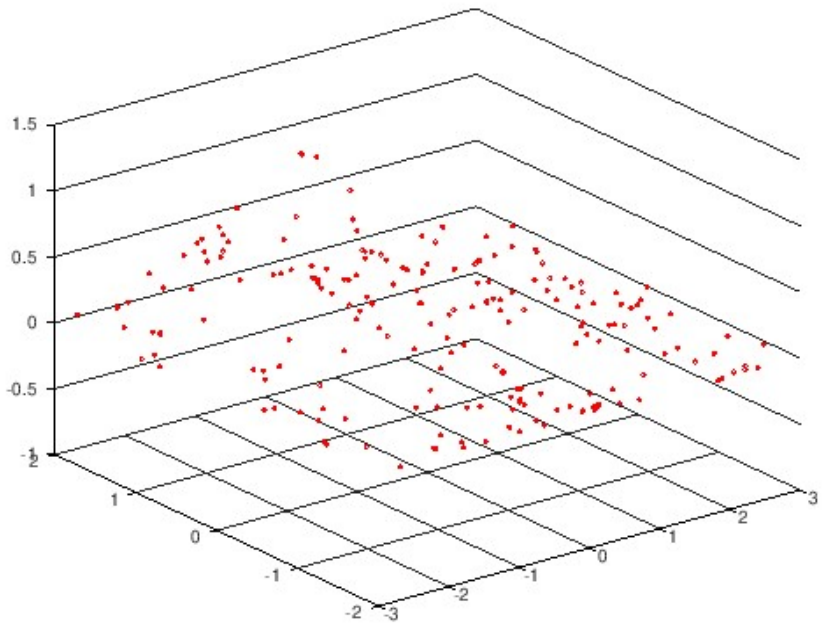
2.2 Think in Matrix

In [26]:
```
x = -3+6*rand(199,1);  y = -2 + 4*rand(199,1);
```

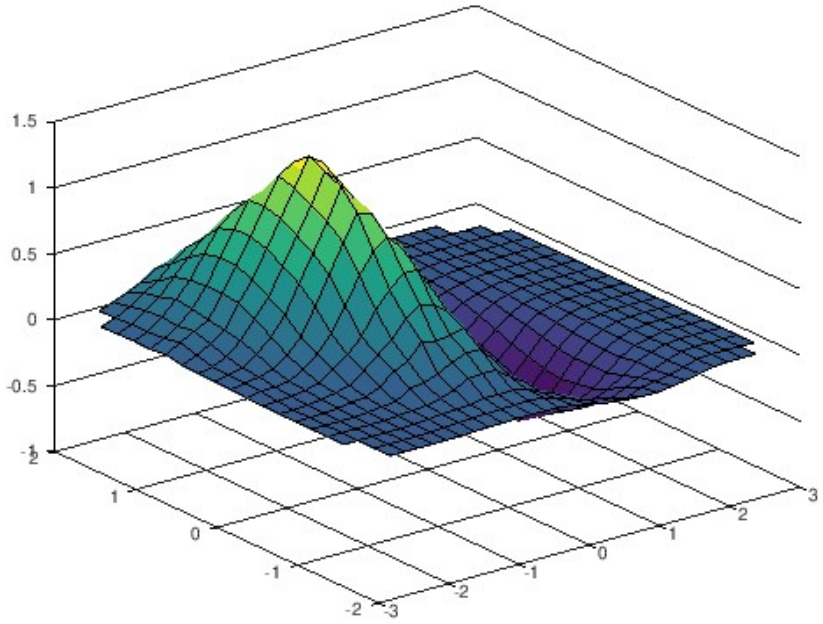
In [30]:
```
z = (x.*x - 2*x).*exp(-x.*x - y.*y - x.*y);
scatter3(x,y,z,'r')
```



In [31]:
```
[xx,yy] = meshgrid(-3:0.2:3, -2:0.2:2);  % a find grid for interpolation
```

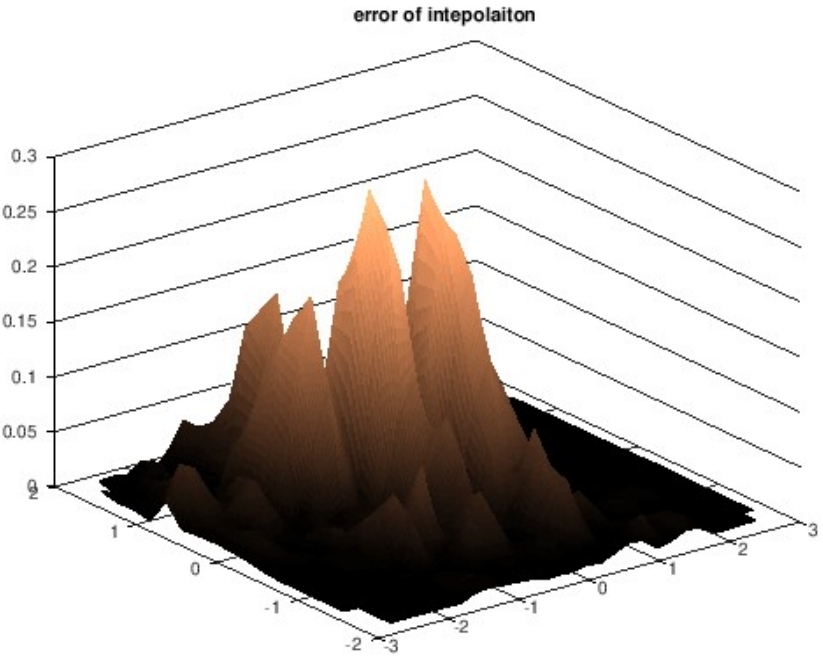
In [32]:
```
z3 = griddata(x,y,z,xx,yy,'linear');  % method 'cubic' has not implemented
```

In [33]:
```
surf(xx,yy,z3)
```



In [ ]:
```
% analytic value on find grid for comparision
zz = (xx.*xx - 2*xx).*exp(-xx.*xx - yy.*yy - xx.*yy);
```

In [44]:
```
surf(xx,yy,abs(z3-zz)); title('error of intepolaiton'); colormap copper; shading interp
```



数组运算示例

In [1]:
```
clear; load triangle; whos
```

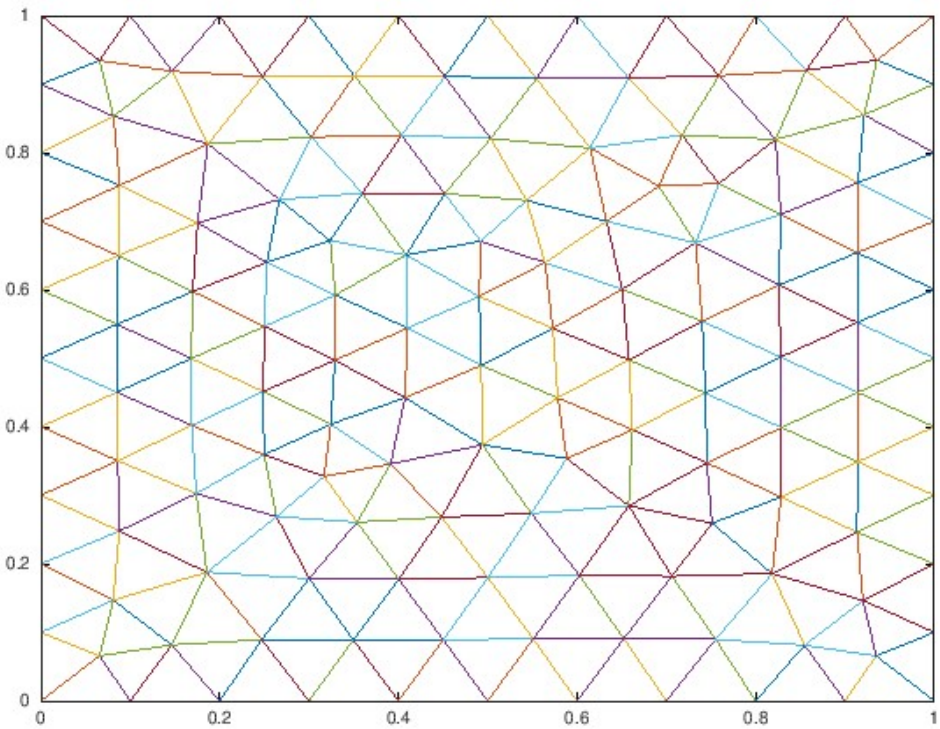
```
Variables in the current scope:

   Attr Name        Size                     Bytes  Class
   ==== ====        ====                     =====  =====
        tri       240x3x2                   11520  double

Total is 1440 elements using 11520 bytes
```

In [56]:
```
tri3 = permute(tri, [2,3,1]);  size(tri3)    % adjust it onsite!
```

```
ans =

     3     2   240
```

In [58]:
```
plot(tri3([1:end,1],1,:),tri3([1:end,1],2,:));
```

```
warning: plot: N-d inputs have been squeezed to less than three dimensions
warning: called from
    __plt__ at line 74 column 11
    plot at line 223 column 10
```



练习: 求每个三角形面积？ (参考PPT2的P20)

In [ ]:
```
```

附：补充例子展示

一、Statistics中的向量化概念

常用统计量I

- 均值mean(X,DIM)

$$E(x) = \sum_{k=1}^{n} x_k p_k$$

- 方差var(X,DIM):

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- 标准差std(X,DIM):

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

例：随机取 8 只活塞环，测得其直径如下(mm)。请计算平均值、方差和标准差。

$$74.001, 74.005, 74.003, 74.001, 74.000, 73.998, 74.006, 74.002$$

In [13]:
```
x = [74.001, 74.005, 74.003, 74.001, 74.000, 73.998, 74.006, 74.002];
```

In [16]:
```
y = [x;x;x]
```

```
y =

   74.001   74.005   74.003   74.001   74.000   73.998   74.006   74.002
   74.001   74.005   74.003   74.001   74.000   73.998   74.006   74.002
   74.001   74.005   74.003   74.001   74.000   73.998   74.006   74.002
```

In [18]:
```
mean(y,2)
```

```
ans =

   74.002
   74.002
   74.002
```

In [19]:
```
mean(x)
```

```
ans =  74.002
```

In [ ]:

In [ ]:

In [ ]:

此外，还提供了以下几个求平均函数：

- nanmean: 求随机变量算术平均
- geomean: 求随机变量几何平均
- harmmean: 求随机变量和谐平均
- trimmean: 求随机变量调和平均

In [ ]:

常用统计量II

- 偏斜度(\alert{skewness})
$$v_1 = E\left[\left(\frac{x - E(x)}{\sqrt{D(x)}}\right)^3\right]$$
其中，$v_1 > 0$ 说明分布形状偏右；$v_1 < 0$ 表示偏左；
- 峰度(\alert{kurtosis})
$$v_2 = E\left[\left(\frac{x - E(x)}{\sqrt{D(x)}}\right)^4\right]$$
其中，$v_2 > 0$说明分布偏离正态分布严重，有较大“尾巴”；$v_2 = 0$恰好是正态分布

例：15名学生体重(单位：kg)分别为
$$75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5, 66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0$$
请计算他们体重的均值、标准差、偏斜度和峰度。

In [ ]:

常用统计量III

对于二维随机变量(X, Y)，除了X和Y的数学期望和方差外，还有：

- 协方差(cov)
$$cov(x, y) = E\big(x - E(x)\big)\big(y - E(y)\big)$$
- 协方差矩阵
$$cov(X, Y) = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$
其中，$C_{11} = E\{[X - E(X)]^2\}, C_{12} = E\{[X - E(X)][Y - E(Y)]\}, C_{21} = E\{[X - E(X)][Y - E(Y)]\}, C_{22} = E\{[Y - E(Y)]^2\}$
- 相关系数(corrcoef)
$$corrcoef(x, y) = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

这里仅作介绍，不再举例

In [ ]:

参数估计

这是一类重要的统计推断的方法，就是利用样本对总体进行统计推断的方法。

例: 设某种清漆的9个样品，起干燥时间（小时）分别为

$$time = [6.0, 5.7, 5.8, 6.5, 7.0, 6.3, 5.6, 6.1, 5.0]$$

已知干燥时间总体服从正态分布$N(\mu, \sigma^2)$。那么,μ和σ的置信度为0.95的置信区间为

$$[Mu, Sig, MuRange, SigRange] = normfit(time, 0.05);$$

- 类似参数估计函数还有: binofit,expfit, gamfit, poissfit, unifit, weibfit等

In [21]:
```
pkg load statistics
```

假设检验

假设检验是统计推断另一类重要问题。在总体的分布函数完全未知或只知道形式但不知到具体参数的情况下，提出的某些关于总体的假设并验证该假设是否成立。实际使用中主要有如下几种:

- 单个总体$N(\mu, \sigma^2)$均值μ的检验：ztest、ttest
- 两个正态总体均值差的检验(t检验)：ttest2
- 秩和检验：ranksum
- 中值检验：signrank、signtest

In [23]:
```
ttest
```

```
error: 'ttestr' undefined near line 1 column 1
```

In [ ]:
```

```

例：车间包装葡萄糖每袋重量是一个服从正态分布的随机变量。正常时均值为0.5kg,标准差为0.015。随机抽取9袋糖，称得重量依次为

$$w = [0.497, 0.506, 0.518, 0.524, 0.498, 0.511, 0.52, 0.515, 0.512].$$

请问机器是否工作正常?

解：在显著水平为0.05、原假设为$mean = 0.5, std = 0.015$时，检验

$$h = ztest(w, 0.5, 0.015, 0.05, 0)$$

In [ ]:
```

```

例: 某个电子元件寿命服从正态分布，而μ, σ均**未知**。测得16个元件寿命（小时）为

$$t = [159, 280, 101, 212, 224, 379, 179, \$\$264, 222, 362, 168, 250, 149, 260, 485, 170].$$

问是否有理由相信元件的平均寿命大于225小时?

解: 备择假设$mean(t) > 225$，因此此处假设$mean(t) \leq 225$:

$$h = ttest(t, 225, 0.05, 1)$$

In [ ]:

二、Numerical Optimization中的向量化概念

In [24]:
```
pkg load optim
```

In [3]:
```
linprog
```

```
error: Invalid call to linprog.  Correct usage is:

 -- Function File: X = linprog (F, A, B)
 -- Function File: X = linprog (F, A, B, AEQ, BEQ)
 -- Function File: X = linprog (F, A, B, AEQ, BEQ, LB, UB)
 -- Function File: [X, FVAL] = linprog (...)

Additional help for built-in functions and operators is
available in the online version of the manual.  Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at http://www.octave.org and via the help@octave.org
mailing list.
```

In [ ]:

例1：某车间生产A、B两种产品，为了生产A和B，所需原料分别为2和3个单位，而所需的工时分别为4和2个单位。现在可以应用的原材料为100个单位，工时为120个单位，每生产一台A和B分别可获得利润6元和4元。那么应该安排生产A、B各多少台才能获得最大利润?

解：根据题意不难得到下述数学模型：

$$\begin{aligned} max\ \ z &= 6x + 4y \\ s.t. 2x + 3y &\leq 100 \\ 4x + 2y &\leq 120 \\ x, y &\geq 0 \end{aligned}$$

为了能调用标准的函数（往往是求极小值），将上述求极大值问题转化成求极小值。那么

$$\begin{aligned} \min z\ \ &= -6x - 4y \\ s.t. 2x + 3y &\leq 100 \\ 4x + 2y &\leq 120 \\ x, y &\geq 0 \end{aligned}$$

经过上述建模后，我们就可以用标准的求极小函数，求解过程参考如下脚本：

In [25]:
```
c = [-6,-4]';  a = [2,3;4,2]; b=[100,120]';
vlb=[0,0]';  vub=[Inf,Inf]';
[x,zval] = linprog(c,a,b,[],[],vlb,vub)
```

```
x =

   20
   20

zval = -200
```

一个练习:[四元线性规划]

$$\max\left(\frac{3}{4}x_1 - 150x_2 + \frac{1}{50}x_3 - 6x_4\right)$$

subject to

$$\begin{cases} \frac{1}{4}x_1 - 60x_2 - \frac{1}{50}x_3 + 9x_4 & \leq & 0 \\ -\frac{1}{2}x_1 + 90x_2 + \frac{1}{50}x_3 - 3x_4 & \geq & 0 \\ x_3 \leq 1, x_1 \geq -5, x_2 \geq -5, x_3 \geq -5, x_4 & \geq & -5 \end{cases}$$

In [ ]:
```

```

例2：求解如下二次规划问题

$$\begin{aligned} \min & f(x) = \frac{1}{2}(x_1, x_2)\begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - (2,6)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ S.T & \\ & x_1 + x_2 \leq 2 \\ & -x_1 + 2x_2 \leq 2 \\ & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

In [26]:
```
h = [1,-1; -1,2]; c = [-2,-6];
a = [1,1;-1,2;2,1]; b = [2;2;3];
[x,fval] = quadprog(h,c,a,b,[],[],[0;0],[inf;inf])
```

```
x =

   0.66667
   1.33333

fval = -8.2222
```

In [ ]:
```

```

无约束非线性优化可用的函数主要是fminsearch()和fminunc()

In [15]:
```
fminsearch
```

```
error: Invalid call to fminsearch.  Correct usage is:

 -- X = fminsearch (FUN, X0)
 -- X = fminsearch (FUN, X0, OPTIONS)
 -- [X, FVAL] = fminsearch (...)

Additional help for built-in functions and operators is
available in the online version of the manual.  Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at http://www.octave.org and via the help@octave.org
mailing list.
```

In [ ]:
```

```

In [16]:
```
fminunc
```

```
error: Invalid call to fminunc.  Correct usage is:

 -- fminunc (FCN, X0)
 -- fminunc (FCN, X0, OPTIONS)
 -- [X, FVAL, INFO, OUTPUT, GRAD, HESS] = fminunc (FCN, ...)

Additional help for built-in functions and operators is
available in the online version of the manual.  Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at http://www.octave.org and via the help@octave.org
mailing list.
```

例3:

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

最后，非线性约束优化是最广泛的数值优化问题

$$\begin{aligned} min \quad & F(X), \\ s.t. & A*X <= B, Aeq*X = Beq, \\ & C(X) <= 0, Ceq(X) = 0, \\ & LB <= X <= UB. \end{aligned}$$

方法不一而足。matlab提供了fmincon()来完成这个任务，完整调用格式(参考help fmincon中的例子):

$$x = fmincon(FUN, X0, A, B, Aeq, Beq, LB, UB, NONLCON, OPTIONS)$$

然而值得指出的是：但是在octave的optim包中，目前尚未实现！

In [17]:
```
fmincon
```

```
warning: the 'fmincon' function belongs to the optim package from Octave Forge
but has not yet been implemented.

Please read <http://www.octave.org/missing.html> to learn how you can
contribute missing functionality.
error: 'fmincon' undefined near line 1 column 1
```

In [ ]:
```

```

稀疏矩阵

In [30]:
```
clear;
```

In [31]:
```
load m/sparse7585   % try other larger sparse matrix
```

In [32]:
```
whos
```

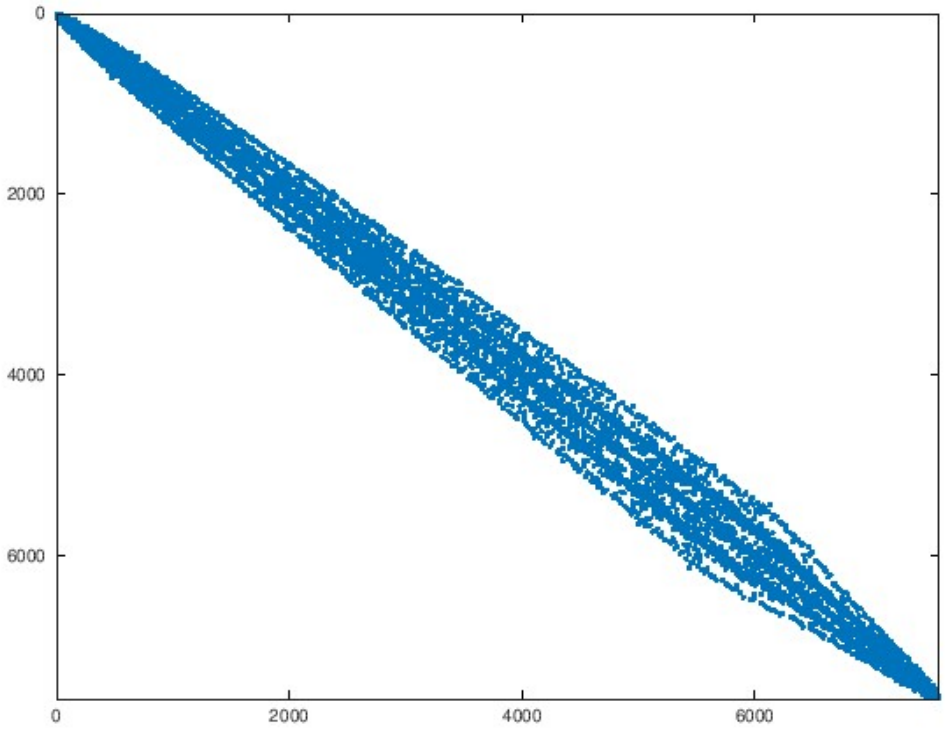
```
Variables in the current scope:

   Attr Name        Size                     Bytes  Class
   ==== ====        ====                     =====  =====
        A        7585x7585                 869280  double
        b        7585x1                     60680  double

Total is 57539810 elements using 929960 bytes
```

In [36]:
```
tic; x = A\b; toc
```

```
Elapsed time is 0.240335 seconds.
```

In [ ]:
```

```

In [33]:
```
spy(A)
```



In [39]:
```
condest(A)
```

```
ans =  4057.4
```

In [37]:
```
A_full = full(A);
```

In [38]:
```
whos
```

```
Variables in the current scope:

   Attr Name        Size                     Bytes  Class
   ==== ====        ====                     =====  =====
        A           7585x7585                869280  double
        A_full      7585x7585              460257800  double
        b           7585x1                     60680  double
        x           7585x1                     60680  double

Total is 115079620 elements using 461248440 bytes
```

In [10]:
```
tic; x_full = A_full\full(b); toc;
```

```
Elapsed time is 43.8837 seconds.
```

练习： 请对比求解不同规模稀疏矩阵计算（求条件数、矩阵分解、求线性方程组、求拟等操作），并列出你的数值结果（慎用大规模矩阵，注意内存。 如果要测试满矩阵算法，建议自己构造，规模不要大于10，000阶），参考PPT2最后一页

练习：RageRank算法实践与调研。 参考文献：

- PageRank算法 -- 从原理到实现 (https://blog.csdn.net/u013007900/article/details/88961913?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.nonecase)
- Mathworks帮助中心：使用 PageRank 算法对网站进行排名 (https://ww2.mathworks.cn/help/matlab/math/use-page-rank-algorithm-to-rank-websites.html;jsessionid=44865e9c12ba8d0247fe0cddf117)
- PageRank简单的迭代实现MATLAB (https://www.jianshu.com/p/16162a0058cf)

In [ ]:

迭代分形系统

Mandelbrot集：

$$z_{n+1} = z_n^2 + c$$

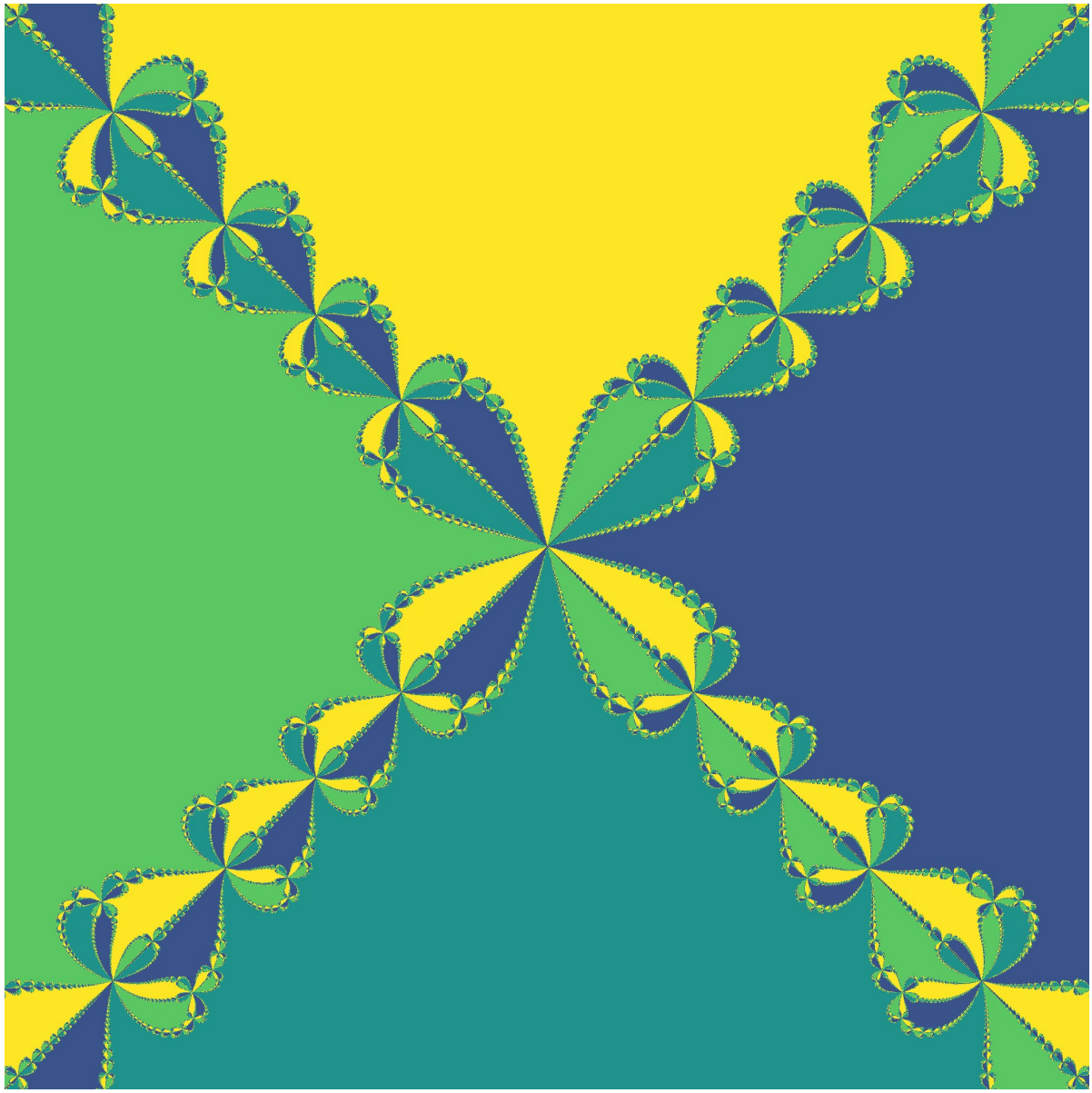
其中，c是一个复数。加入给定一个复数z_0，比如$z_0 = 0$，那么这个递推式会生成一个序列：$[z0, z1, z2, z3, \ldots]$:

- 如果这个序列收敛在一个有限值范围内，那么c就在Mandelbrot集合中
- 当序列中有一项的绝对值大于2时，这个序列一定趋向于无穷，一般说来，如果这个序列前60项都没有一项大于2，那么可以认为这时c就在Mandelbrot集合中!

我的实现

In [12]:

```
%
%    显示多项式p^4-1=0
%    的吸引盆以及Julia集
%
%

t = -2:0.05:2;
[x,y] = meshgrid(t);
z0 = x + i*y;
z1 = z0 - 0.25*(z0.^4-1)./z0.^3;
k = 1;
err = abs(z1-z0);
while k < 100 && norm(err,1) > 0.0000001
    z0 = z1;
    z1 = z0 - 0.25*(z0.^4-1)./z0.^3;
    err = abs(z1-z0);
    k = k + 1
end
idx1 = find(abs(z1-1)<0.01);
idx2 = find(abs(z1-i)<0.01);
idx3 = find(abs(z1+1)<0.01);
idx4 = find(abs(z1+i)<0.01);

[m,n] = size(x);
flag = zeros(m,n);
flag(idx1) = 1;
flag(idx2) = 2;
flag(idx3) = 3;
flag(idx4) = 4;

imagesc(flag);
```



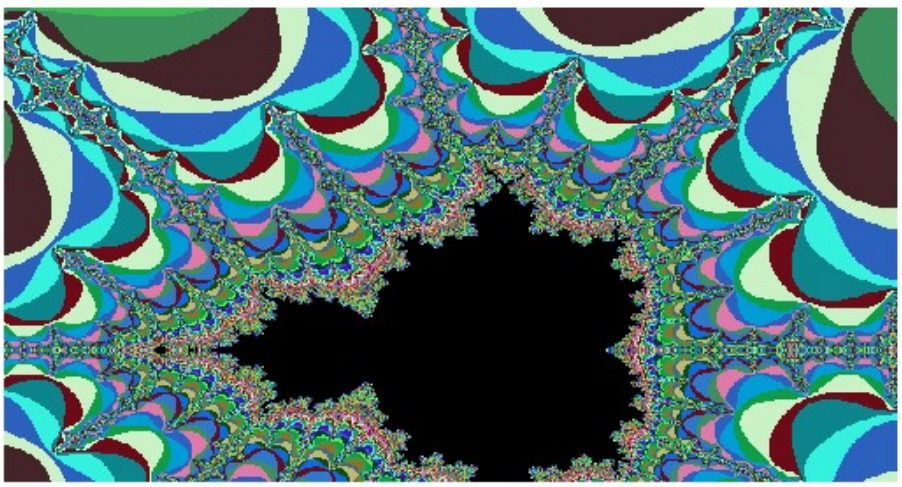
另一个实现

In [2]:
```
xc = -1.478;   %图片中心点
yc = 0;
zoom = 200;    %放大倍数
res = 512;    %分辨率

% 在 (-1.478,0) , 放大300倍,    xc=-1.478,zoom=300,

x0 = xc - 2 / zoom;
x1 = xc + 2 / zoom;
y0 = yc - 2 / zoom;
y1 = yc + 2 / zoom;

x = linspace(x0, x1, res);
y = linspace(y0, y1, res);
[xx, yy] = meshgrid(x, y);
```

In [11]:
```
```

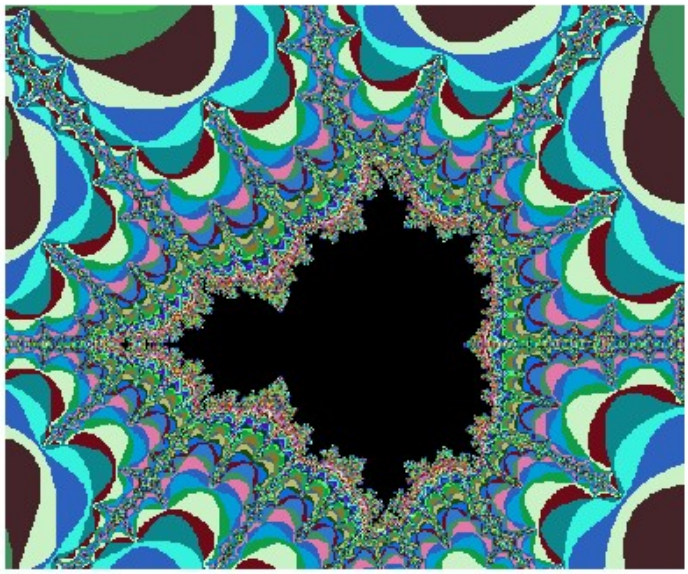
In [3]:
```
iter=100;
C = xx + yy * 1i;
z = zeros(size(C));
N = uint8(zeros(res, res, 3));
color = uint8(round(rand(iter, 3) * 255));
for k = 1: iter

    z = z.^2 + C;    %% 主要执行的迭代计算

    %% 分类点[x0,x1]x[y0,y1]
    [row, col] = find(abs(z) > 2);     % 每次迭代都做判断，为了让不收敛的数值控制在有效范围内，避免出现Nan
    k1 = zeros(size(row)) + 1;
    k2 = zeros(size(row)) + 2;
    k3 = zeros(size(row)) + 3;

    p1 = sub2ind(size(N), row, col, k1);     N(p1) = color(k, 1);
    p2 = sub2ind(size(N), row, col, k2);     N(p2) = color(k, 2);
    p3 = sub2ind(size(N), row, col, k3);     N(p3) = color(k, 3);


    z(abs(z) > 2) = 0;
    C(abs(z) > 2) = 0;
end
```

In [8]:
```
imagesc(N);
% print -dpng 'test2B.png'
```



In [9]:
```
imwrite(N, 'test2.png');
```

In [11]:
```
imshow(N); % print -dpng 'test2B.png'
```



In [ ]:
```
```

Julia集

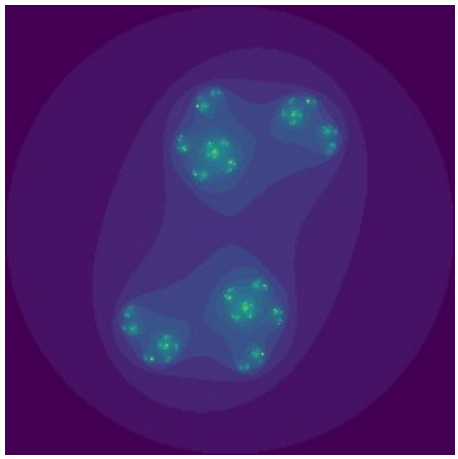
In [23]:
```
c = 0.6 + 0.5i;
```

In [28]:
```
res = 256;
r = max(abs(c),2);  % radius of the circle beyond which every point diverges
t = linspace(-r,r,res); % divide the x-axis

A = ones(res,1)*t + i*(ones(res,1)* t)'; % create the matrix A containing complex n
umbers
B = zeros(res,res); % create the point matrix
```

In [34]:
```
tic;
iter = 50;
for s = 1:iter
    B = B + (abs(A)<=r);
    % the map
    A = A.*A+ones(v,v).*c;
end
toc;
```

```
Elapsed time is 0.282307 seconds.
```

In [35]:
```
imagesc(B);
```



In [ ]:

练习

- 请展示一个任取一中心，放大倍数为100，在5120x5120分辨率，序列项数为10000的Manderbolt集
- $C = 0.2 + 0.65i$; 分辨率为512、放大倍数10的Julia集的图形（序列项数k=30）？

调研与实践：通过查阅更多资料，谈谈你对M集和J集区别与联系的理解，可否找到更有代表性的参数C，which 对应于更典型的集合图像（Matlab下GUI的代码案例，参考附件中的fractral文件夹）。

In [ ]: