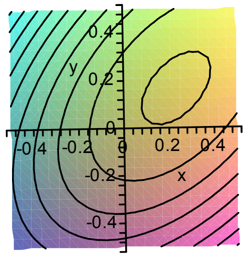
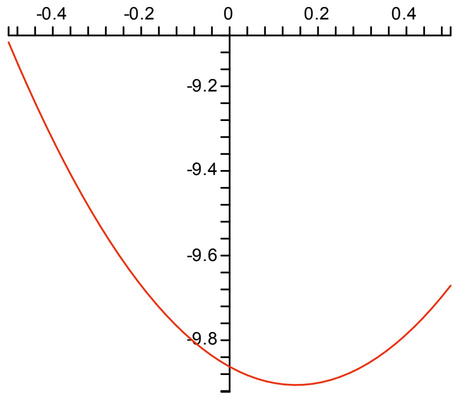
```
> restart;
  E:=(dx1,dx2)->k1*(dx1-0.1)^2+k2*(dx1-dx2-0.2)^2+k3*(dx2-0.3)^2-10;
```

$$E := (dx1, dx2) \to k1\,(dx1 - 0.1)^2 + k2\,(dx1 - dx2 - 0.2)^2 + k3\,(dx2 - 0.3)^2 - 10$$

```
> k1:=1:k2:=1:k3:=1:
> plot3d(E(x,y),x=-0.5..0.5,y=-0.5..0.5);
```



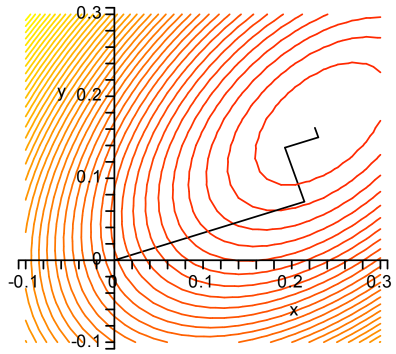
```
> with(LinearAlgebra):
  p:=Vector([0,0]):
  g:=Vector([1,0]):
> plot(E(op(convert(p+x*g,list))),x=-0.5..0.5);
```



```
> FindMin1Dim:=proc(p::Vector,g::Vector)
    return solve(diff(E(op(convert(p+x*g,list))),x)=0,x);
  end proc:
> dFunc:=proc(p::Vector)
    local dx,dy;
    dx:=subs({x=p[1],y=p[2]},diff(E(x,y),x));
    dy:=subs({x=p[1],y=p[2]},diff(E(x,y),y));
    return Vector([dx,dy]);
  end proc:
```

Steepest descent（最急降下法）

```
> p_s:=[]:
  p:=Vector([0,0]):
  for i from 1 to 5 do
    p_s:=[op(p_s),convert(p,list)];
    g:=dFunc(p);
    xx:=FindMin1Dim(p,g);
    p:=p+xx*g;
  end do:
> with(plots):
  pp1:=pointplot(p_s,connect=true):
Warning, the name changecoords has been redefined
> c1:=contourplot(E(x,y),x=-0.1..0.3,y=-0.1..0.3,contours=40):
> display(pp1,c1);
```



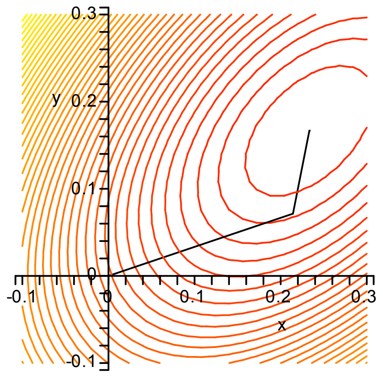
Conjugate gradient(Fletcher-Reeves, Polak-Ribiere)

```
> p_s:=[]:
  p:=Vector([0,0]):
  p_s:=[op(p_s),convert(p,list)]:
  xi:=dFunc(p):
  g:=-xi:
  h:=g:
```

```
xi:=g:
```

```
> for i from 1 to 3 do
    xx:=FindMin1Dim(p,xi):
    p:=p+xx*xi:
    p_s:=[op(p_s),convert(p,list)]:
    xi:=dFunc(p):
    gg:=g.g:
    dgg:=xi.xi:#(xi+g).xi;
    gam:=dgg/gg:
    g:=-xi:
    h:=g+gam*h:
    xi:=h:
  end do:
```

```
> pp2:=pointplot(p_s,connect=true):
  display(pp2,c1);
```

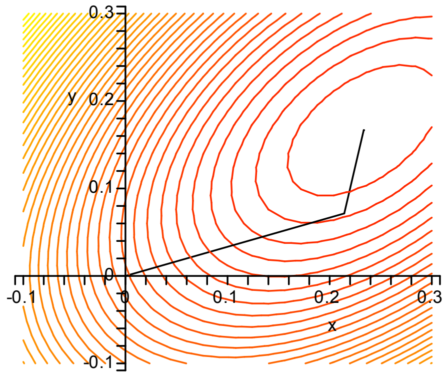


Variable metric(quasi-Newton)

```
> p_s:=[]:
  p:=Vector([0,0]):
  hessian:=Matrix(2,2,[[1,0],[0,1]]):
  p_s:=[op(p_s),convert(p,list)]:
  g:=dFunc(p):
  xi:=-g:
```

```
> for i from 1 to 3 do
    xx:=FindMin1Dim(p,xi);
    p:=p+xx*xi;
    p_s:=[op(p_s),convert(p,list)];
    dg:=g;
    g:=dFunc(p);
    dg:=g-dg;
    hdg:=hessian.dg;
    fac:=dg.xi;
    fae:=dg.hdg;
    fac:=1/fac;
    fad:=1/fae;
    dg:=fac*xi-fad*hdg;
    hessian:=hessian+fac*OuterProductMatrix(xi,xi)
       -fad*OuterProductMatrix(hdg,hdg)
       +fae*OuterProductMatrix(dg,dg);
    xi:=-hessian.g;
  end do:
```

```
> pp3:=pointplot(p_s,connect=true):
  display(pp3,c1);
```



```
>
```