

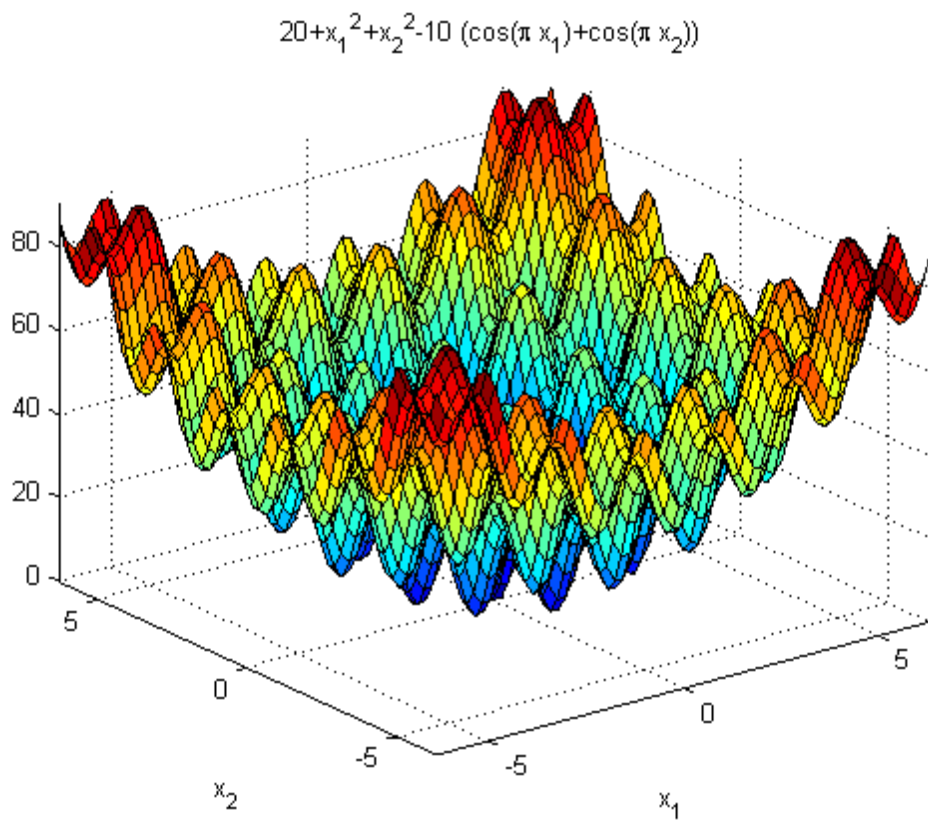
Contents

- [Rastrigin](#)
- [fminunc](#)
-
-
-
-
-

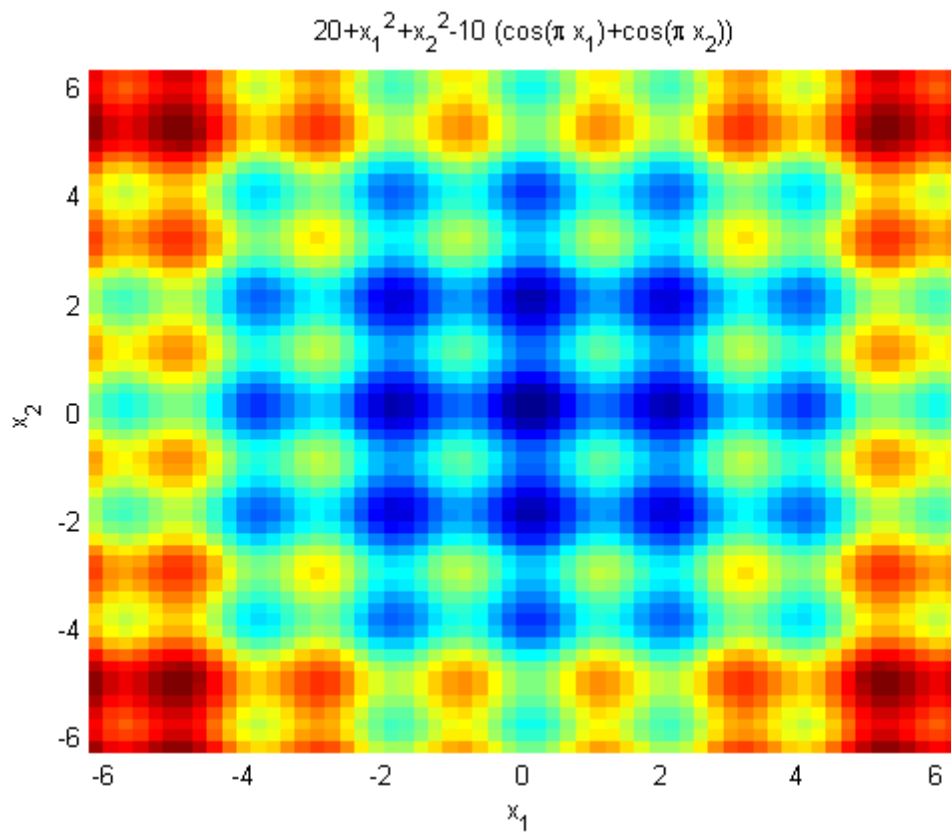
著名的无约束测试基准函数 Rastrigin函数

$$f(x_1,x_2) = 20 + x_1^2 + x_2^2 - 10(\cos \pi x_1 + \cos \pi x_2)$$

```
ezsurf('20+x1^2+x2^2-10*(cos(pi*x1)+cos(pi*x2))')
```



```
ezsurf('20+x1^2+x2^2-10*(cos(pi*x1)+cos(pi*x2))')  
view(0,90), shading flat %俯视图;
```



fminunc函数求解最小值

```
f=@(x)20+x(1)^2+x(2)^2-10*(cos(pi*x(1))+cos(pi*x(2)));
x1=fminunc(f,[2,3]);f(x1)
x2=fminunc(f,[-1,2]);f(x2)
x3=fminunc(f,[8,2]);f(x3)
x4=fminunc(f,[-4,6]);f(x4)
% 观察最优解的情况。多数情况下并不是全局最小点。
% 为避免局部最小，可以采用并行算法，如遗传算法，粒子群，蚁群算法
```

警告: Gradient must be provided for trust-region
algorithm;
using line-search algorithm instead.

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the function tolerance.

ans =

7.8409

警告: Gradient must be provided for trust-region
algorithm;
using line-search algorithm instead.

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the function tolerance.

ans =

3.9205

警告: Gradient must be provided for trust-region
algorithm;
using line-search algorithm instead.

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the function tolerance.

ans =

66.6213

警告: Gradient must be provided for trust-region
algorithm;
using line-search algorithm instead.

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the function tolerance.

ans =

50.9570

(a,b)

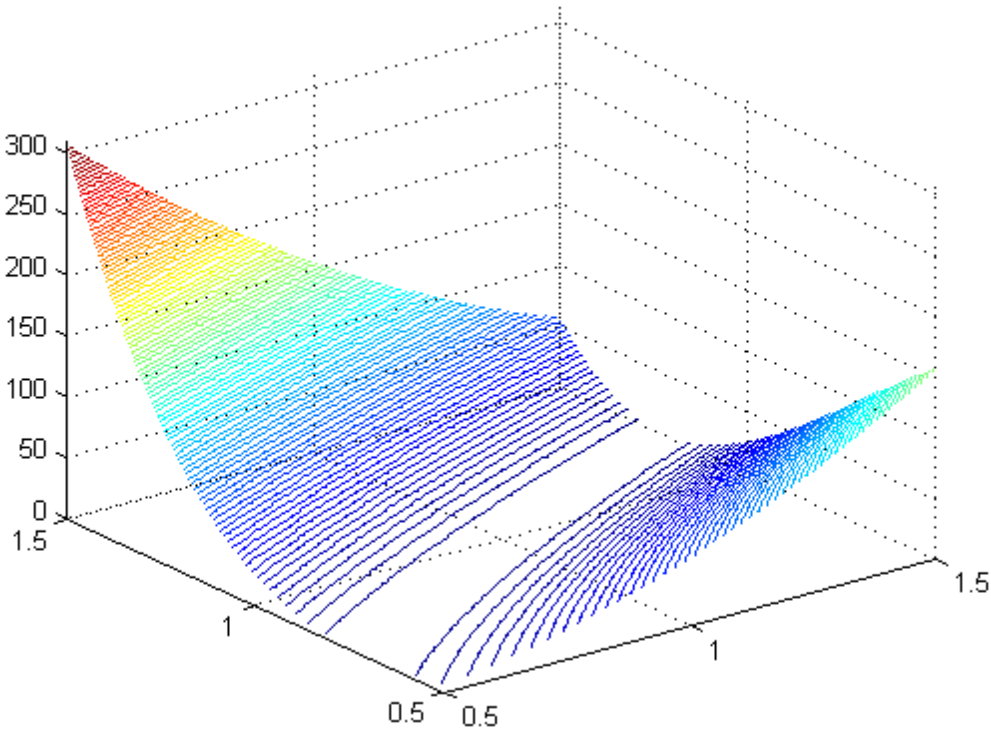
```
% function [x,f0]=fminunc_global(f,a,b,n,N,varargin)
% k0=0;f0=Inf;
% if strcmp(class(f),'struct')
%     k0=1;
% end
% for i=1:N
%     x0=a+(b-a)*rand(n,1);
%     if k0==1
%         f.x0=x0;
%         [x1 f1 key]=fminunc(f);
%     else
%         [x1 f1 key]=fminunc(f,x0,varargin{:});
%     end
%     if key>0 & f1<f0
%         x=x1;
%         f0=f1;
%     end
% end
% f=@(x)20+x(1)^2+x(2)^2-10*(cos(pi*x(1))+cos(pi*x(2)));
% [x,f0]=fminunc_global(f,-2*pi,2*pi,2,50); %获取全局最优解
```

```
F=[]; tic, for i=1:100, [x,f0]=fminunc_global(f,-100,100,2,50); F=[F,f0]; end, toc
```

利用梯度信息求解最优化问题

Rosenbrock函数 $f(x_1, x_2)=100(x_2-x_1^2)^2+(1-x_1)^2$ 无约束最优化问题。

```
[x, y]=meshgrid(0.5:0.01:1.5);
z=100*(y.^2-x).^2+(1-x).^2;
contour3(x, y, z, 100), zlim([0, 310])
%最小值在图中一个很窄的白色区域内，目标函数又称为香蕉函数。
```



```
f=@(x) 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
ff=optimset; ff.TolX=1e-10;
ff.TolFun=1e-20;
x=fminunc(f, [0;0], ff)
% 真实最小值点为[1, 1]
```

警告: Gradient must be provided for trust-region algorithm;
using line-search algorithm instead.

Local minimum possible.

fminunc stopped because it cannot decrease the objective function along the current search direction.

x =

1.0000

参考文献

%薛定宇，《高等应用数学问题的MATLAB求解》，第四版，清华大学出版社，2018年6月。