2017 年数学建模国赛 A 题解析

数模国赛临门一脚冲刺课程

周吕文

超级数学建模大俱乐部

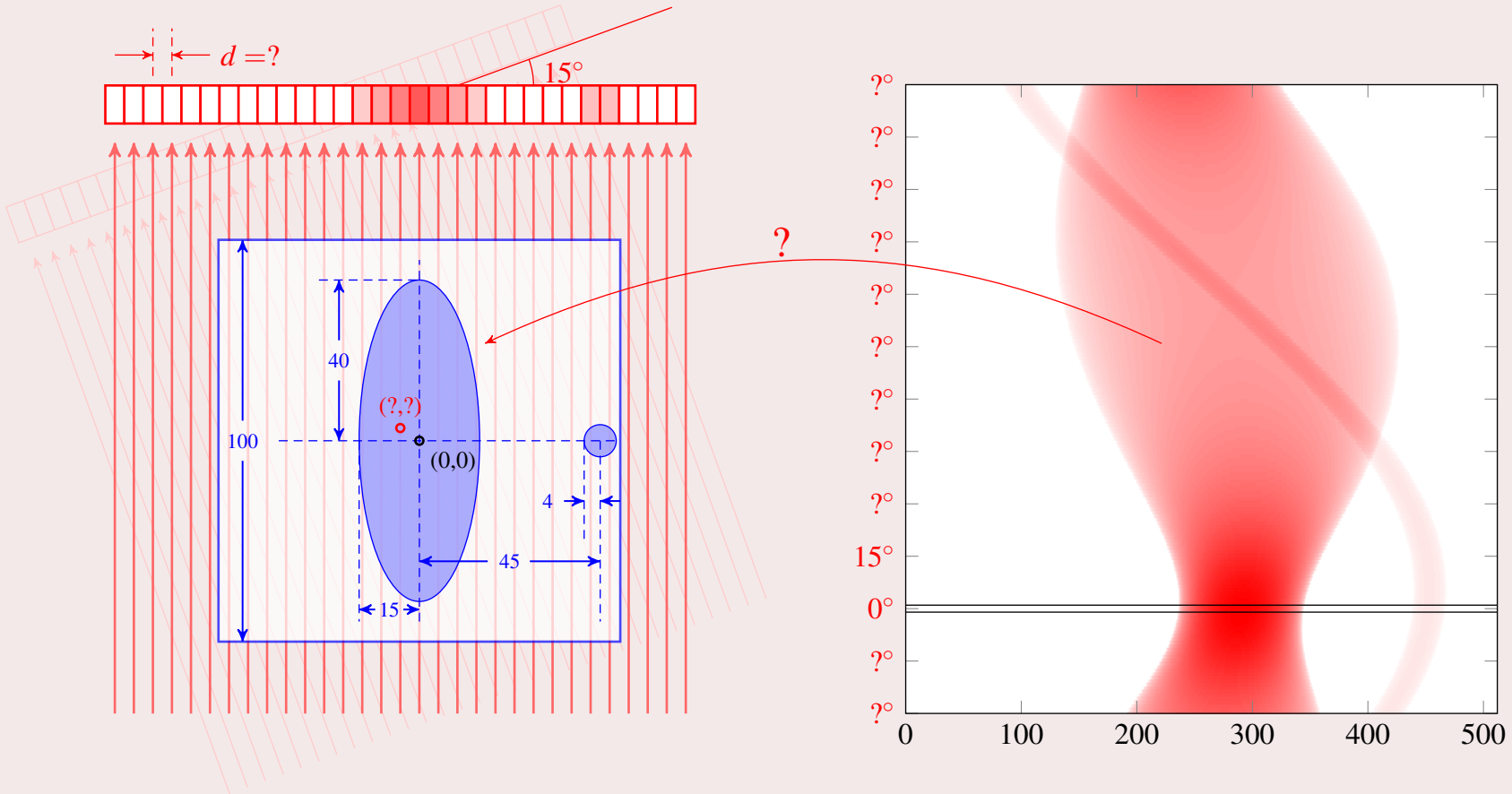
2018 年 9 月 8 日

招码听课

超级数模

提要

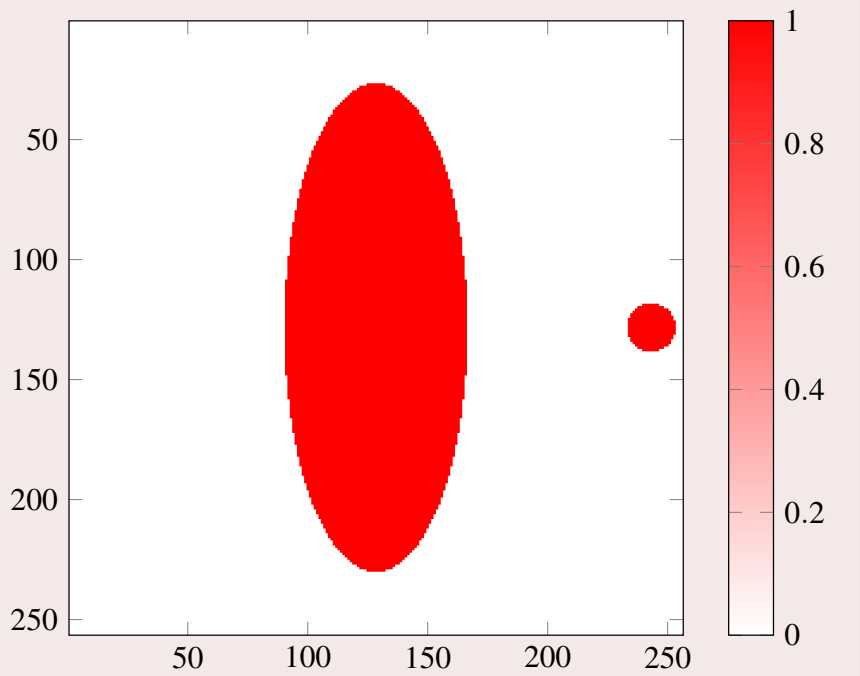
2017 CUMCM A 题：CT 系统参数标定及成像



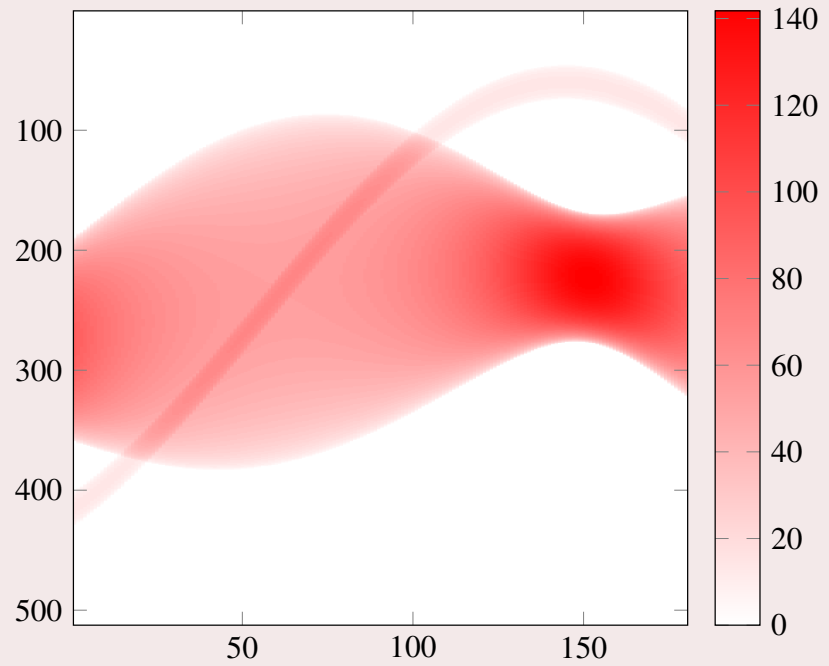
- 由模板及接收信息，确定旋转中心、探测器间距及射线方向
- 由未知介质的接收信息，确定其位置、形状和吸收率等信息
- 分析参数标定的精度和稳定性，在此基础上自行设计新模板

附件 1 & 2

附件 1：256 × 256 数组



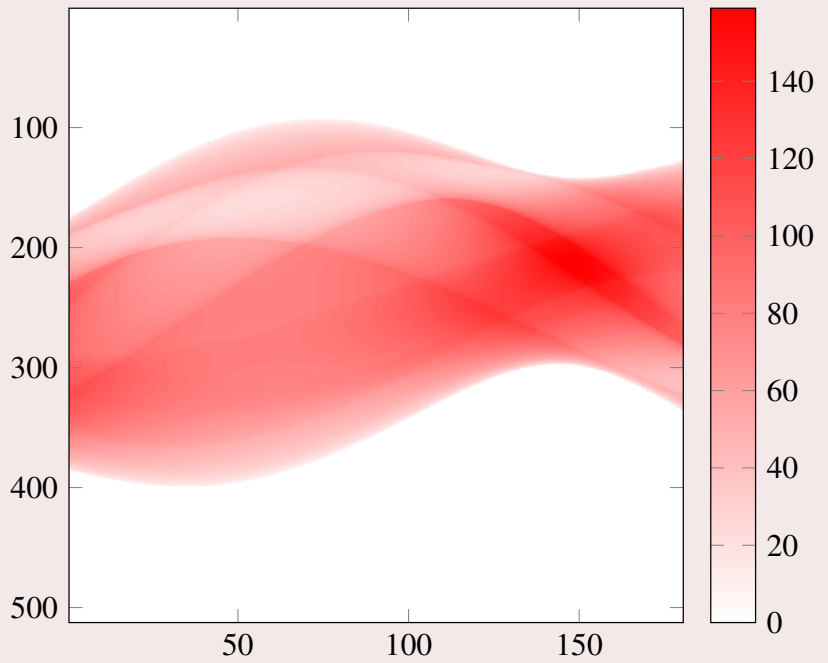
附件 2：512 × 180 数组



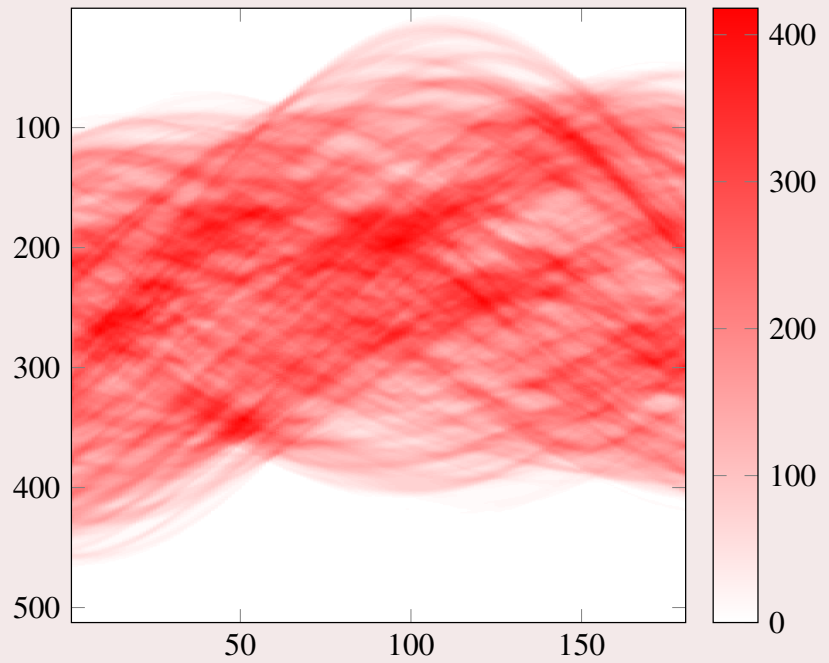
- 确定 CT 旋转中心、探测器单元间距及该 X 射线的 180 个方向。

附件 3 & 5

附件 3：512×180 数组

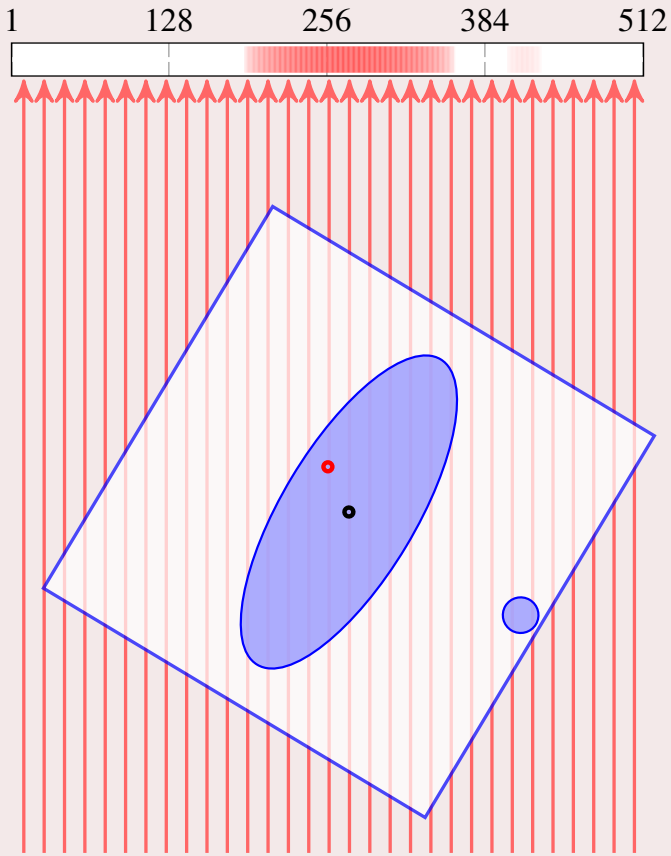


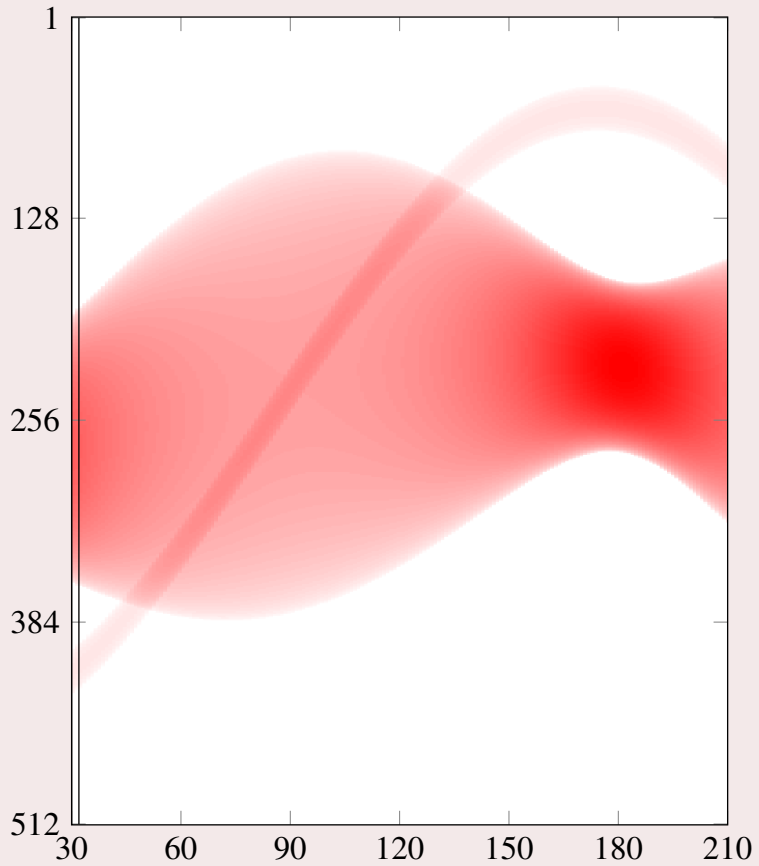
附件 5：512×180 数组



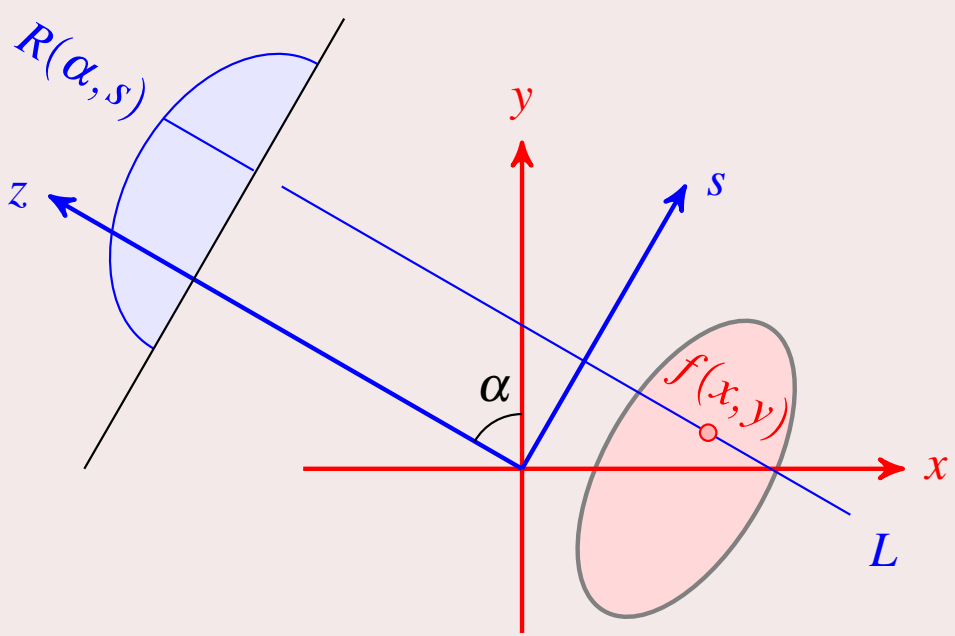
- 确定介质在托盘中的位置、几何形状和指定点的吸收率。

CT 成像系统原理: 投影过程





CT 成像系统原理: Radon 变换



沿线积分

$$R(L) = \int_L f(\mathbf{x})|d\mathbf{x}|$$

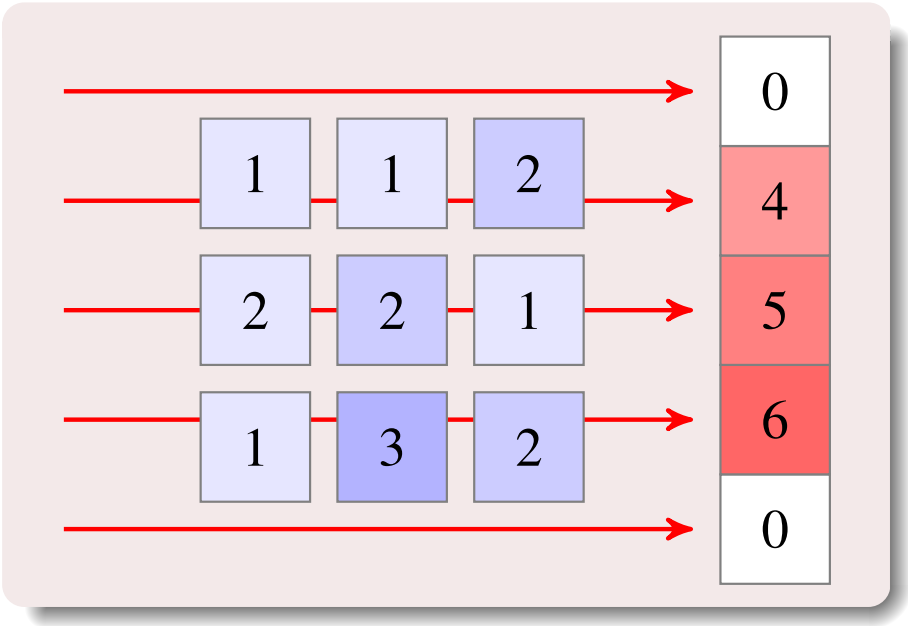
坐标系旋转变换

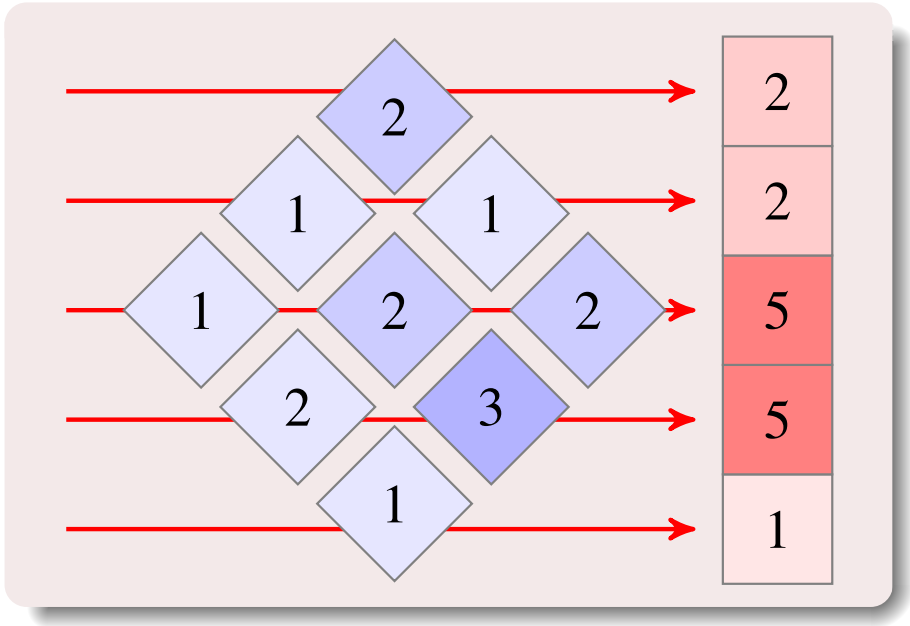
$$\begin{aligned} x &= \quad z\sin\alpha + s\cos\alpha \\ y &= -z\cos\alpha + s\sin\alpha \end{aligned}$$

Radon 变换

$$R(\alpha, s) = \int_{-\infty}^{\infty} f\big(z\sin\alpha + s\cos\alpha, -z\cos\alpha + s\sin\alpha\big)\, dz$$

CT 成像系统原理: Radon 变换

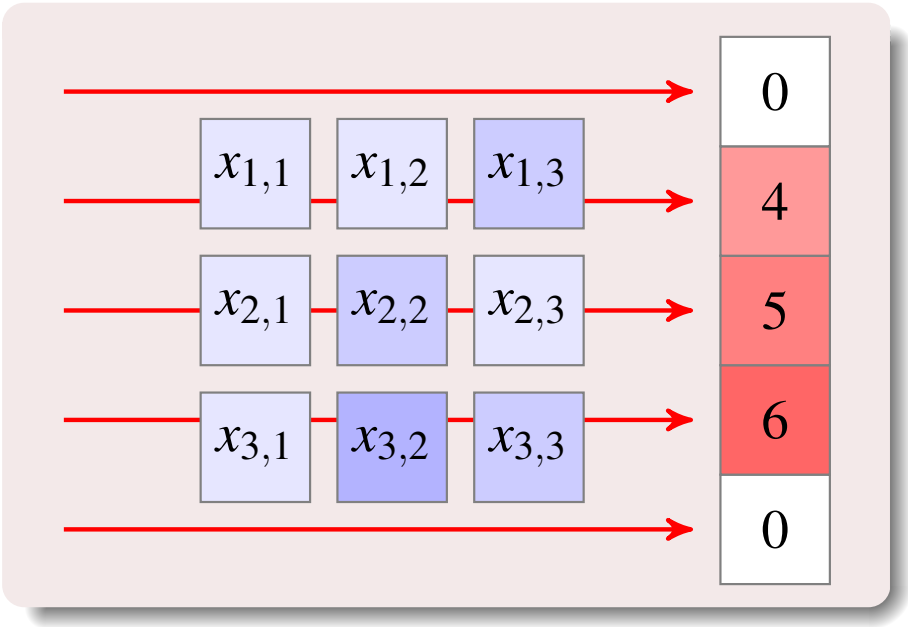


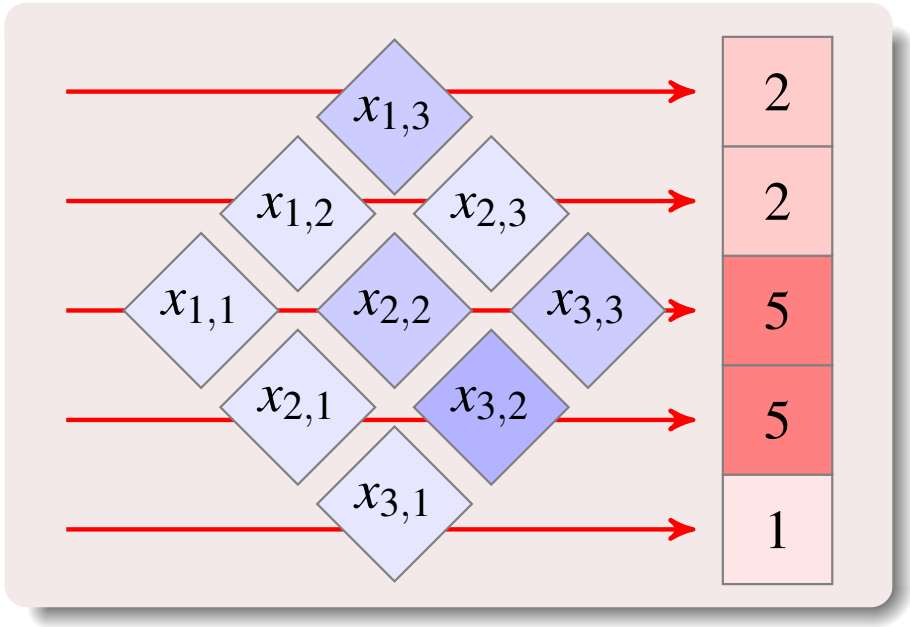


Radon 变换的相关参考

- Radon transform — Wikipedia, The Free Encyclopedia
- CT (Computed Tomography) Scans – A Level Physics

CT 成像系统原理: 反 Radon 变换

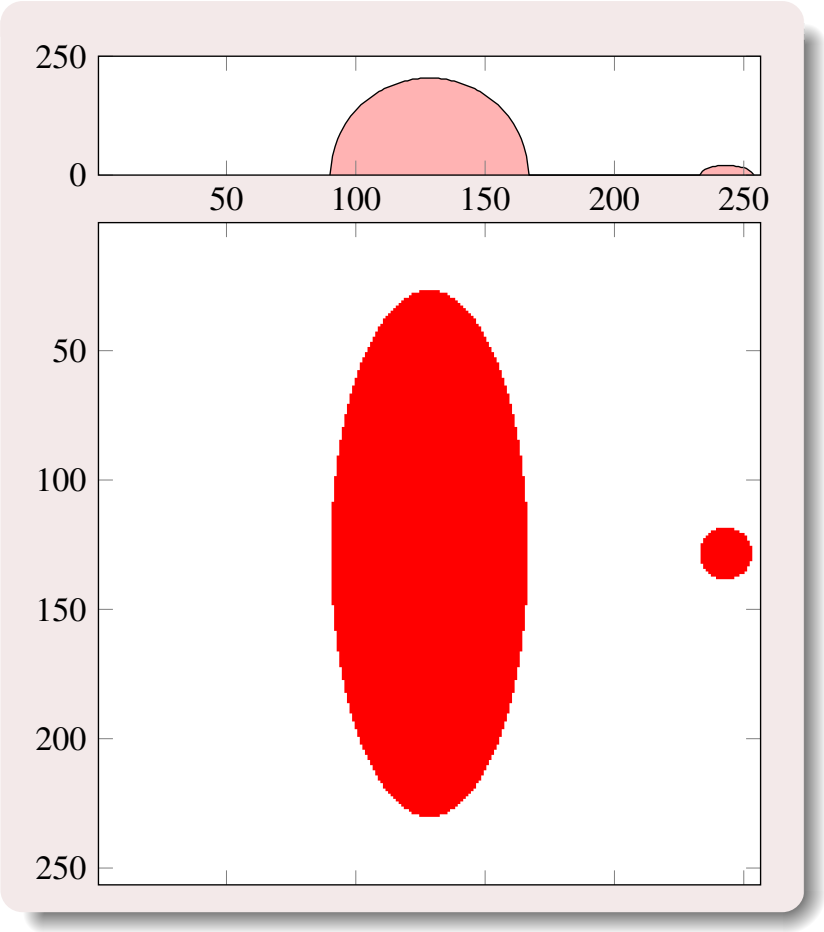


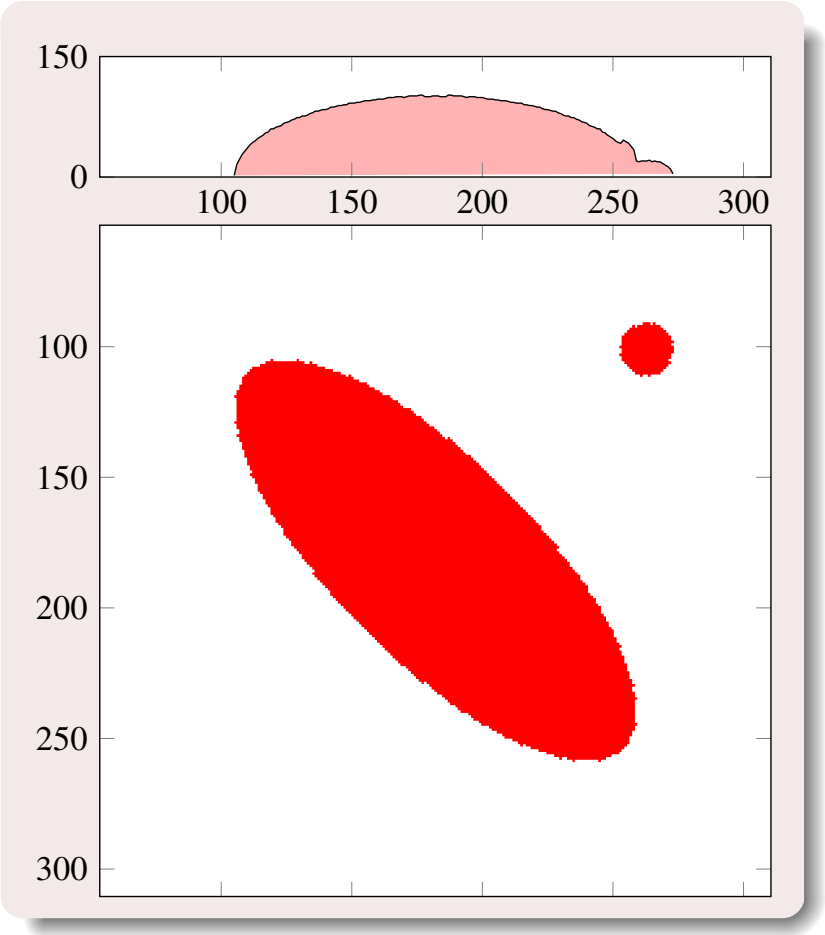


反 Radon 变换

- 已经投影，反推 $\mathbf{x}$

CT 成像系统原理: Radon 变换

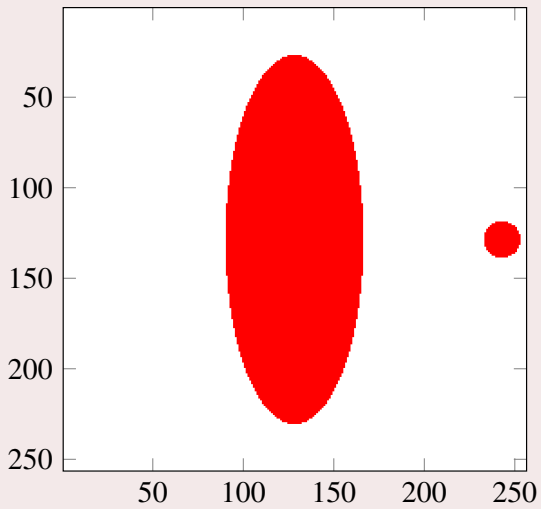


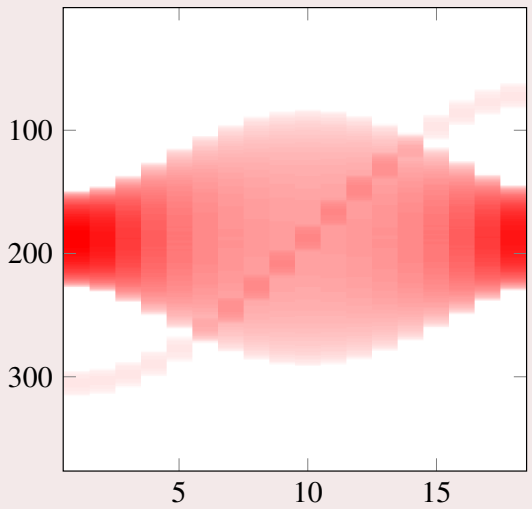


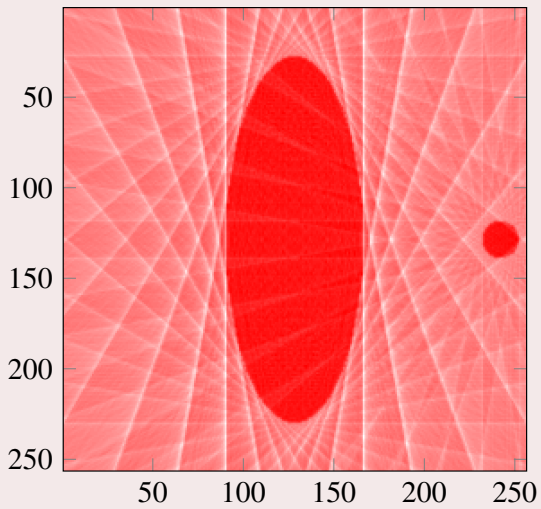
图的旋转变换

```
01 p0 =  load('1.dat'); plot(sum(p0))
02 p1 = imrotate(p,45); plot(sum(p1))
```

CT 成像系统原理: Radon 变换与反变换







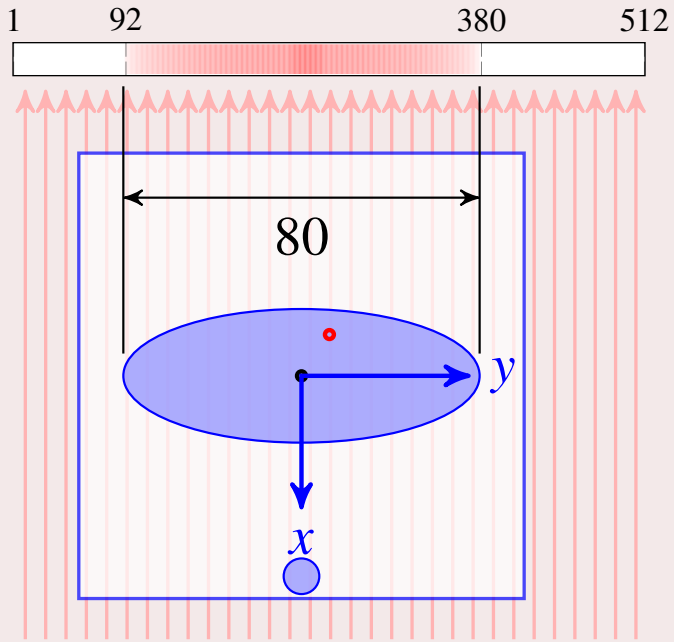
radon & iradon

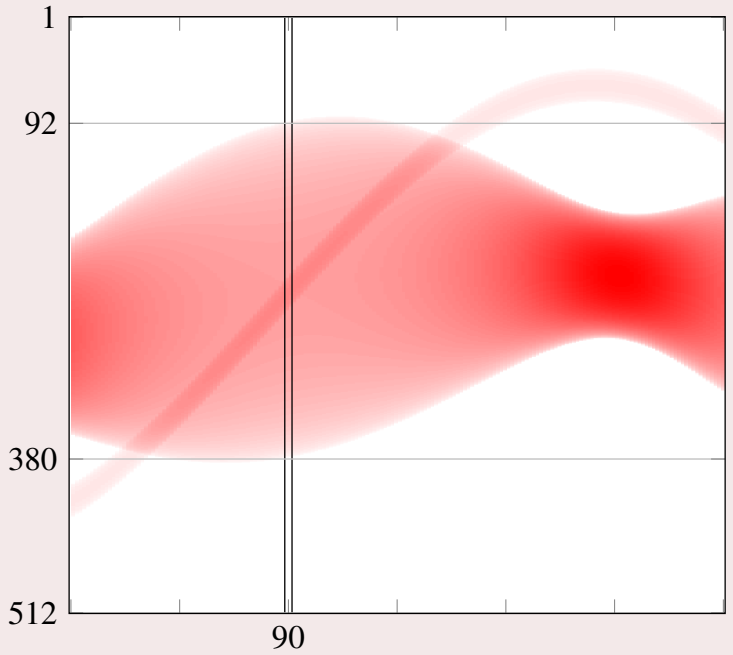
```
01 subplot(1,3,1); P = load('1.dat');          imagesc(P);
02 subplot(1,3,2); R =  radon(P, 0:10:179); imagesc(R);
03 subplot(1,3,3); I = iradon(R, 0:10:179); imagesc(I);
```

提要

问题一：探测器单元间距

$$d = L/N = 80/(380 - 92 + 1) = 0.2768$$

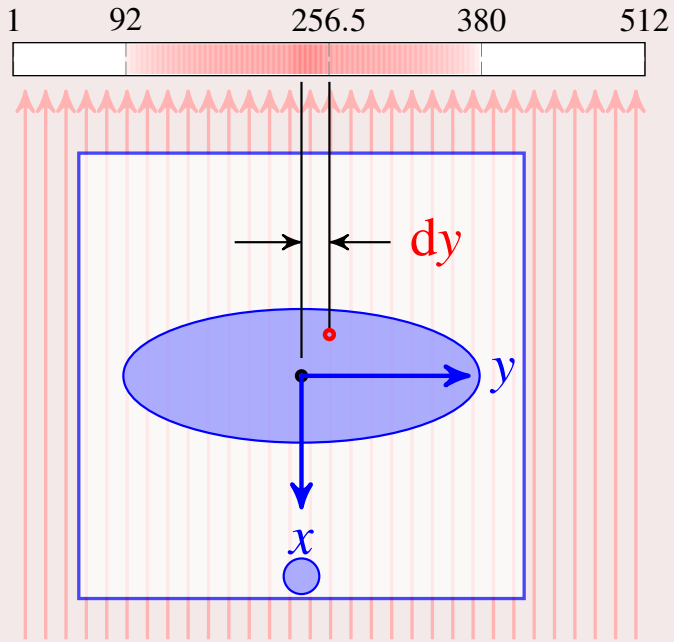


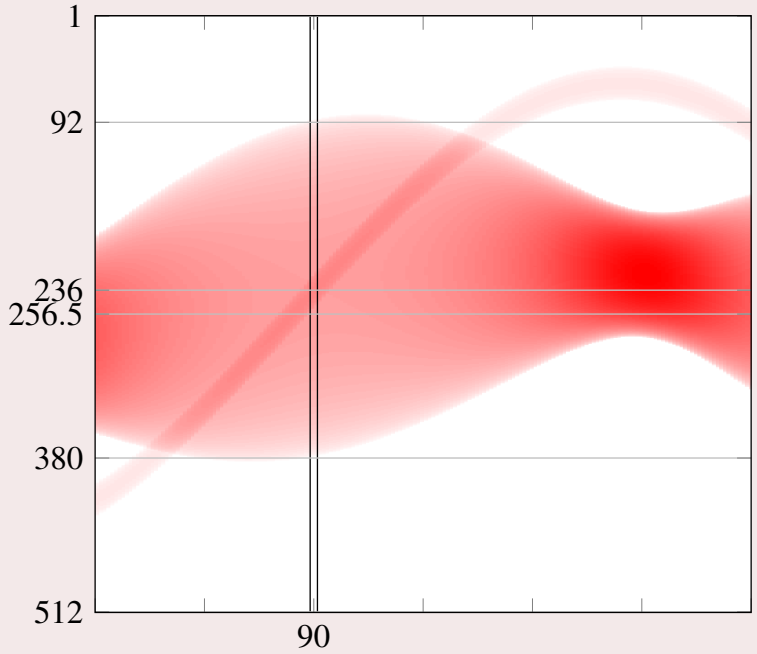



```
01 proj = load('2.dat');
02 width = sum(proj>0);
03 d = 80/max(width);
```

问题一：系统旋转中心位置

$\mathrm{d}y = (256.5 - 236)d$

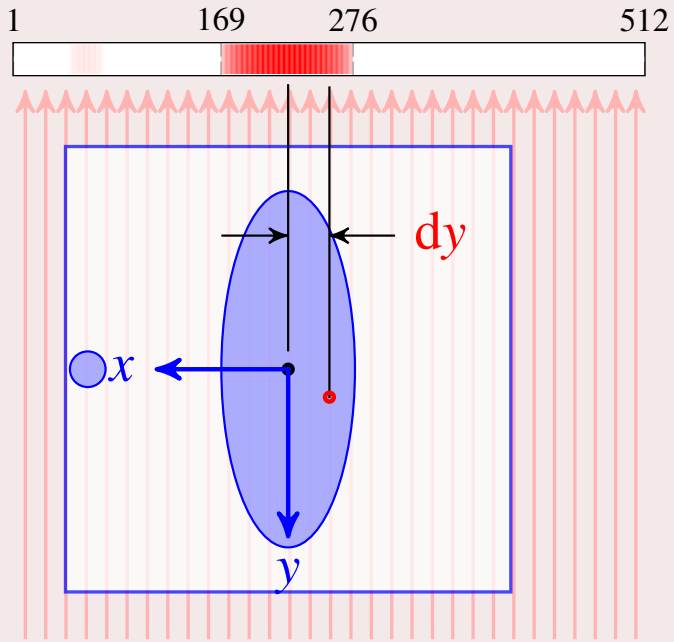


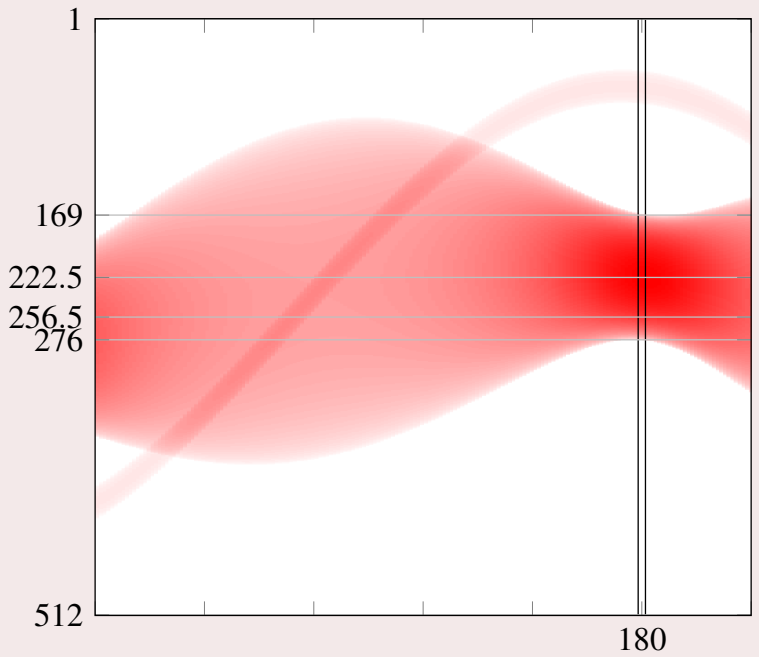



```
01 [wmax, imax] = max(width);
02 idx = find(proj(:,imax)>0);
03 yc = ( 256.5 - (max(idx)+min(idx))/2 ) * d;
```

问题一：系统旋转中心位置

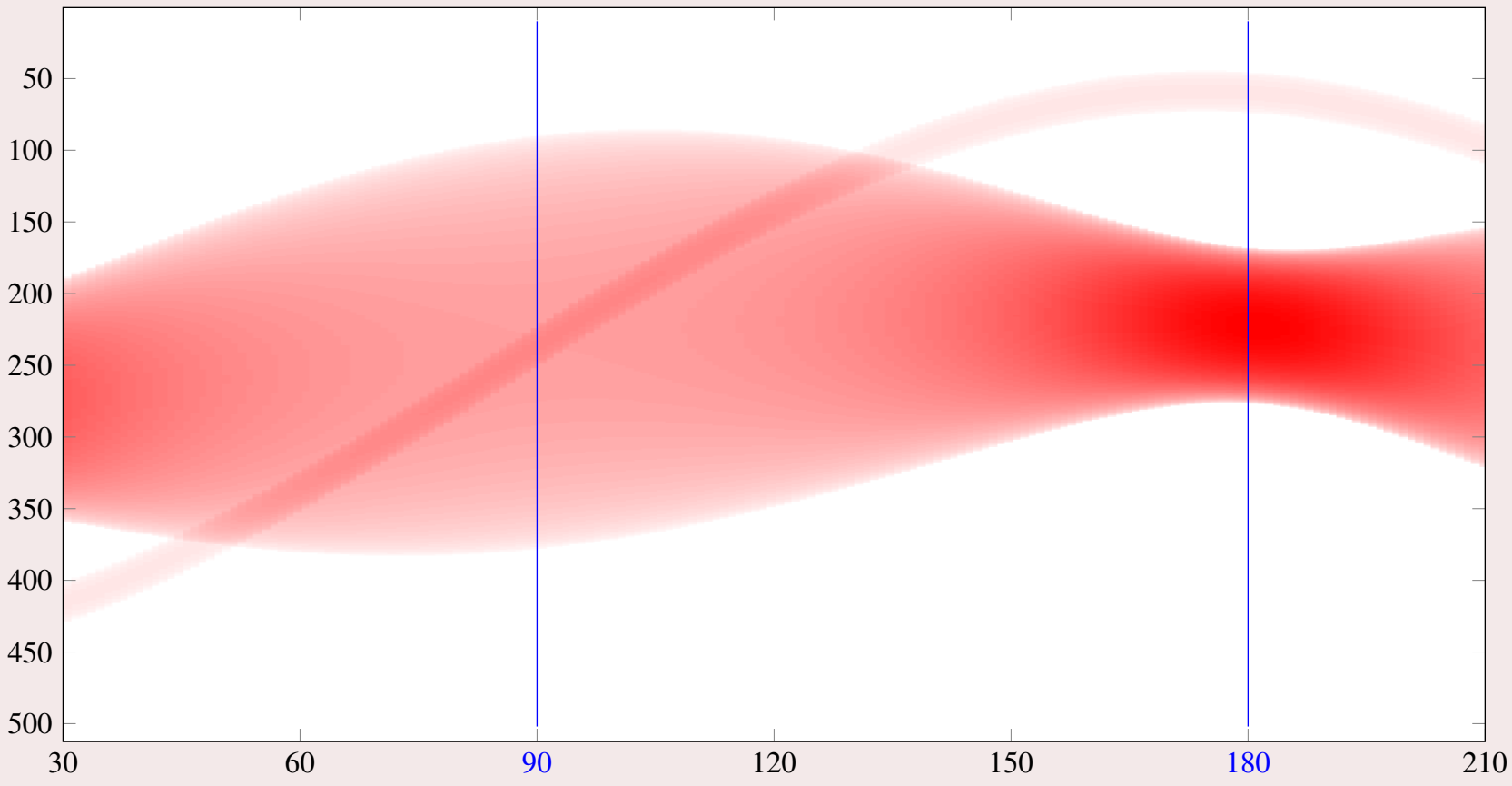
$$\mathrm{d}x = (256.5 - 222.5)d$$





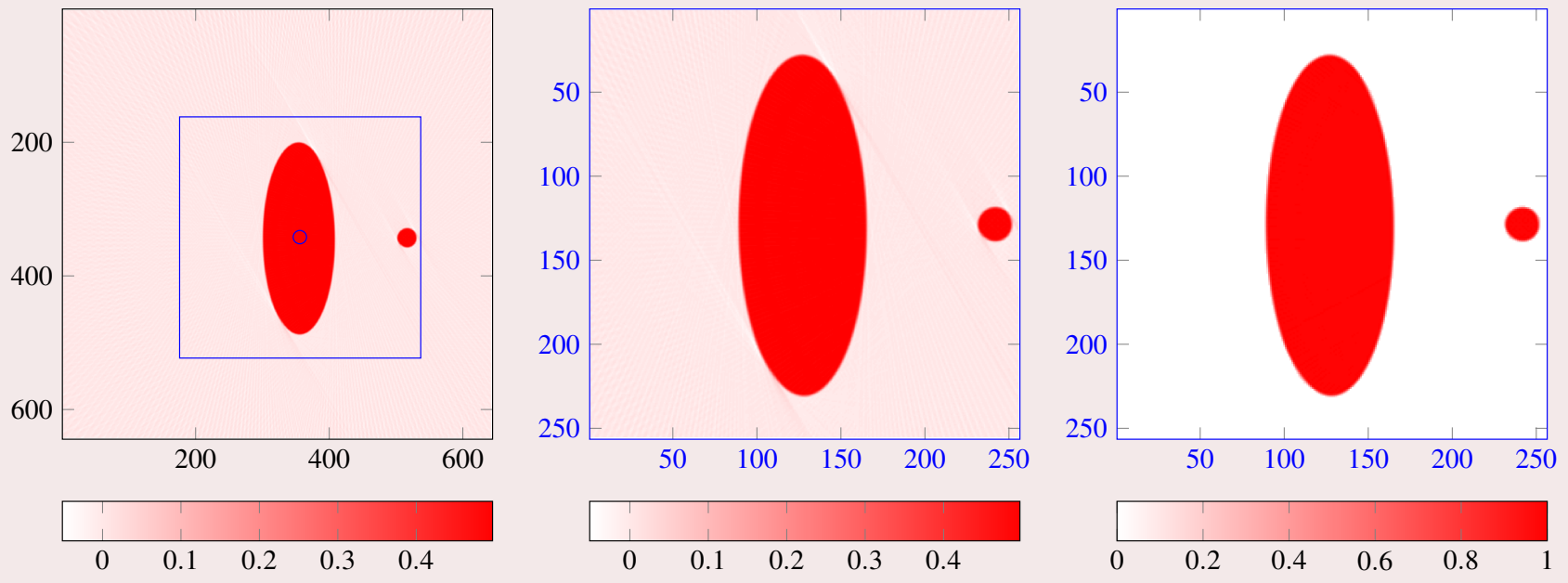

```
01 [wmin, imin] = min(width);
02 idx = find(proj(:,imin)>0); idx = idx(idx>100);
03 xc = - ( 256.5 - (max(idx)+min(idx))/2 ) * d;
```

问题一：X 射线的 180 个方向



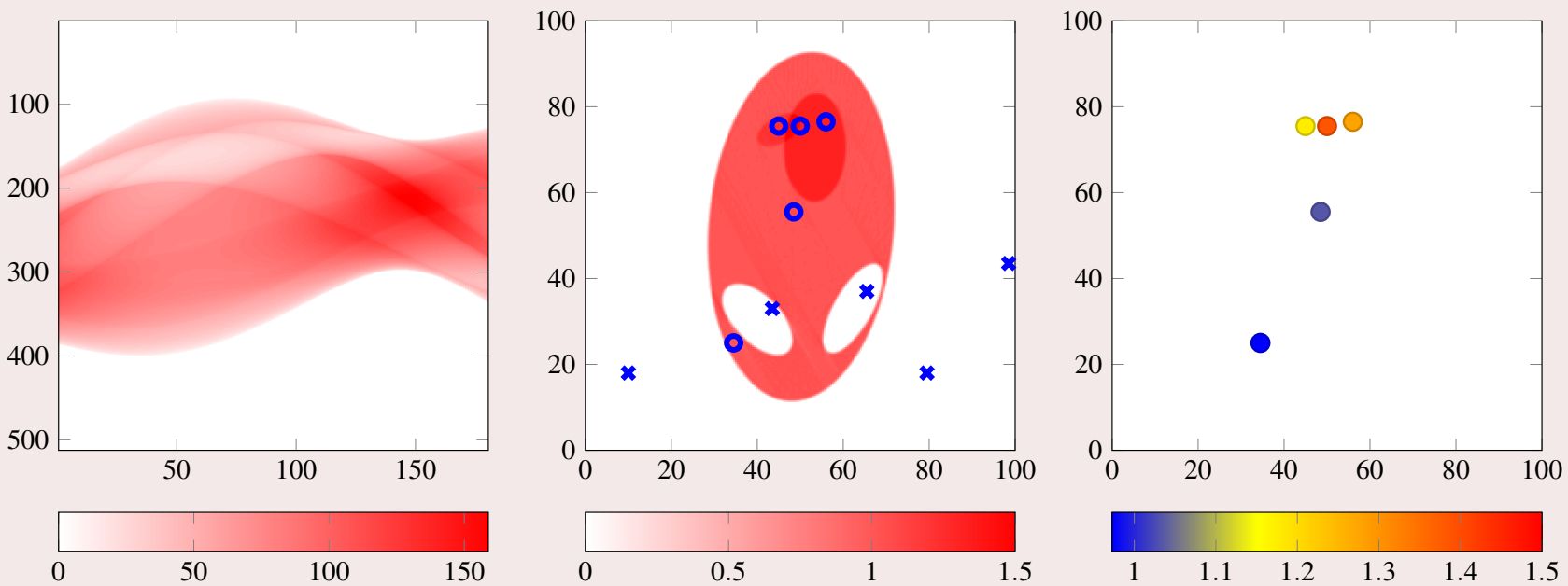

```
01 phi = 180 - imin; % 90 - imax
```

问题一：验正

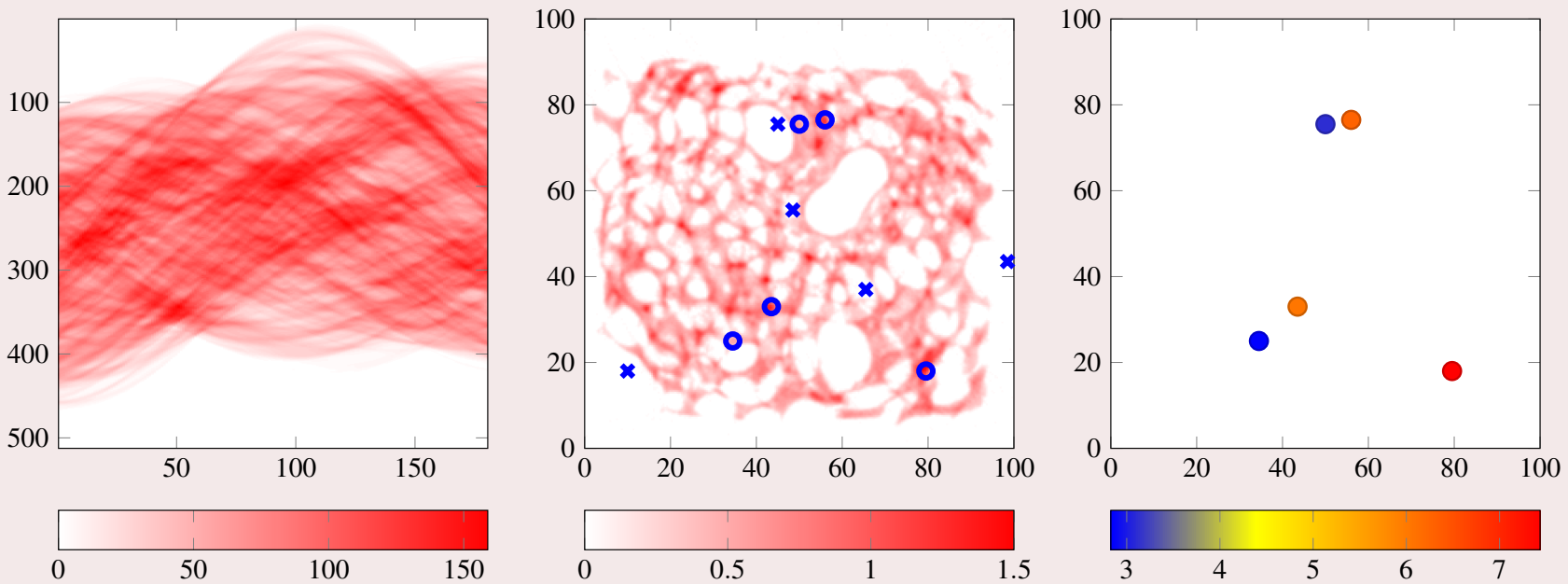



```
01 proj = [zeros(200,180); proj; zeros(200,180)];
02 img = iradon(proj, [0:179]+phi, 'Hann'); imagesc(img);
03 nhalf = size(img,1)/2;
04 xidx = ceil(-xc/d) + nhalf + [ceil(-50/d):ceil(50/d)];
05 yidx = ceil( yc/d) + nhalf + [ceil(-50/d):ceil(50/d)];
06 img = img(yidx,xidx);                          imagesc(img);
07 img = imresize(img, [256, 256])*prod(size(img))/256^2;
08 img(img<1e-1) = 0;                            imagesc(img);
```

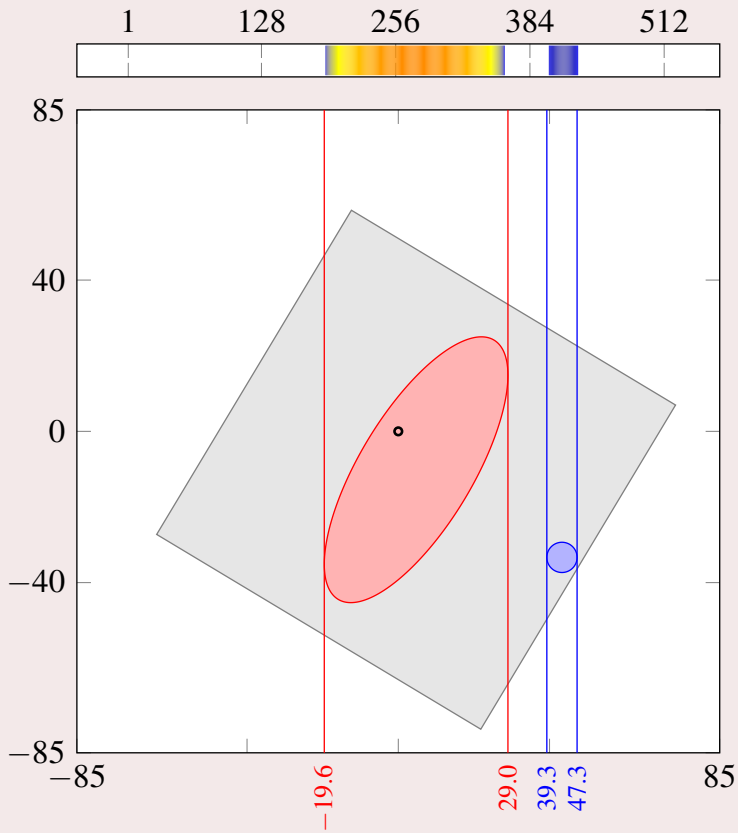
问题二

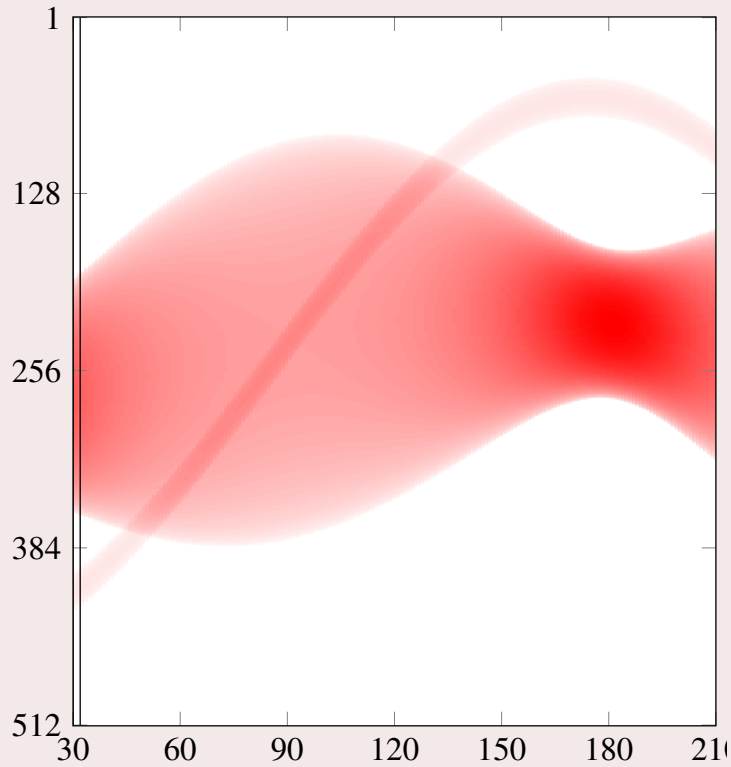

100
200
300
400
500
50
100
150
0
50
100
150
100
80
60
40
20
0
0
20
40
60
80
100
0
0.5
1
1.5
100
80
60
40
20
0
0
20
40
60
80
100
1
1.1
1.2
1.3
1.4
1.5

问题三


100
200
300
400
500
50
100
150
0
50
100
150
100
80
60
40
20
0
0
20
40
60
80
100
0
0.5
1
1.5
100
80
60
40
20
0
0
20
40
60
80
100
3
4
5
6
7

椭圆的投影


1
128
256
384
512
85
40
0
−40
−85
−85
85
−19.6
29.0
39.3
47.3


1
128
256
384
512
30
60
90
120
150
180

椭圆的投影

椭圆方程

$$\frac{(x\cos\theta - y\sin\theta)^2}{a^2} + \frac{(x\sin\theta + y\cos\theta)^2}{b^2} = 1$$

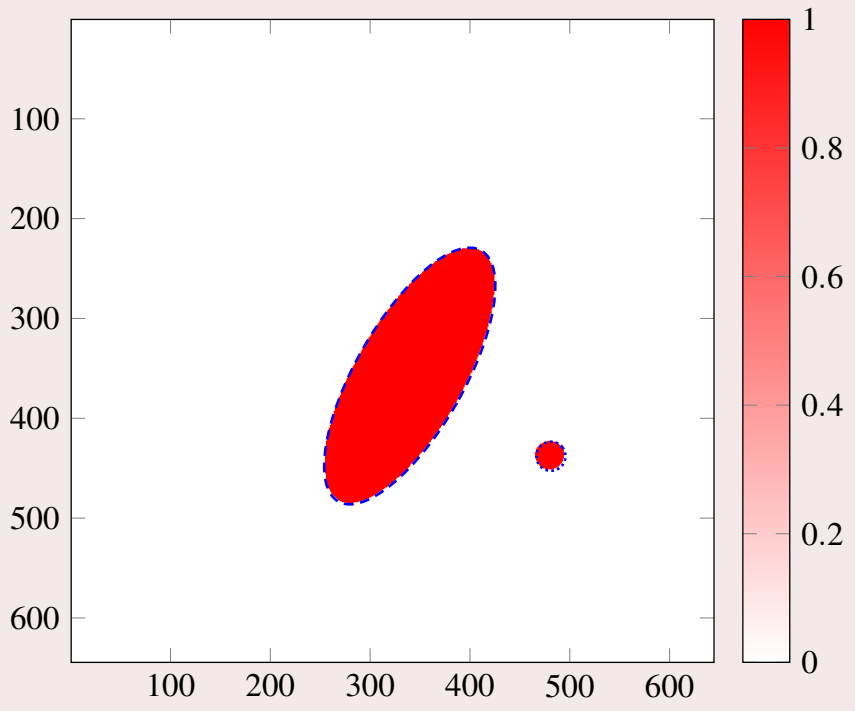
导数

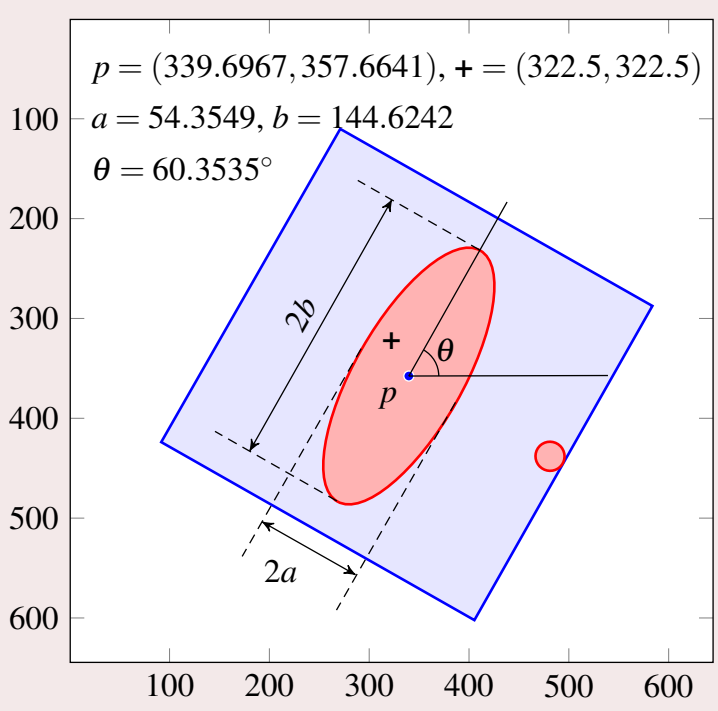
$$\frac{\mathrm{d}y}{\mathrm{d}x} = -\frac{a^2 x\sin^{\theta} + y(a^{-}b^2)\sin\theta\cos\theta + b^2 x\cos^2\theta}{a^2 y\cos^{\theta} + x(a^{-}b^2)\sin\theta\cos\theta + b^2 y\sin^2\theta}$$

椭圆投影边界

$$x = \pm\sqrt{a^2\cos^{\theta} + b^2\sin^2\theta}, \qquad y = \pm\sqrt{a^2\sin^{\theta} + b^2\cos^2\theta}$$

椭圆的拟合






```
01 proj = [zeros(200,180); load('2.dat'); zeros(200,180)];
02 img = iradon(proj,[0:179]); bw = im2bw(img,0.2); imagesc(bw);
03 obj = regionprops(bw, 'Centroid', 'Orientation',...
04                 'MajorAxisLength','MinorAxisLength' );
05 p = obj(1).Centroid;        theta = obj(1).Orientation;
06 a = obj(1).MinorAxisLength/2; b = obj(1).MajorAxisLength/2;
```

主程序 & 计算参数

main.m

```
01 [phi, d, xc, yc] = getparm;
02
03 proj = load('3.dat'); % proj = load('5.dat');
04 [xyrate, img] = proj2img(proj, phi, d, xc, yc);
```

proj2img.m

```
01 function [phi, d, xc, yc] = getparm
02 phantom = load('data/1.dat'); proj = load('data/2.dat');
03 width = sum(proj>0);
04 [wmax, imax] = max(width); [wmin, imin] = min(width);
05 phi = 180-imin; d = 80/wmax;
06 idy = find(proj(:,imax)>0);
07 yc =  (256.5-(max(idy)+min(idy))/2 )*d;
08 idx = find(proj(:,imin)>0); idx = idx(idx>100);
09 xc = -(256.5-(max(idx)+min(idx))/2 )*d;
```

投影恢复为图像

proj2img.m

```
01 function [xyrate, img] = proj2img(proj, phi, d, xc, yc)
02 proj = [zeros(200,180); proj; zeros(200,180)];
03 img = iradon(proj,[0:179]+phi,'Hann');
04 xidx = ceil(-xc/d) + size(img,1)/2 + [ceil(-50/d):ceil(50/d)];
05 yidx = ceil( yc/d) + size(img,1)/2 + [ceil(-50/d):ceil(50/d)];
06 img = imrescale(img(yidx,xidx), [256,256]);
07
08 imgud = flipud(img); xy = load('4.dat'); d = 100/256;
09 ic = ceil(xy(:,1)/d); ir = ceil(xy(:,2)/d);
10 idx = sub2ind(size(imgud), ir, ic); xyrate = [xy imgud(idx)];
11
12 imagesc([0,100],[0,100],imgud); colorbar; hold on;
13 plot(xy(imgud(idx)==0,1), xy(imgud(idx)==0,2), 'wx')
14 plot(xy(imgud(idx)~=0,1), xy(imgud(idx)~=0,2), 'r+')
15 set(gca,'ydir','normal');
16
17 function new = imrescale(old, newsize)
18 scale = prod(size(old)) / prod(newsize);
19 new = imresize(old, newsize) * scale;  new(new<1e-1) = 0;
```

提要

评审

略

知识要点

检索

- 检索英文关键词，图片搜索，可加入“matlab”等关键词
- 例：“CT reconstruction matlab”，“Radon transform matlab”

模型

- 封闭二维几何曲线在坐标轴上投影
- 了解 CT 成像原理，及其背后的数学原理和模型

程序

- 坐标系旋转变换
- 简单图像处理、图像中的椭圆拟合
- radon 和 iradon 变换

作业

- 利用椭圆拟合的方法，重新求解第一问
- 求任意椭圆在 $y = x$ 直线上的投影长度
- 找一张照片，用 radon 变换为投影，再用 iradon 将投影还原为图像

Thank You!!!