GRG
Generalized Reduced Gradient Algorithm

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

author

Consider the nonlinear programming problem

$$
\begin{aligned}
&\textbf{Minimize } f(x_1, x_2, \cdots x_n)\\
&\textbf{subject to}\\
&\qquad h_i(x_1, x_2, \cdots x_n) = 0,\ i = 1, 2, \cdots m\\
&\qquad\qquad a_j \le x_j \le b_j,\ j=1, 2, \cdots n
\end{aligned}
$$

The GRG (Generalized Reduced Gradient) algorithm is similar in concept to the Simplex method for LP:

SIMPLEX		GRG
reduced costs	⟷	reduced gradient
basic variables	⟷	dependent variables (state variables)
nonbasic variables	⟷	independent variables (decision variables)

There are, however, several differences between the two algorithms:

In GRG, *unlike the simplex method,*

- nonbasic (*independent*) variables need not be at their bound (lower or upper)
- at each iteration, several nonbasic (*independent*) variables may have their values changed (increased or decreased)
- the basis need not change at each iteration

At the beginning of each iteration, the n variables are partitioned into two sets:

- Dependent variables (one per equation)
- Independent variables

(after re-ordering the variables):

$$x = \begin{bmatrix} x_D \\ x_I \end{bmatrix} \text{ where } \begin{cases} x_D = \text{vector of } m \text{ dependent variables} \\ x_I = \text{vector of } (n-m) \text{ independent variables} \end{cases}$$

In the same manner, we partition the gradient of the objective and the bounds:

$$a = \begin{bmatrix} a_D \\ \hdashline a_I \end{bmatrix},\ b = \begin{bmatrix} b_D \\ \hdashline b_I \end{bmatrix},\ \nabla f(x) = \begin{bmatrix} \nabla_D f(x) \\ \hdashline \nabla_I f(x) \end{bmatrix}$$

and the Jacobian matrix:

$$J(x) = \left[\, J_D(x) \,\vdots\, J_I(x) \,\right] = \begin{bmatrix} \nabla_D h_1(x) & \nabla_I h_1(x) \\ \nabla_D h_2(x) & \nabla_I h_2(x) \\ \vdots & \vdots \\ \nabla_D h_m(x) & \nabla_I h_m(x) \end{bmatrix}$$

Suppose that we are given an initial point X^0 which satisfies:

1) $\mathbf{h}_i(X^0) = 0 \;\forall\; \mathbf{i}$

2) $\mathbf{a}_D < X^0_D < \mathbf{b}_D$ *(nondegeneracy)*

3) $J_D(X^0)$ is nonsingular, i.e., $\left[J_D(X^0)\right]^{-1}$ exists

4) $\mathbf{a}_I \leq X^0_I \leq \mathbf{b}_I$

Denote the change in X by $\delta = \begin{bmatrix} \delta_D \\ \delta_I \end{bmatrix}$

For "small" δ, the change in the objective is

$$\Delta \mathbf{f} = \left(\mathbf{f}(X^0 + \delta) - \mathbf{f}(X^0)\right) \approx \left[\nabla \mathbf{f}(X^0)\right]^T \bullet \delta$$

i.e.,

$$\Delta \mathbf{f} \approx \begin{bmatrix} \nabla_D \mathbf{f}(X^0) \\ \nabla_I \mathbf{f}(X^0) \end{bmatrix}^T \bullet \begin{bmatrix} \delta_D \\ \delta_I \end{bmatrix} = \nabla_D \mathbf{f}(X^0) \bullet \delta_D + \nabla_I \mathbf{f}(X^0) \bullet \delta_I$$

We want to choose δ so that we maintain feasibility:

$$\left(\mathbf{h}_i(X^0 + \delta) - \mathbf{h}_i(X^0)\right) = \Delta \mathbf{h}_i \approx \left[\nabla \mathbf{h}_i(X^0)\right]^T \bullet \delta = 0 \;\forall\; \mathbf{i}$$

i.e., $$\Delta \mathbf{h}_i \approx \nabla_D \mathbf{h}_i(X^0) \bullet \delta_D + \nabla_I \mathbf{h}_i(X^0) \bullet \delta_I = 0 \;\forall\; \mathbf{i}$$

This system of equations (linear in δ) may be written:

$$\Delta \mathbf{h} = J(X^0) \bullet \delta = J_D(X^0) \bullet \delta_D + J_I(X^0) \bullet \delta_I = 0$$

Since we assume that $J_D(X^0)$ is nonsingular,

$$J_D(X^0) \bullet \delta_D + J_I(X^0) \bullet \delta_I = 0$$

$$\Rightarrow \boxed{\delta_D = -\left[J_D(X^0)\right]^{-1} J_I(X^0) \bullet \delta_I}$$

This equation tells us the required changes in the *dependent* variables which are required to maintain feasibility when the *independent* variables are changed by the amount δ_I

We now make the substitution

$$\boxed{\delta_D = -\left[J_D(X^0)\right]^{-1} J_I(X^0) \bullet \delta_I}$$

into the estimate of change in the objective function:

$$\Delta f \approx \nabla_D f(X^0) \bullet \delta_D + \nabla_I f(X^0) \bullet \delta_I$$

$$\Delta f \approx \nabla_D f(X^0) \left[-\left[J_D(X^0)\right]^{-1} J_I(X^0)\ \delta_I\right] + \nabla_I f(X^0)\ \delta_I$$

$$\Delta f \approx \left[\nabla_I f(X^0) - \nabla_D f(X^0) \left[J_D(X^0)\right]^{-1} J_I(X^0)\right] \delta_I \equiv \Gamma_I\ \delta_I$$

That is,

$$\Delta f \approx \Gamma_I\ \delta_I$$

where the "reduced gradient" Γ_I is defined as

$$\boxed{\Gamma_I \equiv \nabla_I f(X^0) - \nabla_D f(X^0) \left[J_D(X^0)\right]^{-1} J_I(X^0)}$$

This gives us an estimate of the change in the objective when we change the independent variables X_I by the amount δ_I <u>and</u> change the dependent variables X_D by the amount required to maintain feasibility!

$$\boxed{\Gamma_I \equiv \nabla_I f(X^0) - \nabla_D f(X^0)\left[J_D(X^0)\right]^{-1} J_I(X^0)}$$

Compare the "reduced gradient" in GRG to the "reduced cost" in the Simplex method for LP:

$$\boxed{\bar{c}_j = c_j - z_j = c_j - \pi\, A^j = c_j - c_B\left[A^B\right]^{-1} A^j}$$

$$\left.\begin{array}{l}\textit{simplex}\\ \textit{multiplier}\\ \textit{vector}\end{array}\right\}\ \pi = c_B\left[A^B\right]^{-1}$$

Since the objective is to be minimized, we choose to move each independent variable in the *negative* of the direction given by the reduced gradient, taking into acount the upper & lower bounds on X_I:

$$\text{for each } i \in I, \quad \delta_i = \begin{cases} 0 \ \text{ if } \Gamma_i > 0 \text{ and } x_i^0 = a_i \\ 0 \ \text{ if } \Gamma_i < 0 \text{ and } x_i^0 = b_i \\ -\Gamma_i \ \text{ otherwise} \end{cases}$$

Once the step *direction* δ_I (for the independent variables) is chosen, then the step direction for the dependent variables is determined by

$$\delta_D = -\left[J_D(X^0)\right]^{-1} J_I(X^0) \bullet \delta_I$$

(By the nondegeneracy assumption, i.e.,

$$\mathbf{a}_D < X_D^0 < \mathbf{b}_D$$

some positive step can always be made in the dependent variables.)

$$\delta_i = \begin{cases} 0 & \text{if } \Gamma_i > 0 \text{ and } x_i^0 = a_i \\ 0 & \text{if } \Gamma_i < 0 \text{ and } x_i^0 = b_i \\ -\Gamma_i & \text{otherwise} \end{cases} \qquad \forall i \in I$$

$$\delta_D = -\left[J_D(X^0)\right]^{-1} J_I(X^0) \bullet \delta_I$$

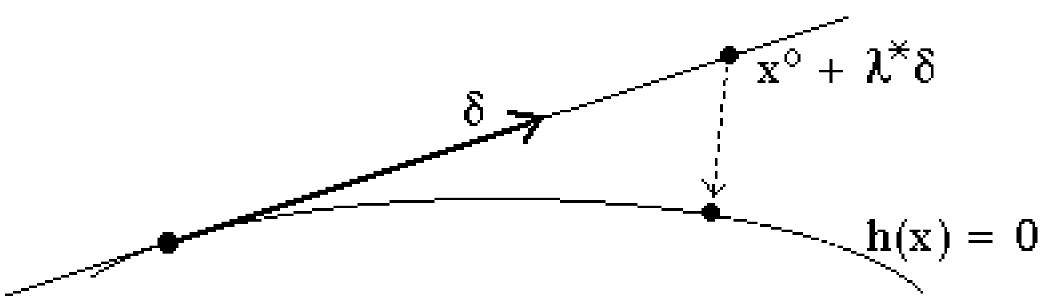
Note that, unlike the Simplex LP method, which chooses a single nonbasic ($\approx$ independent) variable to be changed, GRG simultaneously changes many of the independent variables!

Having found the direction δ in which to move, we next do a one-dimensional search along this direction in order to

$$\begin{array}{l} \underset{\lambda}{\text{Minimize}}\ f(x^\circ + \lambda\,\delta) \\ \text{subject to} \\ \qquad a \le x^\circ + \lambda\,\delta \le b \\ \text{i.e.,} \\ \qquad a - x^\circ \le \lambda\,\delta \le b - x^\circ \end{array}$$

This can be done by any of several one-dimensional search methods, e.g., golden section search, cubic interpolation, etc.

In general, when the constraints are nonlinear, for the optimal stepsize λ^*, $h(x^\circ + \lambda^*\delta) \neq 0$



Then we need to move back onto the feasible surface by solving $h(x)=0$, using $x^\circ + \lambda^*\delta$ as an initial "guess" (e.g., using the Newton-Raphson method).

EXAMPLE

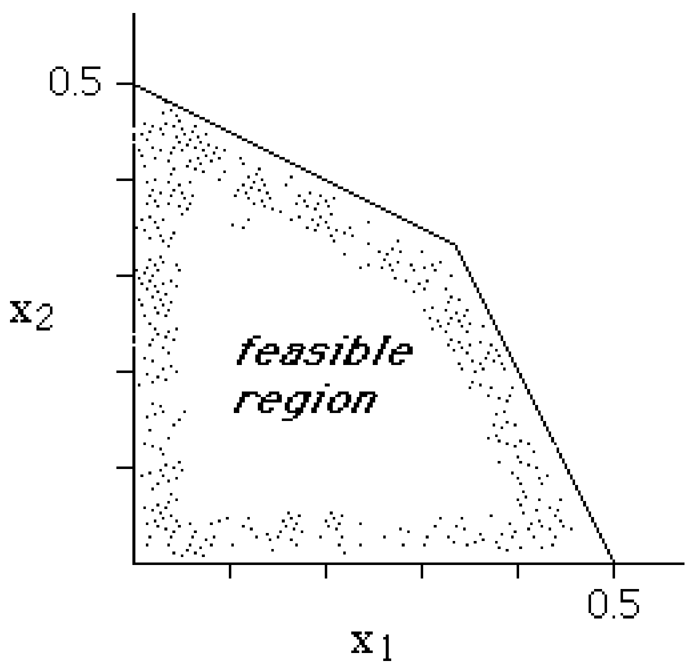
Minimize $f(x) = x_1^2 - x_1 - x_2$

subject to $\begin{cases} g_1(x) = 2x_1 + x_2 \le 1 \\ g_2(x) = x_1 + 2x_2 \le 1 \end{cases}$

$x_j \ge 0,\ j=1,2$

We first write the inequality constraints as equations:

$$\begin{cases} h_1(x) = 2x_1 + x_2 + x_3 - 1 = 0 \\ h_2(x) = x_1 + 2x_2 + x_4 - 1 = 0 \end{cases}$$



For standard GRG form, we need both upper & lower bounds on the variables, which we deduce:

$$2x_1 + x_2 \le 1 \quad \Longrightarrow \quad x_2 \le 1$$

$$x_1 + 2x_2 \le 1 \quad \Longrightarrow \quad x_1 \le 1$$

$$\left.\begin{array}{r} x_3 = 1 - (2x_1 + x_2) \\ 2x_1 + x_2 \ge 0 \end{array}\right\} \Longrightarrow \quad x_3 \le 1$$

$$\left.\begin{array}{r} x_4 = 1 - (x_1 + 2x_2) \\ x_1 + 2x_2 \ge 0 \end{array}\right\} \Longrightarrow \quad x_4 \le 1$$

Standard Form

Minimize $f(x) = x_1^2 - x_1 - x_2$

subject to

$$h_1(x) = 2x_1 + x_2 + x_3 \qquad - 1 = 0$$

$$h_2(x) = x_1 + 2x_2 \qquad + x_4 - 1 = 0$$

$$0 \le x_j \le 1 \ , \ j=1,2,3,4$$

We will use as feasible starting points

$$X^0 = \left(\tfrac{1}{4}, 0, \tfrac{1}{2}, \tfrac{3}{4}\right)$$

$$X^0 = \left(\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}\right)$$

$$X^0 = \left(\tfrac{1}{4}, 0, \tfrac{1}{2}, \tfrac{3}{4}\right)$$

at lower bound

To avoid degeneracy in the initial partition, we cannot allow x_2 to be dependent ("basic"), and so our choice of two dependent variables is limited to x_1, x_3, and x_4.

For the starting partition of the variables, let's define (arbitrarily)

$$x_I = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; x_D = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}, \quad D = \{3,4\} \quad \text{and} \quad I = \{1,2\}$$

$$\nabla f(x) = \begin{bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \hdashline \partial f / \partial x_3 \\ \partial f / \partial x_4 \end{bmatrix} = \begin{bmatrix} 2x_1 - 1 \\ -1 \\ \hdashline 0 \\ 0 \end{bmatrix}$$

$$\nabla_I f(x^\circ) = \begin{bmatrix} -\tfrac{1}{2} \\ -1 \end{bmatrix}$$

$$\nabla_D f(x^\circ) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$J_D(x) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; J_I(x) = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\Gamma_I = \nabla_I f(x^\circ) - \underbrace{\nabla_D f(x^\circ)}_{zero} \left[J_D\right]^{-1} J_I$$

reduced gradient

$$\Rightarrow \quad \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -1 \end{bmatrix} \quad \& \quad X^0 = \left(\tfrac{1}{4}, 0, \tfrac{1}{2}, \tfrac{3}{4}\right)$$

Computing the step direction:

$$\left.\begin{matrix} 0 < x_1^0 < 1 \\ 0 = x_2^0 < 1 \end{matrix}\right\} \Rightarrow \begin{matrix} \delta_1 = -\Gamma_1 = \frac{1}{2} \\ \delta_2 = -\Gamma_2 = 1 \end{matrix}$$

(Neither independent variable is at its upper bound, and so $\delta_I = -\Gamma_I$)

$$\delta_I = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}$$

$$\delta_D = -\left[J_D(X^0)\right]^{-1} J_I(X^0) \bullet \delta_I$$

$$\Rightarrow \quad \delta_D = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -5/2 \end{bmatrix}$$

objective function

```
Z←F X;Q;C
⍝
⍝       Quadratic objective function for GRG Example 2
⍝
X←2↑X  ⋄ Q←2 2⍴1 0 0 0 ⋄ C←¯1 ¯1
Z←(X+.×Q+.×X)+C+.×X
```

Equality Constraints

```
V←H X;COEF;RHS
⍝
⍝        Constraint functions for GRG example problem
⍝             (2 linear equality constraints)
⍝
COEF←2 4⍴2 1 1 0 1 2 0 1 ⋄ RHS←1 1
V←(COEF+.×X)-RHS
```

APL code

i	Lower Bnd	Upper Bnd
1	0	1
2	0	1
3	0	1
4	0	1

Gradient for objective function

```
G←GRADIENT X;Q;C
⍝
⍝        Gradient for objective function of GRG Example 2
⍝
Q←2 2⍴1 0 0 0 ⋄ C←¯1 ¯1
G←C+2×Q+.×2↑X
G←G,0 0
```

Jacobian of Equality Constraints

```
J←JACOBIAN X;COEF
⍝
⍝        Jacobian matrix of linear equality constraints
⍝                 for GRG example problem 2
⍝
COEF←2 4⍴2 1 1 0 1 2 0 1
J←COEF
```

APL code

Iteration 1

```
x = 0.25 0 0.5 0.75
F(x)= ¯0.1875
    Dependent Index Set: 3 4
    Independent Index Set: 1 2
h(x) = 0 0
Gradient = ¯0.5 ¯1 0 0


Negative of Reduced Gradient = 0.5 1
Search Direction = 0.5 1 ¯2 ¯2.5
(Normalized Search Direction = 0.2 0.4 ¯0.8 ¯1)
```

δ was normalized by scaling so that

$$\max |\delta_j| = 1$$

Computing Maximum Step Size

Based upon the lower & upper bounds:

$$\begin{cases} 0 \le x_1 + \lambda\delta_1 = \frac{1}{4} + \frac{4}{5}\lambda \le 1 & \Rightarrow \quad \lambda \le \frac{5}{4} \times \frac{3}{4} = \frac{15}{16} \\ 0 \le x_2 + \lambda\delta_2 = 0 + \frac{2}{5}\lambda \le 1 & \Rightarrow \quad \lambda \le \frac{5}{2} \\ 0 \le x_3 + \lambda\delta_3 = \frac{1}{2} - \frac{4}{5}\lambda \le 1 & \Rightarrow \quad \lambda \le \frac{5}{4} \times \frac{1}{2} = \frac{5}{8} \\ 0 \le x_4 + \lambda\delta_4 = \frac{3}{4} - \lambda \le 1 & \Rightarrow \quad \lambda \le \frac{3}{4} \end{cases}$$

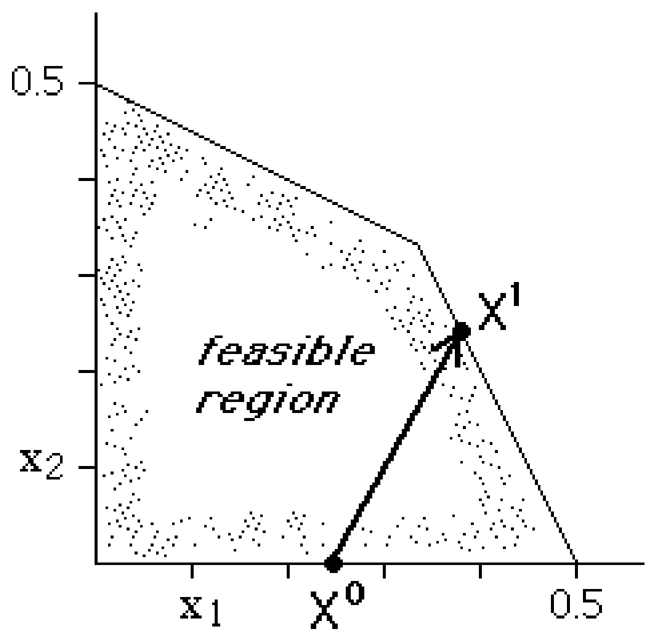
lowest upper bound !

```
Max Step Size = 0.625
```

$= \frac{5}{8}$ *lowest upper bound !*

```
Optimal Step Size = 0.625
   x = 0.375 0.25 0 0.125
   h(x)=0 0, F(x)= ¯0.484375
```

Note that x (which contributed the maximum stepsize) has reached its lower bound!



X_3 is the slack in the inequality constraint, and so GRG has moved to the boundary of that constraint as X_3 decreases to 0.

Dependent variables cannot be at either lower or upper bound, and so X_3 must become independent, and replaced by either X_1 or X_2.

(X_4 is already dependent.)

```
Variable(s) 3 has reached a bound
 and must be removed from D

Variables 1 2 are candidates to enter D

Try entering variable 1
Determinant of J[;D]= 2
```

checking that the Jacobian submatrix is nonsingular!

```
3 is replaced by 1 in Dependent Variable Set.
   h(x)=0 0, F(x)= ¯0.484375
```


Iteration 2

```
x = 0.375 0.25 0 0.125
F(x)= ¯0.484375
    Dependent Index Set: 1 4
    Independent Index Set: 3 2
h(x) = 0 0
Gradient = ¯0.25 ¯1 0 0

Negative of Reduced Gradient = ¯0.125 0.875
Search Direction = ¯0.4375 0.875 0 ¯1.3125
(Normalized Search Direction = ¯0.333333 0.666667 0 ¯

Max Step Size = 0.125
Optimal Step Size = 0.125

   x = 0.333333 0.333333 0 0
   h(x)=0 0, F(x)= ¯0.555556
```

```
Variable(s) 4 has reached a bound
 and must be removed from D
Variables 2 are candidates to enter D

Try entering variable 2
Determinant of J[;D]= 3

4 is replaced by 2 in Dependent Variable Set.
   h(x)=0 0, F(x)= ¯0.555556
```


Iteration 3

```
x = 0.333333 0.333333 0 0
F(x)= ¯0.555556
    Dependent Index Set: 1 2
    Independent Index Set: 3 4
h(x) = 0 0
Gradient = ¯0.333333 ¯1 0 0

Negative of Reduced Gradient = 0.111111 ¯0.555556

Search Direction = ¯0.0740741 0.037037 0.111111 0
(Normalized Search Direction = ¯0.666667 0.333333 1 0

Max Step Size = 0.5
Optimal Step Size = 0.125
   x = 0.25 0.375 0.125 0
   h(x)=¯1.11022E¯16 ¯1.11022E¯16, F(x)= ¯0.5625
```

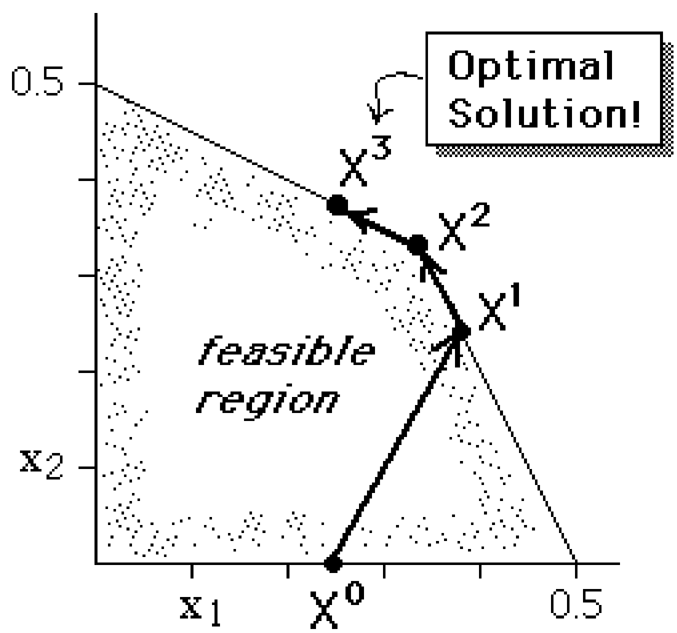
Iteration 4

```
x = 0.25 0.375 0.125 0
F(x)= ¯0.5625
    Dependent Index Set: 1 2
    Independent Index Set: 3 4
h(x) = ¯1.11022E¯16 ¯1.11022E¯16
Gradient = ¯0.5 ¯1 0 0
Negative of Reduced Gradient = 0 ¯0.5

*** GRG HAS CONVERGED ***
```

Generalized Reduced Gradient
Solution

```
x = 0.25 0.375 0.125 0
F(x) = ¯0.5625
∇F(x) = ¯0.5 ¯1 0 0
h(x) = ¯1.11022E¯16 ¯1.11022E¯16
```

Iteration 1

```
x = 0.25 0.25 0.25 0.25
F(x)= ¯0.4375
    Dependent Index Set: 3 4
    Independent Index Set: 1 2
h(x) = 0 0
Gradient = ¯0.5 ¯1 0 0

Negative of Reduced Gradient = 0.5 1

Search Direction = 0.5 1 ¯2 ¯2.5
(Normalized Search Direction = 0.2 0.4 ¯0.8 ¯1)

Max Step Size = 0.25
Optimal Step Size = 0.25
   x = 0.3 0.35 0.05 0
   h(x)=0 0, F(x)= ¯0.56
```



```
Variable(s) 4 has reached a bound
 and must be removed from D

Variables 1 2 are candidates to enter D

Try entering variable 2
Determinant of J[;D]= 2

4 is replaced by 2 in Dependent Variable Set.
   h(x)=0 0, F(x)= ¯0.56
```

©D.L.Bricker, U. of IA, 1999

Iteration 2

```
x = 0.3 0.35 0.05 0
F(x)= ¯0.56
    Dependent Index Set: 3 2
    Independent Index Set: 1 4
h(x) = 0 0
Gradient = ¯0.4 ¯1 0 0

Negative of Reduced Gradient = ¯0.1 ¯0.5

Search Direction = ¯0.1 0.05 0.15 0
(Normalized Search Direction = ¯0.666667 0.333333 1 0)

Max Step Size = 0.45
Optimal Step Size = 0.075

   x = 0.25 0.375 0.125 0
   h(x)=0 0, F(x)= ¯0.5625
```


Iteration 3

```
x = 0.25 0.375 0.125 0
F(x)= ¯0.5625
    Dependent Index Set: 3 2
    Independent Index Set: 1 4
h(x) = 0 0
Gradient = ¯0.5 ¯1 0 0
Negative of Reduced Gradient = 2.22045E¯16 ¯0.5

*** GRG HAS CONVERGED ***
```

Generalized Reduced Gradient Solution

```
x = 0.25 0.375 0.125 0
F(x) = ¯0.5625
∇F(x) = ¯0.5 ¯1 0 0
h(x) = 0 0
```