```
> restart;
> with(LinearAlgebra):
  with(plots):
> A := < <3,2> | <2,6> >;
  b := <2,-8>;
```

$$A := \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}$$

$$b := \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

Quadratic Form

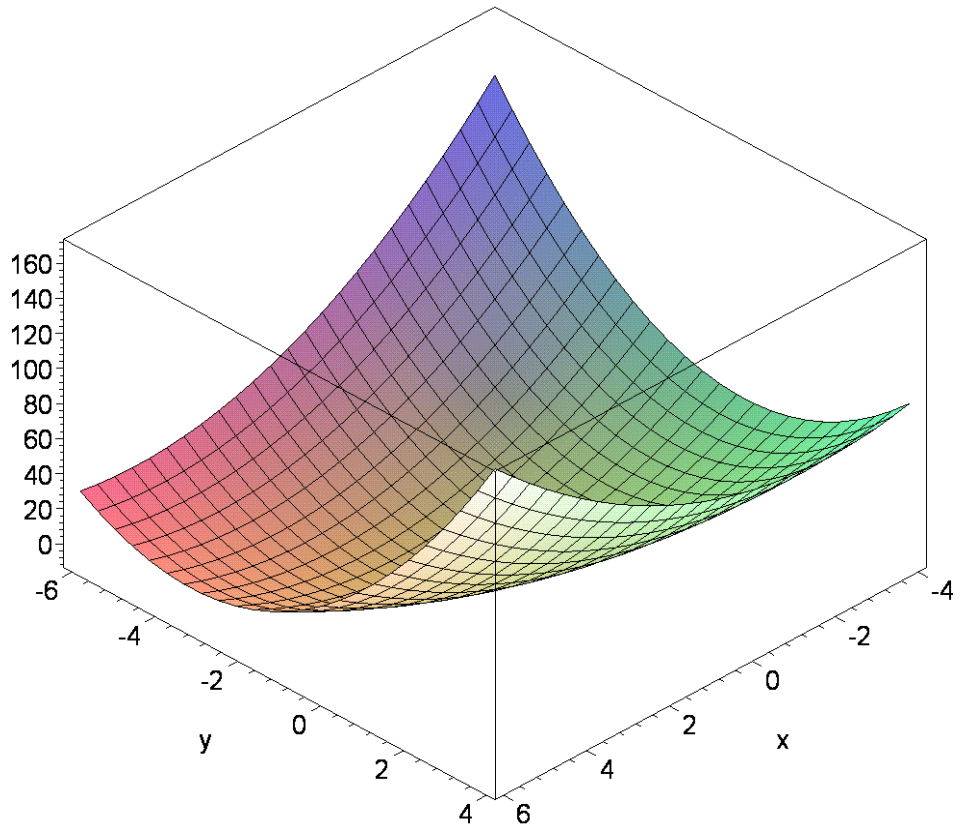
```
> f := x-> 1/2*Transpose(x).A.x - Transpose(b).x;
```

$$f := x \rightarrow \left(\left(\frac{1}{2} \text{LinearAlgebra:-Transpose}(x) \right) . A . x \right) - ((\text{LinearAlgebra:-Transpose}(b)) . x)$$

```
> f(b);
```

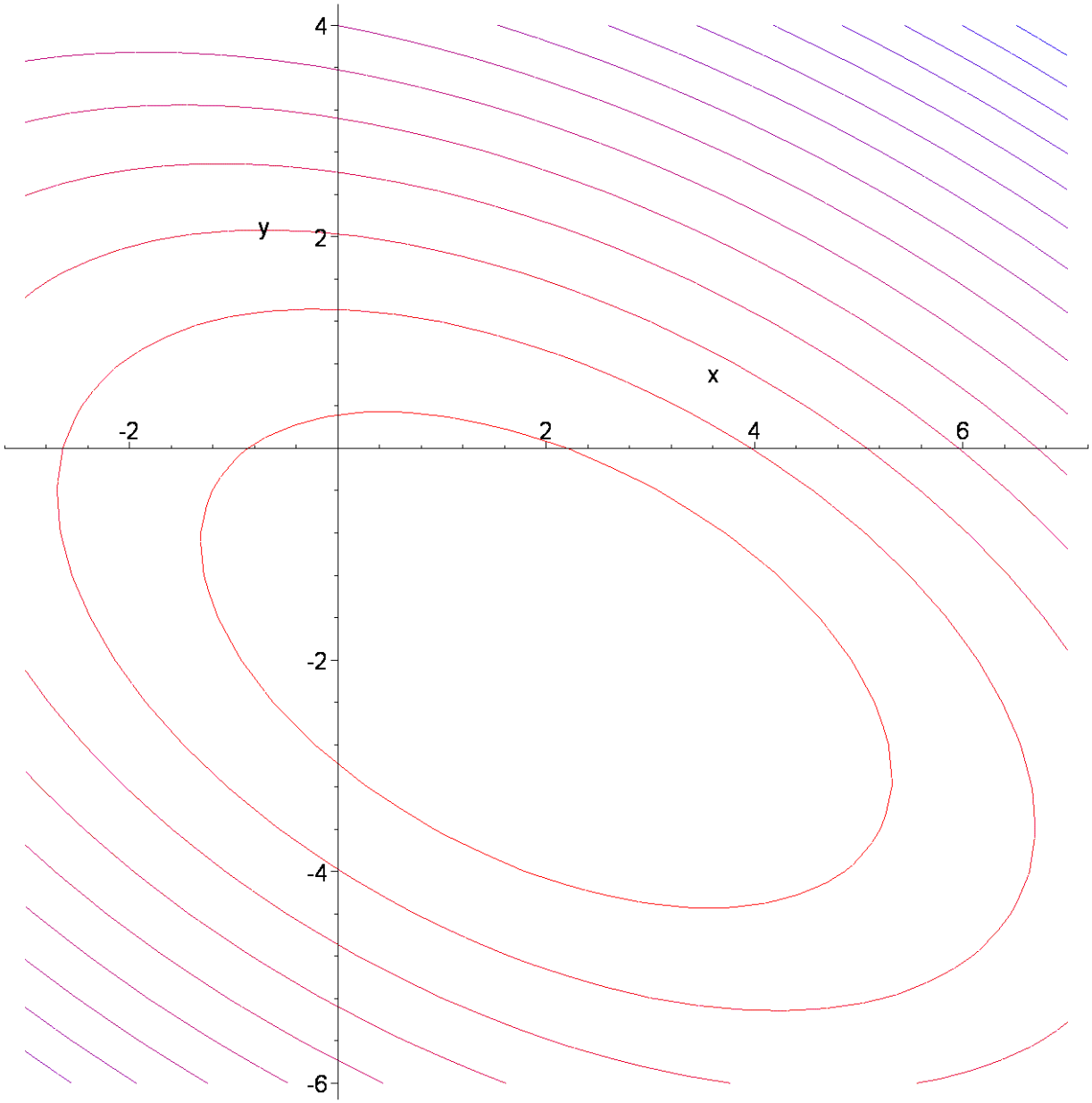
$$98$$

```
> plot3d(f(<x,y>), x=-4..6, y=-6..4,axes=BOXED);
```



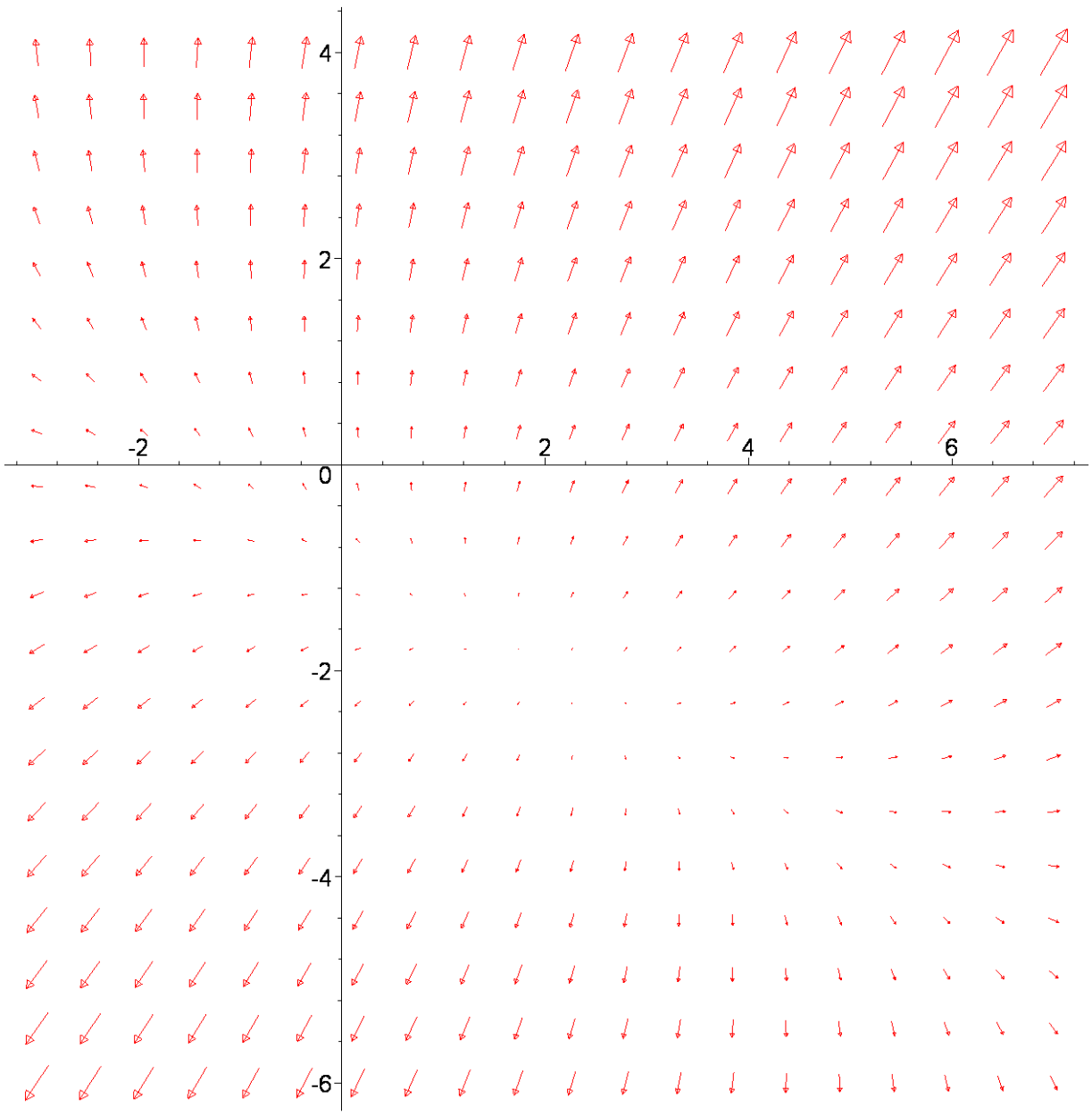

```
> contplot := contourplot(f(<x,y>), x=-3..7,
  y=-6..4,coloring=[red,blue],contours=15,scaling=CONSTRAINED):

  display(contplot);
```



Steepest Descent

```
> gradplot( f(<x,y>), x=-3..7, y=-6..4,arrows=SLIM,color=red);
```



perform one step of Steepest Descent:

```
> steepdesc := proc(A::Matrix, b::Vector, x::Vector)
      local r, alpha;
      # compute residual:
      r := b - A.x;
      # compute alpha
      alpha := ( Transpose(r).r) / ( (Transpose(r).A).r );
      # compute new approximation:
      return x + alpha*r;
  end proc;
```

```
steepdesc := proc(A::Matrix, b::Vector, x::Vector)
local r, α;
      r := b – ( A . x );
      α := ( LinearAlgebra:-Transpose( r ) ) . r / ( LinearAlgebra:-Transpose( r ) ) . A . r;
      return x + α*r
end proc
```

```
> xs[0] := <-3,-3>;
```

$$xs_0 := \begin{bmatrix} -3 \\ -3 \end{bmatrix}$$

```
> for i from 1 to 6 do
     xs[i] := steepdesc(A,b,xs[i-1]);
  end do;
```

$$xs_1 := \begin{bmatrix} \frac{-1208}{3491} \\ \frac{-1753}{3491} \end{bmatrix}$$

$$xs_2 := \begin{bmatrix} \frac{427352}{352591} \\ \frac{-760748}{352591} \end{bmatrix}$$

$$xs_3 := \begin{bmatrix} \frac{2006704822}{1230895181} \\ \frac{-2171235748}{1230895181} \end{bmatrix}$$

$$xs_4 := \begin{bmatrix} \frac{233202924782}{124320413281} \\ \frac{-251728406918}{124320413281} \end{bmatrix}$$

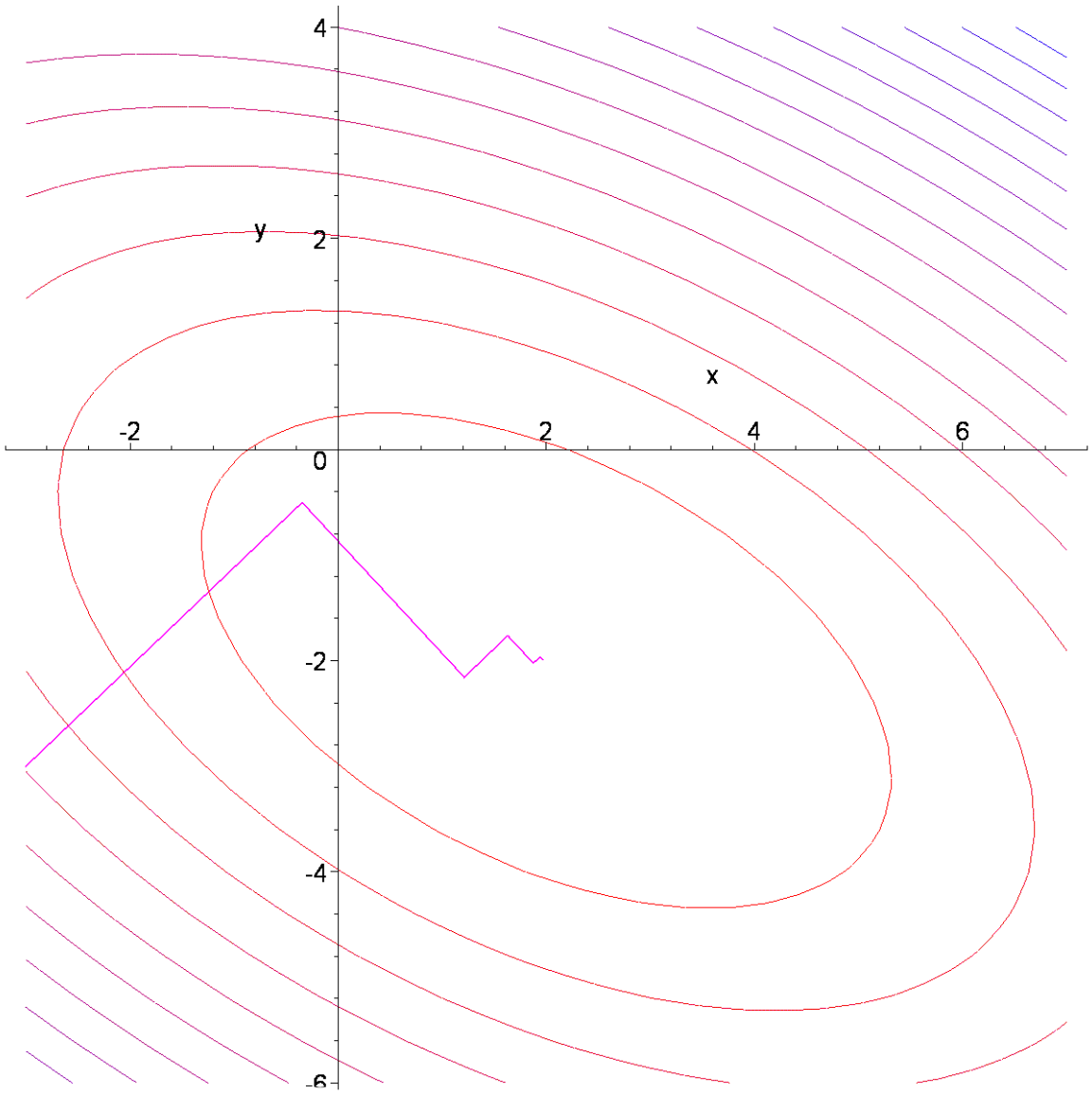
$$xs_5 := \begin{bmatrix} \frac{842717842412302}{434002562763971} \\ \frac{-851860167846418}{434002562763971} \end{bmatrix}$$

$$xs_6 := \begin{bmatrix} \frac{86810695228014662}{43834258839161071} \\ \frac{-87840082168383638}{43834258839161071} \end{bmatrix}$$

```
> xs[0][1], xs[0][2];
```

$$-3, -3$$

```
> steepplot := plot( [seq( [xs[i][1], xs[i][2]], i=0..6)],
  color=magenta,thickness=2):
  display(contplot,steepplot);
```



Influence of the Condition Number

```
> A := < <4,2> | <2,100> >;
  b := <2,-8>;
```

$$A := \begin{bmatrix} 4 & 2 \\ 2 & 100 \end{bmatrix}$$

$$b := \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

```
> f := x-> 1/2*Transpose(x).A.x - Transpose(b).x;
```

$$f := x \rightarrow \left(\left(\frac{1}{2} \text{LinearAlgebra:-Transpose}(x) \right) . A . x \right) - ((\text{LinearAlgebra:-Transpose}(b)) . x)$$

```
> xs[0] := <-8,-1/2>;
  for i from 1 to 12 do
     xs[i] := steepdesc(A,b,xs[i-1]);
  end do;
```

$$xs_0 := \begin{bmatrix} -8 \\ \frac{-1}{2} \end{bmatrix}$$

$$xs_1 := \begin{bmatrix} \frac{-526949}{69884} \\ \frac{22863}{87355} \end{bmatrix}$$

$$xs_2 := \begin{bmatrix} \frac{-3973392393}{620395210} \\ \frac{-1051260131}{2481580840} \end{bmatrix}$$

$$xs_3 := \begin{bmatrix} \frac{-522936430235277}{86711397711280} \\ \frac{5308104527363}{27097311784775} \end{bmatrix}$$

$$xs_4 := \begin{bmatrix} \frac{-3931329172270197117}{769780433181888200} \\ \frac{-1113130469796453499}{3079121732727552800} \end{bmatrix}$$

$$xs_5 := \begin{bmatrix} \frac{-516760975363907682355773}{10759067158496614993760 0} \\ \frac{119649815120160380653 3}{8405521217575480463875} \end{bmatrix}$$

$$xs_6 := \begin{bmatrix} \frac{-387000604773888307172278 7613}{95513618699553699607104 4000} \\ \frac{-118815001156493109700144 8011}{38205447479821479842841 76000} \end{bmatrix}$$

$$xs_7 := \begin{bmatrix} \frac{-507890892825137924921929237606797}{133497474583992214866857677792000} \\ \frac{515212837901554355585018897581}{52147451009371958932366280387 50} \end{bmatrix}$$

$$xs_8 := \begin{bmatrix} \frac{-378470193642197829158466050934252215 7}{11851238306193908874805290345984800 00} \\ \frac{-127946897237073200316667198606913317 9}{47404953224775635499221161383939200 00} \end{bmatrix}$$

$$xs_9 := \begin{bmatrix} \frac{-495666301522030562791448729005356911946333}{16564238755801102556137858210776035264000 0} \\ \frac{4101061574463870847757913202387257141 59}{64704057639848056859913508635843887750 00} \end{bmatrix}$$

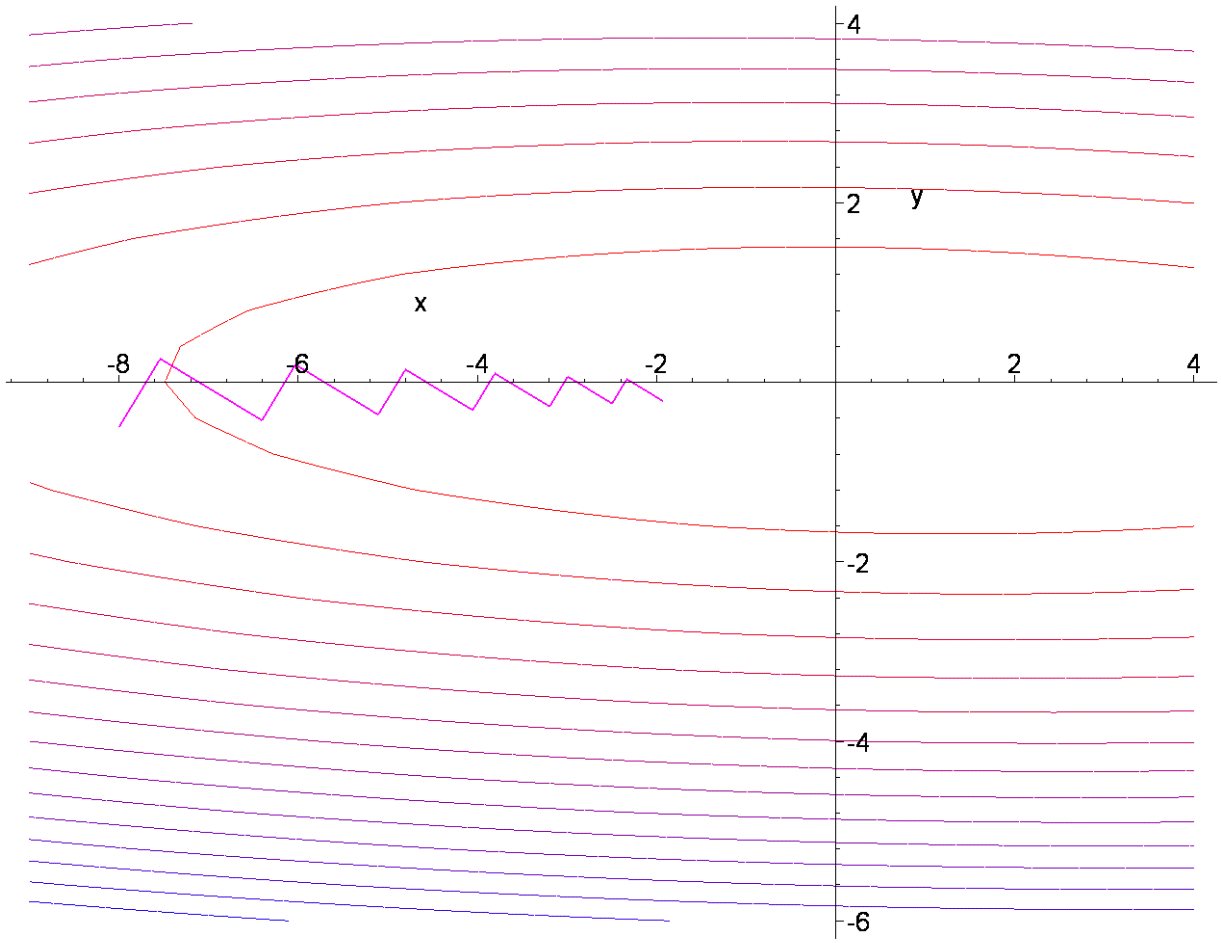
$$xs_{10} := \begin{bmatrix} \frac{-366955820055822084047261964381892475230771737 3}{14704902955462428794211383626616425305616000 00} \\ \frac{-139099596397960278535146001644524379666222673 1}{58819611821849715176845534506465701222464000 00} \end{bmatrix}$$

$$xs_{11} := \begin{bmatrix} \frac{-47926832637509288343676448674441053486442652013623 7}{2055274876279072747709336666724924532115337088000000} \\ \frac{27759570263674078936903644017253427358356119495 1}{8028417485465127920739596354394236453575535500000} \end{bmatrix}$$

$$xs_{12} := \begin{bmatrix} \frac{-35173047283389941728547895581336291473745470219526867 97}{1824570271416746831778963625885051753385390499872000000} \\ \frac{-15275806069408378266985171445249744872521173232927272 59}{7298281085666987327115854503540207013541561999488000000} \end{bmatrix}$$

```
> steepplot := plot( [seq( [xs[i][1], xs[i][2]], i=0..12)],
  color=magenta,thickness=2):
  contplot := contourplot(f(<x,y>), x=-9..4,
  y=-6..4,coloring=[red,blue],contours=15,scaling=CONSTRAINED):

  display(contplot,steepplot);
```



Comment:
The speed of convergence also depends on the choice of the starting approximation.

Try to find better starting approximations for the example above!
However: In a numerical method, one can usually never rely on the fact that a favourable starting condition is available.

```
>
```

Conjugate Gradient

```
> conjgrad := proc(A::Matrix, b::Vector, xstart::Vector,
  it::posint)
     local i, r, r2, d, alpha, beta, xs;
     xs[0] := xstart;
     # compute initial residual and search direction:
     r := b - A.xstart; d := r;
     for i from 1 to it do
        # compute alpha
        r2 := Transpose(r).r;
        alpha := r2 / ( (Transpose(d).A).d );
        # compute new approximation:
        xs[i] := xs[i-1] + alpha*d;
        # update residual
        r := r - alpha* A.d;
        # compute beta and next search direction:
        beta := ( Transpose(r).r) / r2;
        d := r + beta*d;
     end do;
     return xs;
  end proc;
```

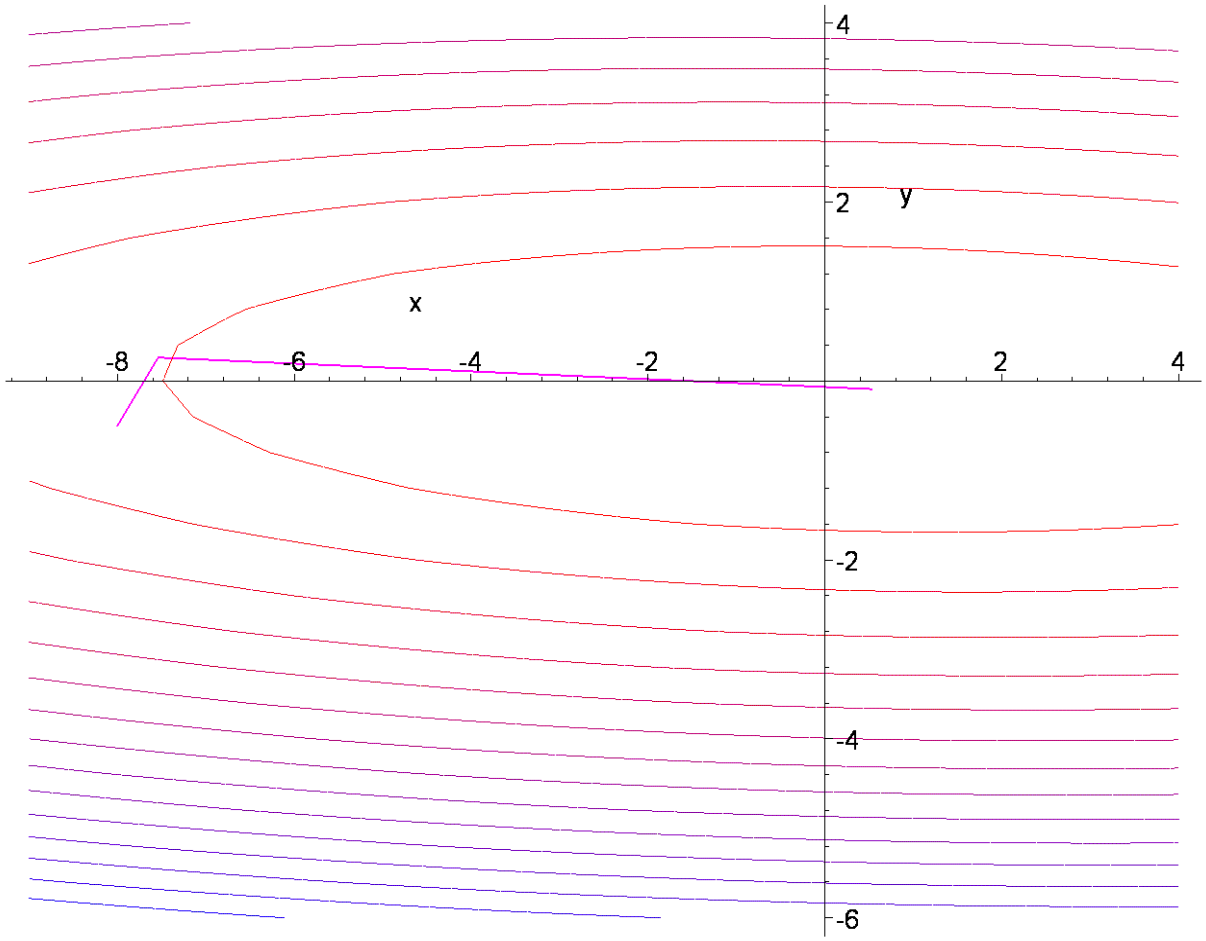
```
conjgrad := proc(A::Matrix, b::Vector, xstart::Vector, it::posint)
local i, r, r2, d, α, β, xs;
     xs[ 0 ] := xstart;
     r := b – ( A . xstart );
     d := r;
     for i to it do
          r2 := ( LinearAlgebra:-Transpose( r ) ) . r;
          α := r2 / ( LinearAlgebra:-Transpose( d ) ) . A . d;
          xs[ i ] := xs[ i – 1 ] + α*d;
          r := r – ( ( α*A ) . d );
          β := ( LinearAlgebra:-Transpose( r ) ) . r / r2;
          d := r + β*d
     end do;
     return xs
end proc
```

```
> xs := conjgrad(A,b,<-8,-1/2>,2);
```

$$xs := xs$$

```
> steepplot := plot( [seq( [xs[i][1], xs[i][2]], i=0..2)],
  color=magenta,thickness=2):
  contplot := contourplot(f(<x,y>), x=-9..4,
  y=-6..4,coloring=[red,blue],contours=15,scaling=CONSTRAINED):
```

```
> display(contplot,steepplot);
```




```
>
>
```