+1)=NS+NS+1
        DO 361 I=1,N
          ISTATE(I)=-1
          X(I)=-2.0
361       CONTINUE
        RETURN
C
C   Penalty Function Problem
C   ------------------------
C
1047    CONTINUE
        FKNOWN=.FALSE.
        NS=N
        FLOWBD=0.0
        DO 470 I=1,NS-1
          IP=2*I-1
          NVAR(I)=IP
          INVAR(IP)=I
          INVAR(IP+1)=I+1
          ISTATE(N+I)=1
470       CONTINUE
        NVAR(NS)=2*NS-1
        DO 471 I=1,N
          INVAR(NVAR(NS)+I-1)=I
471       CONTINUE
        ISTATE(N+NS)=1
        NVAR(NS+1)=3*N-1
        DO 472 I=1,N
          X(I)=0.0
          ISTATE(I)=1
472       CONTINUE
        RETURN
C
C   PSPDOC Example
C   --------------
C
1048    CONTINUE
        FKNOWN=.FALSE.
        FLOWBD=0.0
        N=4
        NS=2
        NVAR(1)=1
        NVAR(2)=4
        NVAR(3)=7
        INVAR(1)=1
        INVAR(2)=2
        INVAR(3)=3
        INVAR(4)=2
        INVAR(5)=3
```

```
      INVAR(6)=4
      ISTATE(1)=1
      ISTATE(2)=-1
      ISTATE(3)=-1
      ISTATE(4)=-1
      ISTATE(5)=1
      ISTATE(6)=1
      X(1)=3.0
      X(2)=3.0
      X(3)=3.0
      X(4)=3.0
      RETURN
      END
C
C
C
C
C
      SUBROUTINE RANGE(IELF,MODE,W1,W2,NDIMI,NSUBI,NS)
C
C
C     *********************************************
C     *                                           *
C     *   Transfers on the range of the Hessian   *
C     *           of the IELF-th element          *
C     *                                           *
C     *********************************************
C
C     MODE=1 <=> U*W1=W2
C     MODE=2 <=> U*W2=W1
C     MODE=3 <=> U'*W1=W2
C     MODE=4 <=> U'*W2=W1
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION W1(1),W2(1)
C
C
      COMMON/TEST/ITEST
C
C
C   Decide what test problem is considered
C   -------------------------------------
C
      GO TO (1002,1001,1003,1004,1005,1006,1006,1008,1001,1001,
     1       1011,1011,1013,1013,1001,1016,1017,1001,1001,1001,
     1       1021,1001,1011,1011,1011,1011,1001,1001,1001,1001,
     1       1001,1001,1001,1001,1001,1001,1017,1001,1001,1001,
     1       1001,1001,1008,1013,1013,1047,1048,1001,1001,1001
     1       )ITEST
C
C   No nullspace at all
C   -------------------
C
1001  CONTINUE
      DO 1 I=1,NDIMI
        W2(I)=W1(I)
1       CONTINUE
      NSUBI=NDIMI
```

```
      RETURN
C
C   Correct nullspace for test problem 1
C   ------------------------------------
C
1002  CONTINUE
      IF(IELF.GT.1)GO TO 15
        NSUBI=0
        RETURN
15    CONTINUE
        NSUBI=2
        W2(1)=W1(1)
        W2(2)=W1(2)
        RETURN
C
C   Correct nullspace for test problem 2
C   ------------------------------------
C   1)  Uw1=w2
C
1003  CONTINUE
      GO TO (31,32,33,34)MODE
31    CONTINUE
      NSUBI=1
      IF(IELF.GT.1)GO TO 25
        W2(1)=W1(1)
        RETURN
25    CONTINUE
        W2(1)=W1(1)-W1(2)
        RETURN
C
C   2) U'w1=w2
C
33    CONTINUE
      IF(IELF.GT.1)GO TO 26
        W2(1)=W1(1)
        RETURN
26    CONTINUE
        W2(1)=W1(1)
        W2(2)=-W1(1)
        RETURN
C
C   3) Uw2=w1
C
32    CONTINUE
      IF(IELF.GT.1)GO TO 27
        W2(1)=W1(1)
        RETURN
27    CONTINUE
        W2(1)=0.5*W1(1)
        W2(2)=-W2(1)
        RETURN
C
C   4) U'w2=w1
C
34    CONTINUE
        NSUBI=1
        W2(1)=W1(1)
        RETURN
C
C   Correct nullspace for test problem 4
```

```
C     ------------------------------------
C
1004     CONTINUE
         GO TO (41,42,43,44)MODE
C
C    1)  Uw1=w2
C
41       CONTINUE
         NSUBI=2
         IF(IELF.GT.1)GO TO 45
           W2(1)=W1(1)
           W2(2)=W1(2)
           RETURN
45       CONTINUE
           W2(1)=W1(1)-W1(2)
           W2(2)=W1(3)-W1(4)
           RETURN
C
C    2) U'w1=w2
C
43       CONTINUE
         IF(IELF.GT.1)GO TO 46
           W2(1)=W1(1)
           W2(2)=W1(2)
           RETURN
46       CONTINUE
           W2(1)=W1(1)
           W2(2)=-W1(1)
           W2(3)=W1(2)
           W2(4)=-W1(2)
           RETURN
C
C    3) Uw2=w1
C
42       CONTINUE
         IF(IELF.GT.1)GO TO 47
           W2(1)=W1(1)
           W2(2)=W1(2)
           RETURN
47       CONTINUE
           W2(1)=0.5*W1(1)
           W2(2)=-W2(1)
           W2(3)=0.5*W1(2)
           W2(4)=-W2(3)
           RETURN
C
C    4) U'w2=w1
C
44       CONTINUE
         NSUBI=2
         IF(IELF.GT.1)GO TO 48
           W2(1)=W1(1)
           W2(2)=W1(2)
           RETURN
48       CONTINUE
           W2(1)=W1(1)
           W2(2)=W1(3)
           RETURN
C
C    Trivial linear problem
```

```
C     ----------------------
C
1005     CONTINUE
         NSUBI=0
         RETURN
C
C    Mixed trivial test problem and easy non quadratic
C    -------------------------------------------------
1006     CONTINUE
         GO TO (61,62,63,64)MODE
C
C    1) Uwl=w2
C
61       CONTINUE
         IF(IELF.GT.1)GO TO 65
           NSUBI=1
           W2(1)=W1(2)
           RETURN
65       CONTINUE
           NSUBI=2
           W2(1)=W1(1)-W1(3)
           W2(2)=W1(2)-W1(4)
           RETURN
C
C    2) U'wl=w2
C
63       CONTINUE
         IF(IELF.GT.1)GO TO 66
           W2(1)=0.0
           W2(2)=W1(1)
           RETURN
66       CONTINUE
           W2(1)=W1(1)
           W2(3)=-W1(1)
           W2(2)=W1(2)
           W2(4)=-W1(2)
           RETURN
C
C    3) Uw2=wl
C
62       CONTINUE
           IF(IELF.GT.1)GO TO 67
             W2(1)=W1(1)
             RETURN
67         CONTINUE
             W2(1)=0.5*W1(1)
             W2(2)=0.5*W1(2)
             W2(3)=-W2(1)
             W2(4)=-W2(2)
             RETURN
C
C    4) U'w2=wl
C
64       CONTINUE
           IF(IELF.GT.1)GO TO 68
             NSUBI=1
             W2(1)=W1(1)
             RETURN
68         CONTINUE
             NSUBI=2
```

```
            W2(1)=W1(1)
            W2(2)=W1(2)
            RETURN
C
C   TRIDIA
C   ------
C           :
1008    CONTINUE
        GO TO (81,82,83,84)MODE
C
C   1) Uw1=w2
C
81      CONTINUE
          NSUBI=1
          IF(IELF.EQ.1)GO TO 85
            W2(1)=W1(1)-2.0*W1(2)
            RETURN
85        CONTINUE
            W2(1)=W1(1)
            RETURN
C
C   2) U'w1=w2
C
83      CONTINUE
          IF(IELF.EQ.1)GO TO 86
            W2(1)=W1(1)
            W2(2)=-2.0*W1(1)
            RETURN
86        CONTINUE
            W2(1)=W1(1)
            RETURN
C
C   3) Uw2=w1
C
82      CONTINUE
          IF(IELF.EQ.1)GO TO 89
            W2(1)=0.2*W1(1)
            W2(2)=-0.4*W1(1)
            RETURN
89        CONTINUE
            W2(1)=W1(1)
            RETURN
C
C   4) U'w2=w1
C
84      CONTINUE
          NSUBI=1
          W2(1)=W1(1)
          RETURN
C
C   Minimum Surface Problem
C   -----------------------
C
1011    CONTINUE
        DNS=NS+.1
        MO=DSQRT(DNS)
        GO TO (111,112,113,114)MODE
C
C   1) Uw1=w2
C
```

```
111     CONTINUE
        IF(IELF.LE.MO)GO TO 110
        IF(IELF.GT.(NS-MO))GO TO 115
        IBOUN=MOD(IELF,MO)
        IF(IBOUN.EQ.0)GO TO 116
        IF(IBOUN.EQ.1)GO TO 117
          NSUBI=2
          W2(1)=W1(1)-W1(4)
          W2(2)=W1(2)-W1(3)
          RETURN
110     CONTINUE
          IF(IELF.EQ.1)GO TO 118
          IF(IELF.EQ.MO)GO TO 119
            NSUBI=2
            W2(1)=W1(3)
            W2(2)=W1(4)
            RETURN
118       CONTINUE
            NSUBI=1
            W2(1)=W1(4)
            RETURN
119       CONTINUE
            NSUBI=1
            W2(1)=W1(3)
            RETURN
115     CONTINUE
          IF(IELF.EQ.NS)GO TO 120
          IF(IELF.EQ.(NS-MO+1))GO TO 121
            NSUBI=2
            W2(1)=W1(1)
            W2(2)=W1(2)
            RETURN
121       CONTINUE
            NSUBI=1
            W2(1)=W1(2)
            RETURN
120       CONTINUE
            NSUBI=1
            W2(1)=W1(1)
            RETURN
117     CONTINUE
          NSUBI=2
          W2(1)=W1(2)
          W2(2)=W1(4)
          RETURN
116     CONTINUE
          NSUBI=2
          W2(1)=W1(1)
          W2(2)=W1(3)
          RETURN
C
C   2) U'w1=w2
C
113     CONTINUE
        IF(IELF.LE.MO)GO TO 122
        IF(IELF.GT.(NS-MO))GO TO 123
        IBOUN=MOD(IELF,MO)
        IF(IBOUN.EQ.0)GO TO 124
        IF(IBOUN.EQ.1)GO TO 125
          W2(1)=W1(1)
```

```
          W2(2)=W1(2)
          W2(3)=-W1(2)
          W2(4)=-W1(1)
          RETURN
122     CONTINUE
          IF(IELF.EQ.1)GO TO 126
          IF(IELF.EQ.M0)GO TO 127
            W2(1)=0.0
            W2(2)=0.0
            W2(3)=W1(1)
            W2(4)=W1(2)
            RETURN
126       CONTINUE
            W2(1)=0.0
            W2(2)=0.0
            W2(3)=0.0
            W2(4)=W1(1)
            RETURN
127       CONTINUE
            W2(1)=0.0
            W2(2)=0.0
            W2(3)=W1(1)
            W2(4)=0.0
            RETURN
123     CONTINUE
          IF(IELF.EQ.(NS-M0+1))GO TO 128
          IF(IELF.EQ.NS)GO TO 129
            W2(1)=W1(1)
            W2(2)=W1(2)
            W2(3)=0.0
            W2(4)=0.0
            RETURN
128       CONTINUE
            W2(1)=0.0
            W2(2)=W1(1)
            W2(3)=0.0
            W2(4)=0.0
            RETURN
129       CONTINUE
            W2(1)=W1(1)
            W2(2)=0.0
            W2(3)=0.0
            W2(4)=0.0
            RETURN
124     CONTINUE
          W2(1)=W1(1)
          W2(2)=0.0
          W2(3)=W1(2)
          W2(4)=0.0
          RETURN
125     CONTINUE
          W2(1)=0.0
          W2(2)=W1(1)
          W2(3)=0.0
          W2(4)=W1(2)
          RETURN
C
C   3) Uw2=w1
C
112     CONTINUE
```

```
      IF(IELF.LE.MO)GO TO 2122
      IF(IELF.GT.(NS-MO))GO TO 2123
      IBOUN=MOD(IELF,MO)
      IF(IBOUN.EQ.0)GO TO 2124
      IF(IBOUN.EQ.1)GO TO 2125
        W2(1)=0.5*W1(1)
        W2(2)=0.5*W1(2)
        W2(3)=-W2(2)
        W2(4)=-W2(1)
        RETURN
2122  CONTINUE
        IF(IELF.EQ.1)GO TO 2126
        IF(IELF.EQ.MO)GO TO 2127
          W2(1)=0.0
          W2(2)=0.0
          W2(3)=W1(1)
          W2(4)=W1(2)
          RETURN
2126    CONTINUE
          W2(1)=0.0
          W2(2)=0.0
          W2(3)=0.0
          W2(4)=W1(1)
          RETURN
2127    CONTINUE
          W2(1)=0.0
          W2(2)=0.0
          W2(3)=W1(1)
          W2(4)=0.0
          RETURN
2123  CONTINUE
        IF(IELF.EQ.(NS-MO+1))GO TO 2128
        IF(IELF.EQ.NS)GO TO 2129
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=0.0
          W2(4)=0.0
          RETURN
2128    CONTINUE
          W2(1)=0.0
          W2(2)=W1(1)
          W2(3)=0.0
          W2(4)=0.0
          RETURN
2129    CONTINUE
          W2(1)=W1(1)
          W2(2)=0.0
          W2(3)=0.0
          W2(4)=0.0
          RETURN
2124  CONTINUE
        W2(1)=W1(1)
        W2(2)=0.0
        W2(3)=W1(2)
        W2(4)=0.0
        RETURN
2125  CONTINUE
        W2(1)=0.0
        W2(2)=W1(1)
        W2(3)=0.0
```

```
          W2(4)=W1(2)
          RETURN
C
C   4) U'w2=w1
C
114     CONTINUE
        IF(IELF.LE.M0)GO TO 3122
        IF(IELF.GT.(NS-M0))GO TO 3123
        IBOUN=MOD(IELF,M0)
        IF(IBOUN.EQ.0)GO TO 3124
        IF(IBOUN.EQ.1)GO TO 3125
          NSUBI=2
          W2(1)=W1(1)
          W2(2)=W1(2)
          RETURN
3122    CONTINUE
          IF(IELF.EQ.1)GO TO 3126
          IF(IELF.EQ.M0)GO TO 3127
            NSUBI=2
            W2(1)=W1(3)
            W2(2)=W1(4)
            RETURN
3126      CONTINUE
            NSUBI=1
            W2(1)=W1(4)
            RETURN
3127      CONTINUE
            NSUBI=1
            W2(1)=W1(3)
            RETURN
3123    CONTINUE
          IF(IELF.EQ.(NS-M0+1))GO TO 3128
          IF(IELF.EQ.NS)GO TO 3129
            NSUBI=2
            W2(1)=W1(1)
            W2(2)=W1(2)
            RETURN
3128      CONTINUE
            NSUBI=1
            W2(1)=W1(2)
            RETURN
3129      CONTINUE
            NSUBI=1
            W2(1)=W1(1)
            RETURN
3124    CONTINUE
          NSUBI=2
          W2(1)=W1(1)
          W2(2)=W1(3)
          RETURN
3125    CONTINUE
          NSUBI=2
          W2(1)=W1(2)
          W2(2)=W1(4)
          RETURN

C
C   Shifted Minimum Surface Problem
C   ------------------------------
C
```

```
1013    CONTINUE
        DNS=NS+.1
        MO=DSQRT(DNS)
        GO TO (131,132,133,134)MODE
C
C   1) Uw1=w2
C
131     CONTINUE
        IF(IELF.LE.MO)GO TO 130
        IF(IELF.GT.(NS-MO))GO TO 135
        IBOUN=MOD(IELF,MO)
        IF(IBOUN.EQ.0)GO TO 136
        IF(IBOUN.EQ.1)GO TO 137
          NSUBI=3
          W2(1)=W1(1)
          W2(2)=W1(2)-W1(3)
          W2(3)=W1(4)
          RETURN
130     CONTINUE
          IF(IELF.EQ.1)GO TO 138
          IF(IELF.EQ.MO)GO TO 139
            NSUBI=2
            W2(1)=W1(3)
            W2(2)=W1(4)
            RETURN
138       CONTINUE
            NSUBI=1
            W2(1)=W1(4)
            RETURN
139       CONTINUE
            NSUBI=1
            W2(1)=W1(3)
            RETURN
135     CONTINUE
          IF(IELF.EQ.NS)GO TO 141
          IF(IELF.EQ.(NS-MO+1))GO TO 140
            NSUBI=2
            W2(1)=W1(1)
            W2(2)=W1(2)
            RETURN
140       CONTINUE
            NSUBI=1
            W2(1)=W1(2)
            RETURN
141       CONTINUE
            NSUBI=1
            W2(1)=W1(1)
            RETURN
137     CONTINUE
          NSUBI=2
          W2(1)=W1(2)
          W2(2)=W1(4)
          RETURN
136     CONTINUE
          NSUBI=2
          W2(1)=W1(1)
          W2(2)=W1(3)
          RETURN
C
C   2) U'w1=w2
```

```
C
133     CONTINUE
        IF(IELF.LE.MO)GO TO 142
        IF(IELF.GT.(NS-MO))GO TO 143
        IBOUN=MOD(IELF,MO)
        IF(IBOUN.EQ.0)GO TO 144
        IF(IBOUN.EQ.1)GO TO 145
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=-W1(2)
          W2(4)=W1(3)
          RETURN
142     CONTINUE
          IF(IELF.EQ.1)GO TO 146
          IF(IELF.EQ.MO)GO TO 147
            W2(1)=0.0
            W2(2)=0.0
            W2(3)=W1(1)
            W2(4)=W1(2)
            RETURN
146       CONTINUE
            W2(1)=0.0
            W2(2)=0.0
            W2(3)=0.0
            W2(4)=W1(1)
            RETURN
147       CONTINUE
            W2(1)=0.0
            W2(2)=0.0
            W2(3)=W1(1)
            W2(4)=0.0
            RETURN
143     CONTINUE
          IF(IELF.EQ.(NS-MO+1))GO TO 148
          IF(IELF.EQ.NS)GO TO 149
            W2(1)=W1(1)
            W2(2)=W1(2)
            W2(3)=0.0
            W2(4)=0.0
            RETURN
148       CONTINUE
            W2(1)=0.0
            W2(2)=W1(1)
            W2(3)=0.0
            W2(4)=0.0
            RETURN
149       CONTINUE
            W2(1)=W1(1)
            W2(2)=0.0
            W2(3)=0.0
            W2(4)=0.0
            RETURN
144     CONTINUE
          W2(1)=W1(1)
          W2(2)=0.0
          W2(3)=W1(2)
          W2(4)=0.0
          RETURN
145     CONTINUE
          W2(1)=0.0
```

```
      W2(2)=W1(1)
      W2(3)=0.0
      W2(4)=W1(2)
      RETURN
C
C   3) Uw2=w1
C
132   CONTINUE
      IF(IELF.LE.M0)GO TO 2142
      IF(IELF.GT.(NS-M0))GO TO 2143
      IBOUN=MOD(IELF,M0)
      IF(IBOUN.EQ.0)GO TO 2144
      IF(IBOUN.EQ.1)GO TO 2145
        W2(1)=W1(1)
        W2(2)=0.5*W1(2)
        W2(3)=-W2(2)
        W2(4)=W1(3)
        RETURN
2142  CONTINUE
        IF(IELF.EQ.1)GO TO 2146
        IF(IELF.EQ.M0)GO TO 2147
          W2(1)=0.0
          W2(2)=0.0
          W2(3)=W1(1)
          W2(4)=W1(2)
          RETURN
2146    CONTINUE
          W2(1)=0.0
          W2(2)=0.0
          W2(3)=0.0
          W2(4)=W1(1)
          RETURN
2147    CONTINUE
          W2(1)=0.0
          W2(2)=0.0
          W2(3)=W1(1)
          W2(4)=0.0
          RETURN
2143  CONTINUE
        IF(IELF.EQ.(NS-M0+1))GO TO 2148
        IF(IELF.EQ.NS)GO TO 2149
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=0.0
          W2(4)=0.0
          RETURN
2148    CONTINUE
          W2(1)=0.0
          W2(2)=W1(1)
          W2(3)=0.0
          W2(4)=0.0
          RETURN
2149    CONTINUE
          W2(1)=W1(1)
          W2(2)=0.0
          W2(3)=0.0
          W2(4)=0.0
          RETURN
2144  CONTINUE
        W2(1)=W1(1)
```

```
            W2(2)=0.0
            W2(3)=W1(2)
            W2(4)=0.0
            RETURN
2145      CONTINUE
            W2(1)=0.0
            W2(2)=W1(1)
            W2(3)=0.0
            W2(4)=W1(2)
            RETURN
C
C    4) U'w2=w1
C
134       CONTINUE
          IF(IELF.LE.M0)GO TO 3142
          IF(IELF.GT.(NS-M0))GO TO 3143
          IBOUN=MOD(IELF,M0)
          IF(IBOUN.EQ.0)GO TO 3144
          IF(IBOUN.EQ.1)GO TO 3145
            NSUBI=3
            W2(1)=W1(1)
            W2(2)=W1(2)
            W2(3)=W1(4)
            RETURN
3142      CONTINUE
            IF(IELF.EQ.1)GO TO 3146
            IF(IELF.EQ.M0)GO TO 3147
              NSUBI=2
              W2(1)=W1(3)
              W2(2)=W1(4)
              RETURN
3146        CONTINUE
              NSUBI=1
              W2(1)=W1(4)
              RETURN
3147        CONTINUE
              NSUBI=1
              W2(1)=W1(3)
              RETURN
3143      CONTINUE
            IF(IELF.EQ.(NS-M0+1))GO TO 3148
            IF(IELF.EQ.NS)GO TO 3149
              NSUBI=2
              W2(1)=W1(1)
              W2(2)=W1(2)
              RETURN
3148        CONTINUE
              NSUBI=1
              W2(1)=W1(2)
              RETURN
3149        CONTINUE
              NSUBI=1
              W2(1)=W1(1)
              RETURN
3144      CONTINUE
            NSUBI=2
            W2(1)=W1(1)
            W2(2)=W1(3)
            RETURN
3145      CONTINUE
```

```
          NSUBI=2
          W2(1)=W1(2)
          W2(2)=W1(4)
          RETURN
C
C   Discrete Boundary Value Problem
C   -------------------------------
C
1016    CONTINUE
        GO TO (161,162,163,164)MODE
C
C   1) Uw1=w2
C
161     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 165
          NSUBI=2
          W2(1)=W1(1)+W1(3)
          W2(2)=W1(2)
          RETURN
165     CONTINUE
          NSUBI=3
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   2) U'w1=w2
C
163     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 166
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(1)
          RETURN
166     CONTINUE
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   3) Uw2=w1
C
162     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 167
          W2(1)=0.5*W1(1)
          W2(2)=W1(2)
          W2(3)=W2(1)
          RETURN
167     CONTINUE
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   4) U'w2=w1
C
164     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 168
          NSUBI=2
          W2(1)=W1(1)
```

```
        W2(2)=W1(2)
        RETURN
168    CONTINUE
        NSUBI=3
        W2(1)=W1(1)
        W2(2)=W1(2)
        W2(3)=W1(3)
        RETURN
C
C   Broyden Tridiagonal Nonlinear System
C   ------------------------------------
C
1017    CONTINUE
        GO TO (171,172,173,174)MODE
C
C   1) Uw1=w2C
C
171     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 170
          NSUBI=2
          W2(1)=W1(1)+W1(3)+W1(3)
          W2(2)=W1(2)
          RETURN
170     CONTINUE
          NSUBI=3
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   2) U'w1=w2
C
173     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 175
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W2(1)+W2(1)
          RETURN
175     CONTINUE
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   3) Uw2=w1
C
172     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 176
          W2(1)=0.2*W1(1)
          W2(2)=W1(2)
          W2(3)=W2(1)+W2(1)
          RETURN
176     CONTINUE
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   4) U'w2=w1
C
```

```
174     CONTINUE
        IF(IELF.EQ.1.OR.IELF.EQ.NS)GO TO 177
          NSUBI=2
          W2(1)=W1(1)
          W2(2)=W1(2)
          RETURN
177     CONTINUE
          NSUBI=3
          W2(1)=W1(1)
          W2(2)=W1(2)
          W2(3)=W1(3)
          RETURN
C
C   Gaussian-like Problem
C   ---------------------
C
1021    CONTINUE
        GO TO (211,212,213,214)MODE
C
C   1) Uw1=w2
C
211     CONTINUE
          NSUBI=2
          W2(1)=W1(1)-W1(2)
          W2(2)=W1(3)
          RETURN
C
C   2) U'w1=w2
C
213     CONTINUE
          W2(1)=W1(1)
          W2(2)=-W1(1)
          W2(3)=W1(2)
        RETURN
C
C   3) Uw2=w1
C
212     CONTINUE
          W2(1)=0.5*W1(1)
          W2(2)=-W2(1)
          W2(3)=W1(2)
          RETURN
C
C   4) U'w2=w1
C
214     CONTINUE
          NSUBI=2
          W2(1)=W1(1)
          W2(2)=W1(3)
          RETURN
C
C   Probability penalty function
C   ----------------------------
C
1047    CONTINUE
        IF(IELF.LT.NS)GO TO 1001
        GO TO (471,472,473,474)MODE
C
C   1) Uw1=w2
C
```

```
471     CONTINUE
        NSUBI=1
        W2(1)=0
        DO 475 I=1,NDIMI
          W2(1)=W2(1)+W1(I)
475       CONTINUE
        RETURN
C
C  2) U'w1=w2
C
472     CONTINUE
        DO 476 I=1,NDIMI
          W2(I)=W1(1)
476       CONTINUE
        RETURN
C
C  3) Uw2=w1
C
473     CONTINUE
        TEMP=W1(1)/FLOAT(NS)
        DO 477 I=1,NDIMI
          W2(I)=TEMP
477       CONTINUE
        RETURN
C
C   4) U'w2=w1
C
474     CONTINUE
        NSUBI=1
        W2(1)=W1(1)
        RETURN
C
C   PSPDOC Example
C   --------------
C
1048    CONTINUE
        W2(1)=W1(1)
        GO TO (481,482,483,484)MODE
C
C   1) U*w1=w2
C
481     CONTINUE
        NSUBI=2
        W2(2)=W1(2)-W1(3)
        RETURN
C
C   2) U*w2=w1
C
482     CONTINUE
        W2(2)=0.5*W1(2)
        W2(3)=-W2(2)
        RETURN
C
C   3) U'*w1=w2
C
483     CONTINUE
        W2(2)=W1(2)
        W2(3)=-W1(2)
        RETURN
C
```

```
C    4) U'*w2=w1
C
484     CONTINUE
        NSUBI=2
        W2(2)=W1(2)
        RETURN
        END
C
C
C
C
C
        FUNCTION XLOWER(IVAR)
C
C
C       ***************************************
C       *                                     *
C       *   Lower bounds on the variables     *
C       *                                     *
C       ***************************************
C
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
        COMMON/TEST/ITEST
C
C
C    Decide which test is condidered
C    -------------------------------
C
        GO TO (1001,1002,1002,1002,1001,1001,1001,1002,1002,1002,
        1      1002,1002,1002,1002,1002,1002,1002,1002,1002,1002,
        1      1002,1002,1023,1023,1023,1023,1027,1028,1029,1002,
        1      1002,1002,1002,1002,1002,1002,1037,1000,1039,1000,
        1      1000,1000,1000,1023,1023,1001,1000,1002,1002,1002
        1      )ITEST
C
C    No lower bound
C    --------------
C
1000    CONTINUE
        XLOWER=-1.0D20
        RETURN
C
C    Positive variable
C    -----------------
C
1001    CONTINUE
        XLOWER=0.0
        RETURN
C
C    Unconstrained problem
C    ---------------------
1002    CONTINUE
        RETURN
C
C    Spiked Linear Minimum Surface
C    -----------------------------
C
1023    CONTINUE
```

```
      XLOWER=2.5
      RETURN
C
C   Extended Wood's Bounded Problem
C   ------------------------------
C
1027  CONTINUE
      XLOWER=-10.0
      RETURN
C
C   Paviani's Bounded Problem
C   ------------------------
C
1028  CONTINUE
      XLOWER=2.001
      RETURN
C
C   Extended McCormick' Bounded Problem
C   ----------------------------------
C
1029  CONTINUE
      XLOWER=-1.5
      RETURN
C
C  Bounded Broyden Tridiagonal Problem
C  -----------------------------------
C
1037  CONTINUE
      XLOWER=0.65
      RETURN
C
C   Bounded ENGLV1 Problem
C   --------------------
C
1039  CONTINUE
      XLOWER=0.5
      RETURN
      END
C
C
C
C
      FUNCTION XUPPER(IVAR)
C
C
C     **************************************
C     *                                    *
C     *   Upper bounds on the variables    *
C     *                                    *
C     **************************************
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      COMMON/TEST/ITEST
C
C
C   Decide which test is condidered
C   ------------------------------
C
```

```
      GO TO (1001,1002,1002,1002,1001,1001,1001,1002,1002,1002,
     1       1002,1002,1002,1002,1002,1002,1002,1002,1002,1002,
     1       1002,1002,1023,1023,1023,1023,1027,1028,1029,1002,
     1       1002,1002,1002,1002,1002,1002,1037,1000,1039,1040,
     1       1041,1027,1043,1023,1023,1001,1048,1002,1002,1002
     1       )ITEST
C
C   Negative variable
C   -----------------
C
1000  CONTINUE
      XUPPER=0.0
      RETURN
C
C   No upper bound
C   --------------
C
1001  CONTINUE
      XUPPER=1.0D+20
      RETURN
C
C   Unconstrained problem
C   ---------------------
1002  CONTINUE
      RETURN
C
C   Spiked Linear Minimum Surface
C   -----------------------------
C
1023  CONTINUE
      XUPPER=4.0
      RETURN
C
C   Bounded Woods Problem
C   ---------------------
C
1027  CONTINUE
      XUPPER=0.5D0
      RETURN
C
C   Paviani's Bounded Problem
C   -------------------------
C
1028  CONTINUE
      XUPPER=9.999
      RETURN
C
C   Extended McCormick' Bounded Problem
C   -----------------------------------
C
1029  CONTINUE
      XUPPER=3.0
      RETURN
C
C   Bounded Broyden Tridiagonal Problem
C   -----------------------------------
C
1037  CONTINUE
      XUPPER=0.71
      RETURN
```

```
C
C   Bounded ENGLV1 Problem
C   --------------------
C
1039    CONTINUE
        XUPPER=0.63
        RETURN
C
C   Bounded Freudenstein and Roth Problem
C   ------------------------------------
C
1040    CONTINUE
        XUPPER=3.0
        RETURN
C
C   Extended Bounded SCHMVT Problem
C   ------------------------------
C
1041    CONTINUE
        IF(MOD(IVAR,2).EQ.0)GO TO 410
        XUPPER=0.75
        RETURN
410     CONTINUE
        XUPPER=10.0
        RETURN
C
C   Bounded TRIDIA Problem
C   --------------------
C
1043    CONTINUE
        IF(IVAR.NE.1)GO TO 430
        XUPPER=0.1
        RETURN
430     CONTINUE
        XUPPER=0.05
        RETURN
C
C   PSPDOC Example
C   -------------
C
1048    CONTINUE
        XUPPER=-1.0
        RETURN
        END
C
C
C
        SUBROUTINE ELFNCT(IELF,X,FX,GX,NDIMI,NS,IFFLAG,
        1                 FMAX,FNOISE)
C
C
C       ***********************************************
C       *                                             *
C       *    Computation of the element function value *
C       *                                             *
C       ***********************************************
C
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
```

```
      DIMENSION X(1),GX(1)
C
      COMMON/TEST/ITEST
C
C
C   Decide which test is condidered
C   -------------------------------
C
      GO TO (1001,1002,1002,1004,1005,1006,1007,1008,1009,1010,
     1      1011,1011,1013,1013,1010,1016,1017,1018,1019,1020,
     1      1021,1022,1011,1011,1011,1011,1027,1028,1029,1027,
     1      1031,1032,1033,1034,1035,1036,1017,1032,1031,1033,
     1      1035,1010,1008,1013,1013,1047,1048,1010,1017,1011
     1      )ITEST
C
C   First trivial test problem
C   --------------------------
C
1001  CONTINUE
      IF(IELF.GT.1)GO TO 1
        FX=X(1)
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=1.0
        RETURN
1     CONTINUE
        FX=0.5*(X(1)-X(2))**2+X(1)**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=X(1)-X(2)+X(1)+X(1)
        GX(2)=X(2)-X(1)
        RETURN
C
C   Second trivial unconstrained quadratic problem
C   ----------------------------------------------
C
1002  CONTINUE
      IF(IELF.GT.1)GO TO 2
        FX=X(1)**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=X(1)+X(1)
        RETURN
2     CONTINUE
        FX=0.5*(X(1)-X(2))**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=X(1)-X(2)
        GX(2)=X(2)-X(1)
        RETURN
C
C   Idem, but without derivatives
C   -----------------------------
C
1004  CONTINUE
      IF(IELF.GT.1)GO TO 3
        FX=X(1)**2+X(2)**2+1.0
        FNOISE=0.0
        RETURN
3     CONTINUE
```

```
          FX=0.5*((X(1)-X(2))**2+(X(3)-X(4))**2)+1.0
          FNOISE=0.0
          RETURN
C
C   Trivial linear problem
C   ----------------------
C
1005    CONTINUE
        IF(IELF.EQ.1)FX=X(1)+2.0*X(2)
        IF(IELF.EQ.2)FX=3.0*X(1)+10.0*X(2)
        FNOISE=0.0
        RETURN
C
C   Mixed trivial test problem
C   --------------------------
1006    CONTINUE
        IF(IELF.EQ.1)FX=X(1)+X(2)*X(2)/2.0
        IF(IELF.EQ.2)FX=0.5*((X(1)-X(3))**2+(X(2)-X(4))**2)
        FNOISE=0.0
        RETURN
C
C   Easy non quadratic test problem
C   -------------------------------
C
1007    CONTINUE
        IF(IELF.EQ.1)FX=X(1)+DSQRT(1.0+X(2)**2)
        IF(IELF.EQ.2)
        1      FX=DSQRT(1.0+0.5*((X(1)-X(3))**2+(X(2)-X(4))**2))
        FNOISE=0.0
        RETURN
C
C   TRIDIA
C   ------
C
1008    CONTINUE
        IF(IELF.EQ.1)GO TO 80
          DSX=IELF*(2.0*X(2)-X(1))
          FX=DSX*DSX/IELF
          FNOISE=0.0
          IF(IFFLAG.LT.2)RETURN
          GX(1)=-2.0*DSX
          GX(2)=4.0*DSX
          RETURN
80      CONTINUE
          DSX=2.0*X(1)-1.0
          FX=DSX*DSX
          FNOISE=0.0
          IF(IFFLAG.LT.2)RETURN
          GX(1)=4.0*DSX
          RETURN
C
C   STRID
C   -----
C
1009    CONTINUE
C
C   1) Convexity parameter (equals TRIDIA for T=0.)
C
        T=-1.0
C
```

```
      IF(IELF.EQ.1)GO TO 90
      DSX=IELF*(2.0*X(2)-X(1))
      FX=DSX*DSX/IELF+T*(IELF-1)*X(1)**2
      IF(IELF.NE.NS)FX=FX-T*IELF*X(2)**2
      FNOISE=0.0
      IF(IFFLAG.LT.2)RETURN
      GX(1)=-2.0*DSX+T*(IELF-1)*2.0*X(1)
      GX(2)=4.0*DSX
      IF(IELF.NE.NS)GX(2)=GX(2)-T*IELF*2.0*X(2)
      RETURN
90    CONTINUE
      DSX=2.0*X(1)-1.0
      FX=DSX*DSX-T*X(1)**2
      FNOISE=0.0
      IF(IFFLAG.LT.2)RETURN
      GX(1)=4.0*DSX-2.0*T*X(1)
      RETURN
C
C  Extended Rosenbrock function
C  ----------------------------
C
1010  CONTINUE
      XX=X(2)-X(1)**2
      FX=100.0*XX*XX+(X(1)-1.0)**2
      FNOISE=0.0
      IF(IFFLAG.LT.2)RETURN
      GX(1)=-400.0*XX*X(1)
      GX(2)=200.0*XX+2.0*(X(2)-1.0)
      RETURN
C
C  Minimum Surface Problem
C  -----------------------
C
1011  CONTINUE
      Z1=X(1)-X(4)
      Z2=X(2)-X(3)
      FX=DSQRT(1.+.5*(Z1*Z1+Z2*Z2)*NS)
      FXFX=FX+FX
      FX=FX/NS
      FNOISE=0.0
      IF(IFFLAG.LT.2)RETURN
      GX(1)=Z1/FXFX
      GX(4)=-GX(1)
      GX(2)=Z2/FXFX
      GX(3)=-GX(2)
      RETURN
C
C  Shifted Minimum Surface Problem
C  -------------------------------
C
1013  CONTINUE
      Z1=X(1)-X(4)
      Z2=X(2)-X(3)
      FX=DSQRT(1.+.5*(Z1*Z1+Z2*Z2)*NS)
      FNOISE=0.0
      IF(IFFLAG.LT.2)GO TO 131
      GX(1)=Z1/(FX+FX)
      GX(4)=-GX(1)
      GX(2)=Z2/(FX+FX)
      GX(3)=-GX(2)
```

```
131     CONTINUE
        FX=FX/NS
C
        T=-1.0/NS
C
        DNS=NS+0.1
        MO=DSQRT(DNS)
        IBOUN=MOD(IELF,MO)
        IF(IELF.LE.MO.OR.IBOUN.EQ.1)GO TO 130
          FX=FX+T*X(1)**2
          FNOISE=0.0
          IF(IFFLAG.LT.2)GO TO 130
          GX(1)=GX(1)+2.0*T*X(1)
130     CONTINUE
        IF(IELF.GT.(NS-MO).OR.IBOUN.EQ.0)RETURN
          FX=FX-T*X(4)**2
          FNOISE=0.0
          IF(IFFLAG.LT.2)RETURN
          GX(4)=GX(4)-2.0*T*X(4)
          RETURN
C
C   Discrete Boundary Value Problem
C   -------------------------------
C
1016    CONTINUE
        H=1.0/(N-1.0)
        HH=H*H
        TEMP1=X(2)+I*H+1.0
        TEMP=X(2)+X(2)-X(1)-X(3)+0.5*HH*TEMP1**3
        FX=TEMP**2
        TEMP=TEMP+TEMP
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=-TEMP
        GX(2)=TEMP*(2.0+1.5*HH*TEMP1*TEMP1)
        GX(3)=-TEMP
        RETURN
C
C   Broyden Tridiagonal Nonlinear System
C   ------------------------------------
C
1017    CONTINUE
        TEMP=(3.0-X(2)-X(2))*X(2)-X(1)-X(3)-X(3)+1.0
        FX=TEMP*TEMP
        TEMP=TEMP+TEMP
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=-TEMP
        GX(2)=TEMP*(3.0-4.0*X(2))
        GX(3)=-TEMP-TEMP
        RETURN
C
C   Broyden Banded Function
C   -----------------------
C
1018    CONTINUE
        ILOW=MAX0(1,IELF-5)
        IUP=MIN0(NS,IELF+1)
        FX=1.0
        DO 181 J=ILOW,IUP
```

```
          JJ=J-ILOW+1
          IF(J.EQ.IELF)GO TO 182
            FX=FX-X(JJ)*(1.0+X(JJ))
            GO TO 181
182       CONTINUE
            FX=FX+X(JJ)*(2.0+5.0*X(JJ)*X(JJ))
181       CONTINUE
        TEMP=FX+FX
        FX=FX*FX
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        DO 180 J=ILOW,IUP
          JJ=J-ILOW+1
          GX(JJ)=0.0
180       CONTINUE
        DO 183 J=ILOW,IUP
          JJ=J-ILOW+1
          IF(J.EQ.IELF)GX(JJ)=TEMP*(2.0+15.0*X(JJ)*X(JJ))
          IF(J.NE.IELF)GX(JJ)=-TEMP*(1.0+2.0*X(JJ))
183       CONTINUE
        RETURN
C
C   Extended Powell Singular Function
C   ---------------------------------
C
1019    CONTINUE
        TEMP1=X(1)+10.0*X(2)
        TEMP2=X(3)-X(4)
        TEMP3=X(2)-2.0*X(3)
        TEMP4=X(1)-X(4)
        FX=TEMP1**2+5.0*TEMP2**2+TEMP3**4+10.0*TEMP4**4
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=2.0*TEMP1+40.0*TEMP4**3
        GX(2)=20.0*TEMP1+4.0*TEMP3**3
        GX(3)=10.0*TEMP2-8.0*TEMP3**3
        GX(4)=-10.0*TEMP2-40.0*TEMP4**3
        RETURN
C
C   Extended Wood Function
C   ----------------------
C
1020    CONTINUE
        TEMP1=X(2)-X(1)*X(1)
        TEMP2=1.0-X(1)
        TEMP3=X(4)-X(3)*X(3)
        TEMP4=1.0-X(3)
        TEMP5=X(2)+X(4)-2.0
        TEMP6=X(2)-X(4)
        FX=100.0*TEMP1**2+TEMP2**2+90.0*TEMP3**2
        FX=FX+TEMP4**2+10.0*TEMP5**2+10.0*TEMP6**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=-400.0*TEMP1*X(1)-2.0*TEMP2
        GX(2)=200.0*TEMP1+20.0*TEMP5+20.0*TEMP6
        GX(3)=-360.0*TEMP3*X(3)-2.0*TEMP4
        GX(4)=180.0*TEMP3+20.0*TEMP5-20.0*TEMP6
        RETURN
C
C   Gaussian-like Problem
```

```
C    --------------------
C
1021     CONTINUE
         ALPHA=10.0/NS
         BETA=0.1
         XY=X(1)-X(2)
         TEMP0=BETA+X(3)**2
         TEMP4=-XY*XY/TEMP0
         TEMP1=DEXP(TEMP4)
         TEMP2=ALPHA+X(3)**2
         FX=TEMP2*(2.0-TEMP1)
         FNOISE=0.0
         IF(IFFLAG.LT.2)RETURN
         TEMP5=TEMP2*TEMP1
         GX(1)=2.0*TEMP5*XY
         GX(2)=-GX(1)
         GX(3)=2.0*X(3)*(2.0-TEMP1+2.0*TEMP5*TEMP4/TEMP0)
         RETURN
C
C    Diagonal Quadratic
C    -----------------
C
1022     CONTINUE
         ALPHA=100.0
         BETA=100.0
         FX=ALPHA*X(2)**2+X(1)**2+BETA*X(3)**2
         FNOISE=0.0
         IF(IFFLAG.LT.2)RETURN
         GX(1)=2.0*X(1)
         GX(2)=2.0*ALPHA*X(2)
         GX(3)=2.0*ALPHA*X(3)
         RETURN
C
C    Extended Wood's Bounded Problem
C    ------------------------------
C
1027     CONTINUE
         TEMP1=X(2)-X(1)**2
         TEMP2=1.-X(1)
         TEMP3=X(4)-X(3)**2
         TEMP4=1.-X(3)
         TEMP5=X(2)-1.0
         TEMP6=X(4)-1.0
         FX=100.0*TEMP1**2+TEMP2**2+90.0*TEMP3**2+TEMP4**2
         FX=FX+10.1*(TEMP5**2+TEMP6**2)+19.8*TEMP5*TEMP6
         FNOISE=0.0
         IF(IFFLAG.LT.2)RETURN
         GX(1)=-400.0*TEMP1*X(1)-2.0*TEMP2
         GX(2)=200.0*TEMP1+20.2*TEMP5+19.8*TEMP6
         GX(3)=-360.0*TEMP3*X(3)-2.0*TEMP4
         GX(4)=180.0*TEMP3+20.2*TEMP6+19.8*TEMP5
         RETURN
C
C    Paviani's Bounded Problem
C    ------------------------
C
1028     CONTINUE
         TEMP=1.0
         FX=0.0
         DO 280 I=1,10
```

```
          TEMP=TEMP*X(I)
          FX=FX+(DLOG(X(I)-2.0))**2+(DLOG(10.0-X(I)))**2
280       CONTINUE
        TEMP=TEMP**0.2
        FX=FX-TEMP
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        DO 281 I=1,10
          TEMP1=X(I)-2.0
          TEMP2=10.0-X(I)
          GX(I)=DLOG(TEMP1)/TEMP1-DLOG(TEMP2)/TEMP2
          GX(I)=2.0*GX(I)-0.2*TEMP/X(I)
281       CONTINUE
        RETURN
C
C   Extended McCormick' Bounded Problem
C   ----------------------------------
C
1029    CONTINUE
        FX=DSIN(X(1)+X(2))+(X(1)-X(2))**2-1.5*X(1)+2.5*X(2)+1.0
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=DCOS(X(1)+X(2))+2.0*(X(1)-X(2))-1.5
        GX(2)=DCOS(X(1)+X(2))-2.0*(X(1)-X(2))+2.5
        RETURN
C
C  Extended ENGVL1 Problem
C  -------------------
C
1031    CONTINUE
        TEMP1=X(1)**2+X(2)**2
        FX=TEMP1**2-4.0*X(1)+3.0
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=4.0*(X(1)*TEMP1-1.0)
        GX(2)=4.0*X(2)*TEMP1
        RETURN
C
C   Extended CRGLVY Problem
C   ---------------------
C
1032    CONTINUE
        TEMP0=DEXP(X(1))
        TEMP1=TEMP0-X(2)
        TEMP2=X(2)-X(3)
        TEMP3=DSIN(X(3)-X(4))
        TEMP4=DCOS(X(3)-X(4))
        TEMP5=TEMP3/TEMP4
        TEMP6=X(4)-1.0
        FX=TEMP1**4+100.0*TEMP2**6+TEMP5**4+X(1)**8+TEMP6**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        TEMP1=TEMP1**3
        TEMP2=TEMP2**5
        TEMP3=TEMP3**3/TEMP4**5
        GX(1)=4.0*TEMP0*TEMP1+8.0*X(1)**7
        GX(2)=-4.0*TEMP1+600.0*TEMP2
        GX(3)=-600.0*TEMP2+4.0*TEMP3
        GX(4)=-4.0*TEMP3+2.0*TEMP6
        RETURN
```

```
C
C   Extended Freudenstein and Roth Problem
C   --------------------------------------
C
1033    CONTINUE
        TEMP1=X(1)+X(2)*((5.0-X(2))*X(2)-2.0)-13.0
        TEMP2=X(1)+X(2)*((X(2)+1.0)*X(2)-14.0)-29.0
        FX=TEMP1**2+TEMP2**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=2.0*(TEMP1+TEMP2)
        GX(2)=2.0*(TEMP1*(-3.0*X(2)**2+10.*X(2)-2.)
        1       +TEMP2*(3.*X(2)**2+2.*X(2)-14))
        RETURN
C
C   Extended Powell Badly Scaled problem
C   ------------------------------------
C
1034    CONTINUE
        TEMP1=DEXP(-X(1))
        TEMP2=DEXP(-X(2))
        TEMP3=10000.0*X(1)*X(2)-1.0
        TEMP4=TEMP1+TEMP2-1.0001
        FX=TEMP3**2+TEMP4**2
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        TEMP3=20000.0*TEMP3
        TEMP4=2.0*TEMP4
        GX(1)=X(2)*TEMP3-TEMP1*TEMP4
        GX(2)=X(1)*TEMP3-TEMP2*TEMP4
        RETURN
C
C   Extended SCHMVT Problem
C   -----------------------
C
1035    CONTINUE
        TEMP1=X(1)-X(2)
        TEMP2=X(1)+X(3)
        PI=3.1415926535D0
        TEMP3=1.0+TEMP1**2
        TEMP4=(PI*X(2)+X(3))/2.0
        TEMP5=((X(1)+X(3))/X(2))-2.0
        TEMP6=DEXP(-TEMP5**2)
        FX=3.0-1.0/TEMP3-DSIN(TEMP4)-TEMP6
        FNOISE=0.0
        IF(IFFLAG.LT.2)RETURN
        TEMP3=2.0*TEMP1/TEMP3**2
        TEMP4=DCOS(TEMP4)
        TEMP6=2.0*TEMP5*TEMP6/X(2)
        GX(1)=TEMP3+TEMP6
        GX(2)=-TEMP3-0.5*PI*TEMP4
        1              -TEMP2*TEMP6/X(2)
        GX(3)=-0.5*TEMP4+TEMP6
        RETURN
C
C   Extended Cube Problem
C   ---------------------
C
1036    CONTINUE
        TEMP1=X(2)-X(1)**3
```

```
      TEMP2=1.0-X(1)
      FX=100.0*TEMP1**2+TEMP2**2
      FNOISE=0.0
      IF(IFFLAG.LT.2)RETURN
      GX(1)=-600.0*TEMP1*X(1)**2-2.0*TEMP2
      GX(2)=200.0*TEMP1
      RETURN
C
C   Probability Penalty Function Problem
C   ------------------------------------
C
1047  CONTINUE
      IF(IELF.EQ.NS)GO TO 470
        TEMP0=DEXP(-X(1)*X(2))
        TEMP1=X(1)+X(2)
        FX=TEMP1*TEMP0
        IF(IFFLAG.LT.2)RETURN
        GX(1)=TEMP0*(1.-X(2)*TEMP1)
        GX(2)=TEMP0*(1.-X(1)*TEMP1)
        RETURN
470   CONTINUE
        A=100.
        TEMP0=-1.0
        DO 471 I=1,NDIMI
          TEMP0=TEMP0+X(I)
471       CONTINUE
        FX=A*TEMP0**2
        IF(IFFLAG.LT.2)RETURN
        TEMP0=A*TEMP0
        TEMP0=TEMP0+TEMP0
        DO 472 I=1,NDIMI
          GX(I)=TEMP0
472       CONTINUE
        RETURN
C
C   PSPDOC Example
C   --------------
C
1048  CONTINUE
      TEMP=X(2)-X(3)
      FX=DSQRT(1.0+X(1)**2+TEMP**2)
      FNOISE=0.0
      IF(IFFLAG.LT.2)RETURN
      GX(1)=X(1)/FX
      GX(2)=TEMP/FX
      GX(3)=-GX(2)
      RETURN
      END
C
C
```

REFERENCES

[1] A.G. Buckley.
Private communication.

[2] A. Griewank and Ph.L. Toint.
Numerical Experiments With Partially Separable Optimization Problems.
Technical Report 83/2, dept of Maths, FUN Namur (Belgium), 1983.

[3] D.M. Himmelblau.
Applied Nonlinear Programming.
McGraw-Hill, New-York, 1972.

[4] W. Hock and K. Schittkowski.
Test Examples for Nonlinear Programming Codes.
Lectures Notes in Economics and Mathematical Systems 187, Springer Verlag, Berlin, 1981.

[5] J.J. More, B.S. Garbow and K.E. Hillstrom.
Testing Unconstrained Optimization Software.
ACM Transactions on Mathematical Software 7(1):17-41, 1981.