```
> restart;
  with( linalg ):
Warning: new definition for   norm
Warning: new definition for   trace
```

Solution of Homework Problems from Chapter 3 of Peressini, Sullivan, and Uhl

Prepared by Douglas B. Meade (meade@math.sc.edu)
9 November 1995

WWS URL: http://www.math.sc.edu/~meade/math524/worksheets/ch3hw3.ms

---

>

Problem #14

Note that this problem implicitly is asking for the use of Broyden's Method for the minimization of a scalar-valued function. That is, we will be applying Broyden's Method to the gradient of the function in Problem #11.

> f := x -> 2*x[1]^2 + x[2]^2 - x[1]*x[2];

$$f := x \to 2\,x_1^{\,2} + x_2^{\,2} - x_1\,x_2$$

> g := x -> vector(2, [4*x[1] - x[2], 2*x[2] - x[1]]);

$$g := x \to \mathrm{vector}\left(2, \left[4\,x_1 - x_2,\ 2\,x_2 - x_1\right]\right)$$

>

Our next step is to implement Broyden's Method in Maple. The ingredients needed to define Broyden's Method are: i) the function g, ii) the initial guess x[0], iii) the matrix DD[0], and iv) the number of iterations, N

In addition to the function g defined above, the appropriate values for the first part of this problem are:

> x[0] := [1,4];

$$x_0 := [1, 4]$$

> N := 2;

$$N := 2$$

> DD[0] := matrix(2,2, [[1,0], [0,1]]);

$$DD_0 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

>

N iterations of Broyden's Method

```
> for k from 0 to N-1 do
    x[k+1] := evalm( x[k] - inverse( DD[k] ) &* g( x[k] ) );
    d[k] := evalm( x[k+1] - x[k] );
    y[k] := evalm( g( x[k+1] ) - g( x[k] ) );
    DD[k+1] := evalm( DD[k] + ( (y[k]-DD[k]&*d[k]) &* transpose( d[k] ) ) / dotprod(
   d[k], d[k] ) );
  od;
```

$$x_1 := [\,1 \quad -3\,]$$

$$d_0 := [\,0 \quad -7\,]$$

$$y_0 := [\,7 \quad -14\,]$$

$$DD_1 := \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$$

$$x_2 := \begin{bmatrix} \frac{-5}{2} & \frac{1}{2} \end{bmatrix}$$

$$d_1 := \begin{bmatrix} \frac{-7}{2} & \frac{7}{2} \end{bmatrix}$$

$$y_1 := \begin{bmatrix} \frac{-35}{2} & \frac{21}{2} \end{bmatrix}$$

$$DD_2 := \begin{bmatrix} \frac{5}{2} & \frac{-5}{2} \\ \frac{-1}{2} & \frac{5}{2} \end{bmatrix}$$

>

b. All that is different for this problem is that DD[0] = Hf(x[0]), instead of the identity matrix.

> DD[0] := matrix(2,2, [[4,-1], [-1,2]]);

$$DD_0 := \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix}$$

>

N iterations of Broyden's Method

```
> for k from 0 to N-1 do
    x[k+1] := evalm( x[k] - inverse( DD[k] ) &* g( x[k] ) );
    d[k] := evalm( x[k+1] - x[k] );
    y[k] := evalm( g( x[k+1] ) - g( x[k] ) );
    DD[k+1] := evalm( DD[k] + ( (y[k]-DD[k]&*d[k]) &* transpose( d[k] ) ) / dotprod(
   d[k], d[k] ) );
  od;
```

$$x_1 := [0 \quad 0]$$

$$d_0 := [-1 \quad -4]$$

$$y_0 := [0 \quad -7]$$

$$DD_1 := \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix}$$

$$x_2 := [0 \quad 0]$$

$$d_1 := [0 \quad 0]$$

$$y_1 := [0 \quad 0]$$

```
Error, division by zero
```

Note that the division by zero occurs because d[1]=0. That is, the minimizer of f is reached in only one step. If you increase the value of N in part a., you will see that the iterates for that initial matrix converge to the solution in five steps. Compare this behavior with that observed for this same problem by Steepest Descent (Problem #11).

END OF WORKSHEET

>