

CT系统的参数标定及未知介质的图像重建

摘 要

本文运用滤波反投影和几何分析的方法研究CT系统，分析了CT系统的系统参数（旋转中心、探测器单元距离、X射线的180个方向，吸收强度），并在此参数标定后的模板上对未知介质在探测器上的投影矩阵进行反投影，实现图像重建，并由此获取未知介质的相关信息（在正方形托盘上的位置、几何形状和吸收率），和已知模板上点坐标的十个点的吸收率。

问题一是参数标定，在给定均匀介质的吸收强度和接受信息的条件下，标定系统的参数。本文运用了几何分析，首先，根据抽样原理，选择等距为 δ 的四条穿过小圆的射线，因为衰减程度和射线通过的距离即射线截小圆的弦长成正比，可以用`matlab`进行方程组求解，得到单元距离 δ 为 $0.2767mm$ ；再通过求解附件二的数据图中探测器上投影最长的和最短的列确定旋转中心的坐标，得到结果为 $(-9.2683, 5.9483)mm$ ；最后，已知旋转中心在探测器上的投影固定，根据通过旋转中心的射线和相切与小圆的射线的距离公式等于投影长度和 δ 的乘积，求解得到180个角度为 -60.1643 度到 -61.6703 度。

问题二是在问题一标定参数模板上的应用，给出未知介质的接受信息，求其相关信息，使用`iradon`函数进行滤波反向投影，实现图像重建，对结果进行去噪，得到未知介质在正方形托盘的位置、几何形状和吸收率。根据附件4中10组位置的数据建立以托盘中心为原点的平面直角坐标系，使得图像重建结果拟合良好。十个点的坐标在结果为 256×256 的矩阵中对应位置即其吸收强度。

问题三同样是已知模板中未知结构的样品的成像，计算过程和方法同问题二的分析一致。

问题四对于精度和稳定性分析，考虑到抽样（探测器间距计算）和数据离散化（旋转中心坐标和角度计算）产生的误差，通过改进抽样方法，使用插值拟合逼近连续而增大结果的精确度；改进模型，保持重建图像的几何形状和位置信息，减小重建图像中的误差，增强程序的稳定性；对新模板的两个参数检验可行性：角度，模板的吸收率。

关键词： 几何分析;滤波反投影;`iradon`函数;参数标定;图像重建

一 问题重述

CT是一种利用射线能量经过不同组织材料样品吸收后的进行断层成像，从而通过图像获得产品内部信息。且给出一个典型的CT系统的示意图如图1：

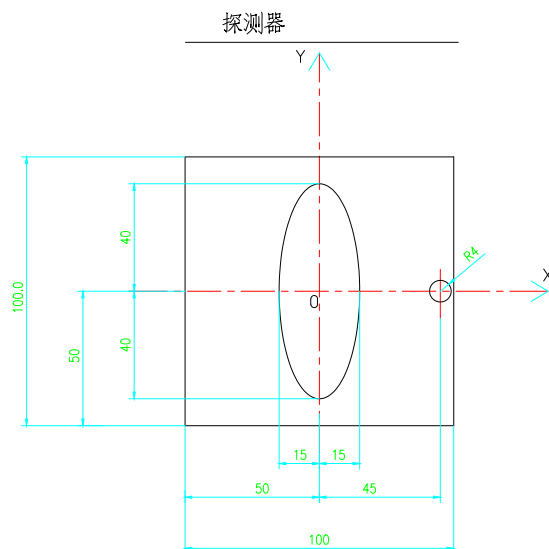


图 1: CT示意图

1.1 问题一

问题一：要求我们利用附件一与附件二的数据，附件一是模板的吸收率数据，附件二是探测器的接受强度的数据。利用两数据可以计算探测器的单元间距，探测器中心位置坐标以及给出的180个旋转角度。

1.2 问题二及问题三

问题二以及问题三：利用问题一所得结论，对应附件3及附件5中已得的信息，推导未知介质的位置，形状以及吸收率等问题，并给出图3 确定的10 个位置的吸收率。

1.3 问题四

基于问题一的精度以及稳定性，自行设计新的模板，改进稳定性和精确程度，并标定模型。

二 问题分析

2.1 问题一的分析与思路

针对问题一，分析附件一与附件二的数据，几何与代数相结合进行计算。求解探测器的单元间距，把附件二的数据近似看做矩形方格对圆形的逼近，列方程组并在`matlab`中用`solve`函数求解；求解旋转中心的位置，找到探测器转动的起始位置，终止位置以及旋转过程中的特殊位置（射线与长轴平行或垂直）。特殊位置需要分别找到探测器上的投影最长和最短时矩阵中的列数，投影长度的中心分别为此时椭圆圆心的横坐标，根据几何关系，最终得到旋转中心的坐标。

确定旋转中心的位置后，我们利用旋转中心的投影在探测器上不变的特性，计算180个角度。确定圆边界和中心的最近距离后，利用圆的切线方程，平行线之间距离公式可以计算得到此列数据对应的角度。参数标定结果为：旋转中心 $(-9.2683, 5.9483)mm$ ，单元距离为 $0.2767mm$ ，180个角度为 -60.1643 度到 -61.6703 度。

2.2 问题二及问题三的分析与思路

由问题可知，此题的关键在于重建图像，由文献可知，重建图像可以采用`iradon`卷积反投影函数，利用此函数可以更为简单的得到重建图像矩阵。由于此函数过程误差累积，最终生成的图像还需进行去噪操作。对比去噪前，去噪后可以使得衰减程度的估计更加精确。最终解决问题，生成数据。

对于特殊的十个位置的吸收率。利用附件4的10组数据，说明了10个点在托盘上的位置。建立以托盘中心为原点的平面直角坐标系，使其能够与CT 图像重建程序运行的结果图2保持一致。运行程序得到的矩阵恰好覆盖托盘且位置中心为托盘中心。此时可以得到 256×256 个坐标即其上对应的值，即吸收强度。将十个点的坐标代入其中，利用插值法计算到这些点对应的吸收强度。

2.3 问题四的分析与思路

题中对于精度和稳定性没有做出说明，因此，我们接下来对这两种性质做出相对清晰的定义，用于模型的优化。问题四的要求是首先对问题一所建立的模型，参数的精度稳定性进行分析。

对于精度，可以从误差分析入手，由于人工计数过程产生的误差不属于系统误差，所以先忽略掉。其次，由于抽样方法产生的误差（即探测器间距计算过程中）或由于数据离散不连续产生的误差（即旋转中心坐标计算过程中，旋转角度的计算过程中）都可以通过改良抽样方法，使用插值拟合逼近连续而改进。

对于稳定性，改进模型，保持重建图像的几何形状和位置信息，减小重建图像中的误差。

由于模型得到了优化，仍采用已有的模板对优化模型进行精度和准确性的检验。然后自行设计新模板，对应标定模型。设计新模板则可以从四方面入手，即：中心位置的确定，旋转角度的确定，探测器的间距确定以及模板吸收强度的确定。根据四方面的不同，确定一个新模板，检验模型可行性。

三 模型假设

- 假设射线均匀，射线的衰减在模板以外的区域不会发生；
- 假设转盘转角不受限制，可以逆时针旋转到任意位置；
- 附件二中的数据与射线穿过模板的长度成正比例；
- 假设每个探测器单元质量良好，可以准确接受并反映信息；

四 符号说明

表 1: 符号表

符号	含义
R	小圆半径.
k, b	设射线的方程为 $y = kx + b$, k 为射线斜率, b 为射线截距.
α	一个关于能量衰减的比例系数.
w	这条直线在探测器的上显示的数值.
ω_x, ω_y	旋转中心的 x, y 坐标值.
O_x, O_y	假定竖直状态的椭圆圆心 x, y 坐标值. (此圆心与探测器中心连线与探测器垂直).
o_1	旋转过程中的竖直状态的椭圆圆心所对应的探测器位置.
o_2	旋转过程中的水平状态的椭圆圆心所对应的探测器位置.
δ	探测器单元间距取值为 $\delta \approx 0.2767mm$.
l	表示两平行线之间的距离.
s	表示计算小圆形边缘与这行数据的间隔格数.
R	投影矩阵.
θ	一个包含所有扫描角度的向量，这时每两个相邻角度间隔相等. 或为一个标量值，等于相邻两个扫描角度的间隔; ^[2] .
$interp$	插值函数.
$filter$	滤波函数.
$frequency_{scaling}$	一个标量值，取值范围 $[0, 1]$. 通过缩放滤波函数的频率修改滤波函数 ^[2] .
$output_{size}$	一个标量，用来规定重建图像的行数和列数.
h	为形状对应系数，且恒不小于1.
α	模型稳定性.
β	稳定性判别值.
S	为模板占据的面积.
\bar{I}	为某个单位上模型反投影得到的吸收率.
I	为对应单位上的实际的吸收率.

五 问题一

5.0.1 探测器单元间距的确定

问题一中需要三个值，CT的旋转中心，180组数据的旋转角度，以及探测器的间距。由附件1 中可以看出托盘上的物件是一个椭圆和一个圆，且两者的质地均匀吸收射线的强度也是均匀的。为求三个值，则需主要分析附件2的内容。附件2 由180列512行的数据组成，表示180 个角度，每个角度的各个探测器接收到的射线强度。

为使数据更加直观，把数据导入Matlab做三维图，以列为 x ,行为 y , 数据的大小表示 z , 可以得到如下图2 （作图代码见附录1）；

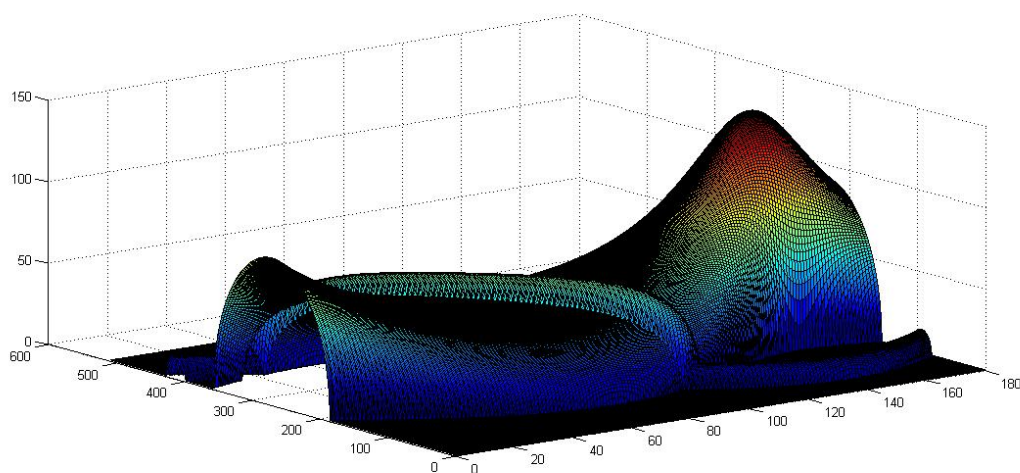


图 2: 附件2数据图

可以看出此图中突出的一条带状吸收强度，就是小圆形产生的。对应到数据中，可以发现在大椭圆与小圆形重合之前，可以用小圆求取光的照射角度和探测器的间距的。

5.1 模型一的建立与求解

首先，由于数据非常密集，可以把数据表格的每一小格看做对圆形的逼近，以小圆圆心为原点，椭圆的长轴方向为 y 轴，横轴方向为 x 轴，具体如图3：

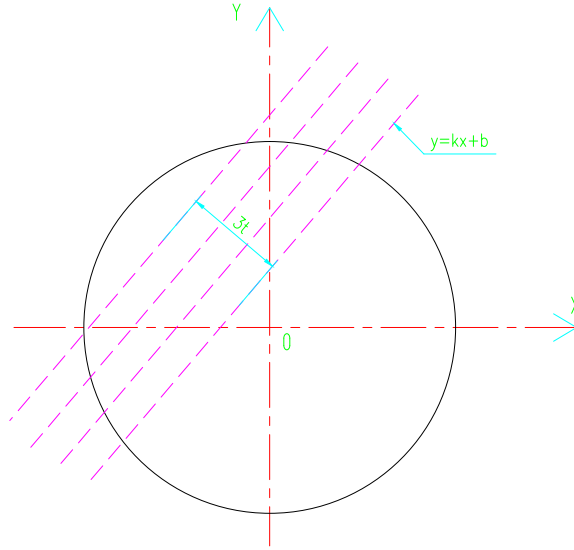


图 3: 小圆分析示意图

选取等距的四条射线，两两间距为 δ ，因为衰减程度和距离有关^[1]，因为附件二中有衰减大于100的数字，而板的长度为100mm，

所以一定在衰减能量和距离中存在一个比值，设此比值为 α ，
所以根据衰减的能量乘以 α 等于长度可以列出以下式子。

$$\sqrt{R^2 - \frac{b^2}{k^2 + 1}} = \alpha * w \quad (1)$$

注：

- R : 小圆半径；
- k, b : 设射线的方程为 $y = kx + b$, k 为射线斜率， b 为射线截距；
- α : 能量衰减系数；
- w : 这条直线在探测器的上显示的数值；

根据抽样原理，进行方便抽样，抽取第112列，第79-82行数据进行计算，

并检验

$$\begin{aligned}
 \sqrt{R^2 - \frac{b^2}{k^2 + 1}} &= \alpha \times 1.2430 \\
 \sqrt{R^2 - \frac{b^2}{\sqrt{k^2 + 1} - \delta}} &= \alpha \times 5.3196 \\
 \sqrt{R^2 - \frac{b^2}{\sqrt{k^2 + 1} - 2\delta}} &= \alpha \times 7.2887 \\
 \sqrt{R^2 - \frac{b^2}{\sqrt{k^2 + 1} - 3\delta}} &= \alpha \times 8.7194
 \end{aligned} \tag{2}$$

利用Matlab计算可得结果为(代码见附录程序2):

$$\delta = 0.276661mm;$$

使用第113列，第78-81行数据进行检验(代码见附录程序2)，得结果为:

$$\delta = 0.276642mm;$$

所以，由上式两次计算取平均值得出（结果由计算器计算得出）:

探测器的单元间隔为:

$$\delta \approx 0.2767mm$$

5.1.1 探测器旋转中心位置的确定

为方便这一小问题的计算我们重新确立坐标轴为，以探测器水平方向为 X 轴，垂直方向为 y 轴，坐标(0,0)为图1椭圆中心。

首先对附件二数据进一步分析得出图4如下（使用excel条件格式→色阶→其他规则→设置颜色即可得出）

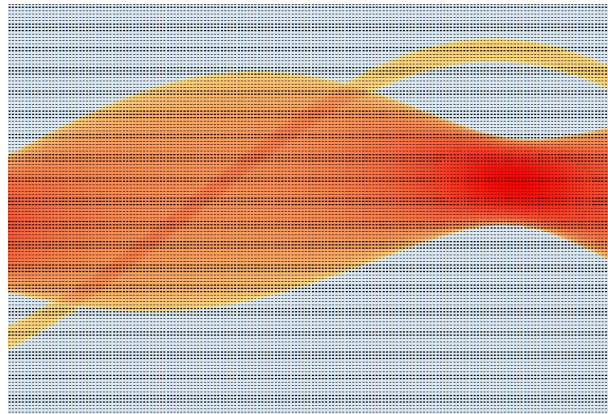


图 4: 数据图

又因为此数据为逆时针旋转得到，通过图中显示的大椭圆与小圆形的间距先减小后增大再减小的趋势，可以得出椭圆与探测器的相对位置变化顺序，选取4个特殊位置，探测器转动，利用运动的相对性可以看做转盘转动，作图如下:

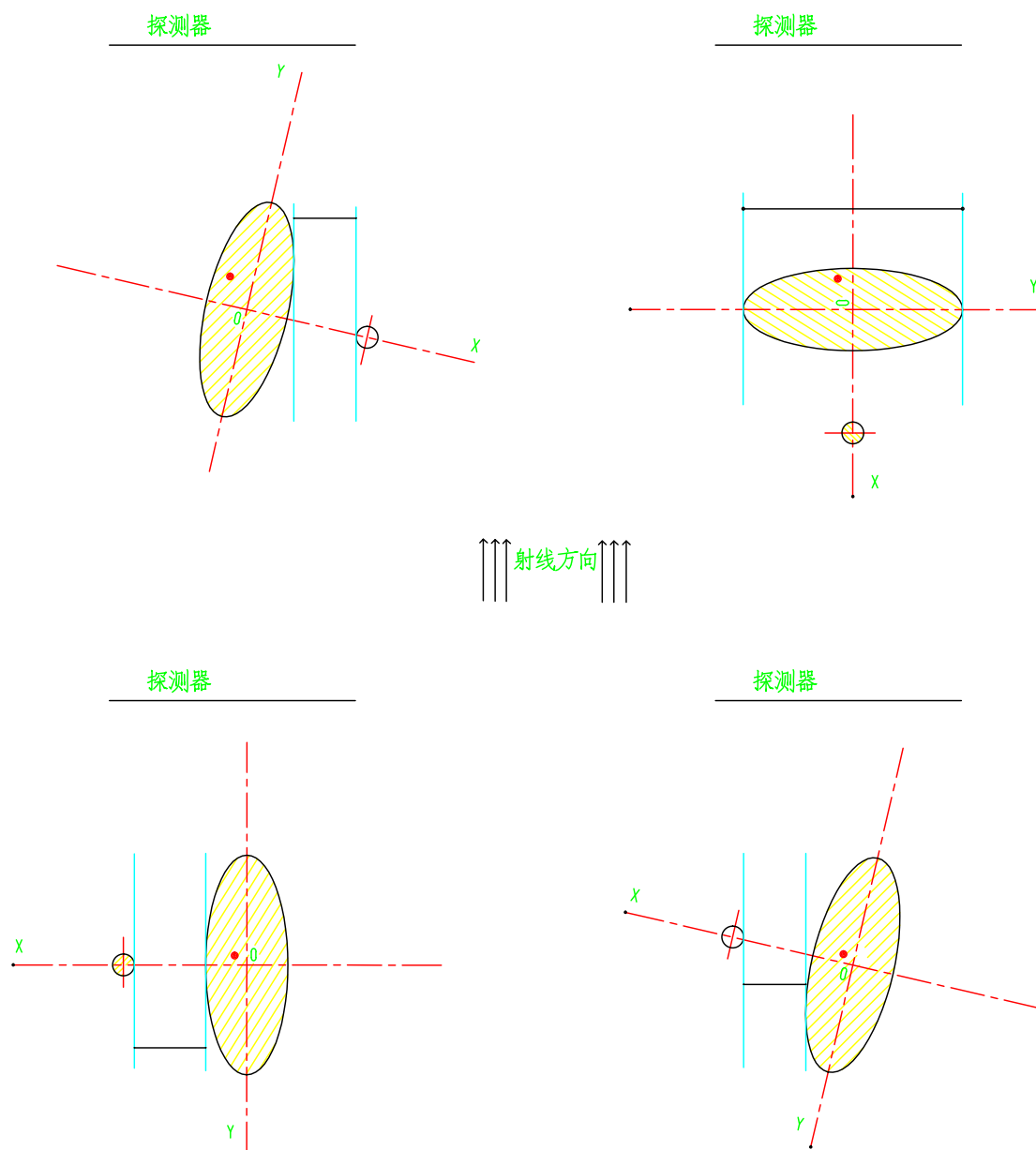


图 5: 初始-水平-倒竖直-终止状态图

设旋转中心为 ω ,椭圆圆心为 o ,由于在物体围绕旋转中心转180度时,物体上同一点前后两次的位置的连线中心即为旋转中心。所以为了方便计算我们假设一个竖直状态为初始状态,由于如果初始状态为竖直的,这时的椭圆的圆心会在正方形板的中心设为 o ,对应探测器板的中心即为第256个与第257个探测器中间,位置很好确定。作图如图所示:

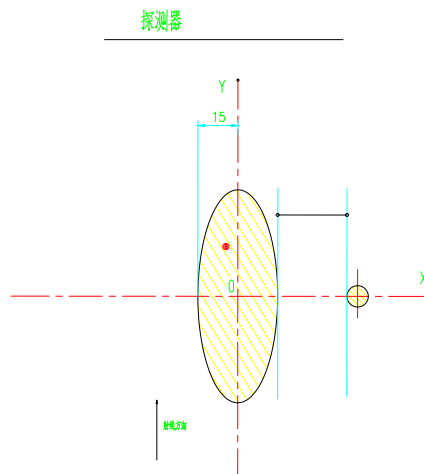


图 6: 假设正竖直状态图

所以由图6及图5（3）即可确定旋转中心在探测器上的 x 。所以,只需计算出旋转过程中的竖直状态的椭圆圆心所对应的探测器位置设为 o_1 ,则旋转中心计算公式为:

$$\omega_x = (o_x - o_1) \times \delta \quad (3)$$

注:

- ω_x : 旋转中心的 x 坐标值;
- o_x : 假定竖直状态的椭圆圆心 x 坐标值 (此圆心与探测器中心连线与探测器垂直);
- o_1 : 旋转过程中的竖直状态的椭圆圆心所对应的探测器位置;
- δ : 探测器单元间距取值为 $\delta \approx 0.2767mm$;

利用`matlab`编程,计算图4中椭圆形成区域红色最窄的区域的宽度的一半就是 o_1 。(只有当射线与椭圆长轴平行的时候才会有投影最少,所以投影最窄的区域就是我们需要的旋转过程中的竖直状态的椭圆,所以这个区域的宽度中心对应的探测器位置就是 o_1)。

根据函数统计每一列中大于零的数的个数,个数最小的即为我们需要的最窄列,结果有2组,其中第150列以及第15列2大于零的个数都为137,选取中的每一列的不为零的首位数和末尾数的平均值,即为光源平行于长轴时椭圆圆心的横坐标。

所以结合公式(2)可以得出:

$$\begin{aligned} o_{1x} &= 223 \\ \omega_x &= (o_x - o_1) \times \delta \\ &\approx -9.2683mm \end{aligned} \quad (4)$$

旋转中心竖直方向的坐标同理，由初始状态的竖直椭圆旋转到水平方向，一共旋转了90度，在探测器上的投影则是为最宽的位置，因为其等于长轴长度，所以同上理以及使用函数计算可得，有8组并且其中大于零的个数都为289，选取中的每一列的不为零的首位数和末尾数的平均值，其物理意义为光源垂直于长轴时椭圆圆心的横坐标。所以可以得出：

$$\begin{aligned} o_{2x} &= 233.75 \\ \omega_y &= (o_y - o_1) \times \delta \\ &\approx 5.9483mm \end{aligned} \quad (5)$$

注：

- ω_y ：旋转中心的 y 坐标值；
- o_y ：假定竖直状态的椭圆圆心 y 坐标值（此圆心与探测器中心连线与探测器垂直）；
- o_2 ：旋转过程中的水平状态的椭圆圆心所对应的探测器位置；
- δ ：探测器单元间距取值为 $\delta \approx 0.2767mm$ ；

所以，探测器旋转中心的坐标为：

$$\omega_y \approx 5.9483mm \quad (6)$$

$$\omega_x \approx -9.2683mm \quad (7)$$

5.1.2 附件二180组角度的确定

根据已经确定的旋转中心，利用附件二小圆形的数据则可以利用以下方法求得角度。

首先，我们知道旋转中心在探测器上的投影固定，且在探测器中心256.5处，计算小圆形和旋转中心的最小距离可以用附件二的数据，即计算小圆形边缘与这行数据的间隔格数乘以探测器单元间隔。由几何关系可以得出如图：

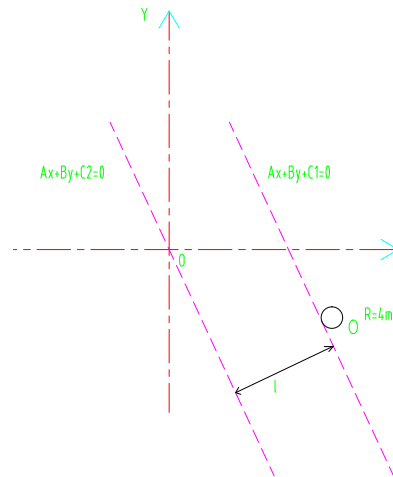


图 7: 角度求解示意图

由图中关系可以得到，如下公式：

$$\begin{aligned}
 A\omega_x + B\omega_y + C_2 &= 0 \\
 \frac{Ao_x + Bo_y + C_1}{\sqrt{A^2 + B^2}} &= R^2 \\
 \frac{|C_1 - C_2|}{\sqrt{A^2 + B^2}} &= l \\
 s * \delta &= l
 \end{aligned} \tag{8}$$

注：

- ω_y, ω_x : 旋转中心的坐标值 $\omega_y \approx 5.9483mm, \omega_x \approx -9.2683mm$;
- o_y, o_x : 小圆圆心 $y = 0mm, x = 45mm$ 坐标值;
- $Ax + By + C_i$: 表示第 i 条直线斜率为 $-A/B$, 截距为 $-C_i/B$ 的直线;
- l : 表示两平行线之间的距离;
- s : 表示计算小圆形边缘与这行数据的间隔格数;
- δ : 表示探测器单元间隔取值为 $\delta \approx 0.2767mm$;
- R : 表示小圆半径。

经过人工计数，得到180组数据的 S , 每组有四个结果，通过结合旋转过程初始，终止状态，可以筛选出的180个角度为下表：

表 2: 180个角度值(单位: 度)

	1	2	3	4	5	6
1	-60.1643	-58.7107	-58.2368	-56.8440	-55.9380	-55.0486
2	-54.1746	-52.8907	-52.0514	-51.2246	-50.0062	-49.2076
3	-48.0287	-46.8711	-46.1104	-45.3580	-44.6136	-43.8767
4	-42.7849	-41.7083	-40.6459	-39.9451	-38.9045	-37.8758
5	-37.5354	-36.8584	-35.8515	-35.5181	-34.8546	-34.1952
6	-33.8671	-33.2136	-32.8883	-31.9180	-30.9554	-29.6834
7	-28.7374	-27.7978	-26.8643	-25.9364	-24.7076	-23.7917
8	-22.8804	-21.6720	-20.7704	-19.8723	-18.6802	-17.7898
9	-16.9021	-15.7225	-14.8405	-13.6677	-12.7901	-11.9141
10	-11.0393	0.0000	11.62239	12.4979	13.6677	14.5470
11	15.4283	16.6068	17.49362	18.6802	19.5738	20.4706
12	21.6720	22.5776	23.48747	24.7076	25.3209	25.6284
13	25.9364	26.5544	27.4860	28.4235	29.0520	29.6834
14	30.6362	31.5963	32.8883	33.5399	34.5244	35.5181
15	36.5216	37.5354	38.5603	39.5969	40.6459	41.7083
16	42.4244	43.5110	44.9848	45.7332	46.4896	47.6405
17	48.8122	49.2076	49.6056	50.4096	50.8157	51.6365
18	52.8907	53.7432	54.60972	55.4913	56.8440	57.7678
19	58.7107	60.1643	60.6601	61.6703	62.7071	63.7729
20	64.3176	65.4329	66.5858	67.7809	68.3960	69.6655
21	70.3217	71.6829	72.3908	73.8694	74.6446	75.4472
22	76.2805	77.1487	79.0104	80.0185	81.0915	81.0915
23	82.2444	83.4989	84.8884	84.8884	86.4705	88.3601
24	88.3601	89.1284	89.1284	89.1284	83.0591	83.0591
25	83.0591	83.0591	83.0591	83.0591	83.0591	83.0591
26	89.1284	89.1284	89.1284	-88.3601	-88.3601	-86.4705
27	-86.4705	-84.8884	-83.4989	-83.4989	-82.2444	-81.0915
28	-80.0185	-79.0104	-78.0565	-77.1487	-76.2805	-75.4472
29	-73.8694	-73.1189	-72.3908	-70.9937	-69.6655	-68.3960
30	-67.1777	-66.0044	-64.8708	-63.7729	-62.7071	-61.6703

六 问题二及问题三

6.1 重建图像

问题二和问题三非常类似，所以本文在此合并解决。首先，问题二（三）是在问题一的基础上，利用问题一已经确定的探测器与置物板指标，利用问题二（三）数据，重建图像。具体做法如下：

首先，导入附件三（四）中位置物体在光源照射下的投影矩阵，对其使用 $iradon$ 函数进行重建图像。通过把投影矩阵设为 R 使用 $iradon$ 函数反投影，得到重建图像矩阵设为 I ，其原理是基于 $R-L$ 滤波器的滤波反投影法^[2]。投影矩阵经过傅里叶变换并和滤波函数 H 相乘，得到滤波后频域上的投影矩阵 $fff(R) \times H$ ，再将其转换到空域，得到空域中滤波后的投影矩阵 R' ，对 R' 插值，成为 R'' ，最后反投影得到重建的图像矩阵 I 。

$iradon$ 卷积反投影函数具体公式如下：

$$I = iradon(R, \theta, interp, filter, frequency_scaling, output_size) \quad (9)$$

注：

- R ：投影矩阵；
- θ ：一个标量值，等于相邻两个扫描角度的间隔^[2]；
- $interp$ ：插值函数；
- $filter$ ：滤波函数；
- $frequency_scaling$ ：一个标量值，取值范围 $[0, 1]$ ，通过缩放滤波函数的频率修改滤波函数；^[2]
- $output_size$ ：一个标量，用来规定重建图像的行数和列数。

使用 $matlab$ 计算函数,得到如图：

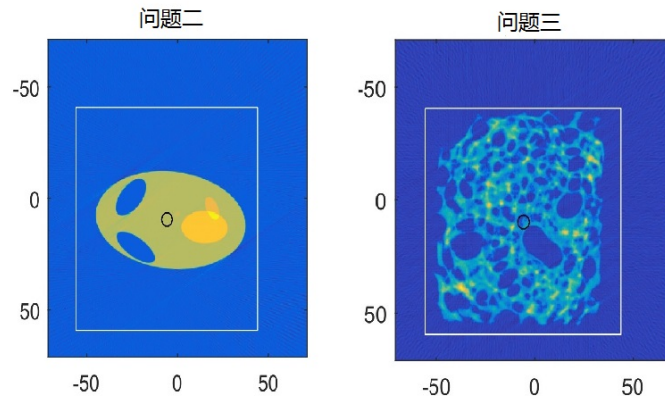


图 8: 问题二及问题三示意图

由问题二得出的图像可知，模板集中在托盘中上部，在边缘部分包含两个空的类圆形；由问题三得出的图像可知，模板覆盖了托盘的大部分，上面有较多的空洞。从图中可以看到问题二和问题三求解的未知介质都是不规则的，不均匀的介质，它们的不同部位的吸收强度大多不同。图形中间的黑色小圆为托盘的中心，两个未知介质在正方形托盘的位置如上图所示。

6.2 10个位置吸收率

附件4的10组数据反映了所求10个点在托盘上的位置。建立以托盘中心为原点的平面直角坐标系如图1，能够与CT图像重建程序运行的结果图2保持一致。程序结果是256*256的矩阵，恰好覆盖托盘且位置中心为托盘中心。此时可以得到256*256个坐标即其上对应的值，即吸收强度。将十个点的坐标代入其中，插值的到这些点对应的吸收强度。

示意图如下：

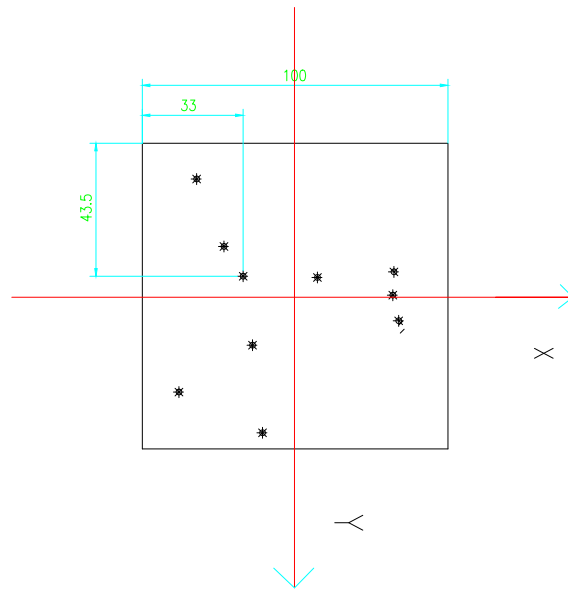


图 9: 示意图

计算所得结果如下：

表 3: 问题二10个特殊点值（保留4位小数）

序号	X轴坐标(mm)	Y轴坐标(mm)	吸收率
1	10.0000	18.0000	-0.0689
2	34.5000	25.0000	-0.1015
3	43.5000	33.0000	0.0119
4	45.0000	75.5000	-0.0005
5	48.5000	55.5000	0.0096
6	50.0000	75.5000	0.0012
7	56.0000	76.5000	-0.0004
8	65.5000	37.0000	0.0008
9	79.5000	18.0000	0.0233
10	98.5000	43.5000	-0.0102

表 4: 问题三10个特殊点值（保留4位小数）

序号	X轴坐标(mm)	Y轴坐标(mm)	吸收率
1	10.0000	18.0000	-0.0689
2	34.5000	25.0000	-0.1015
3	43.5000	33.0000	1.1429
4	45.0000	75.5000	0.0076
5	48.5000	55.5000	0.0429
6	50.0000	75.5000	0.0640
7	56.0000	76.5000	-0.0398
8	65.5000	37.0000	0.0554
9	79.5000	18.0000	1.6159
10	98.5000	43.5000	-0.0256

七 问题四

定义 1 精度：模型的精度是用于定性描述在数学模型下CT系统对未知模板的信息接收的完整程度的量。若系统在任意一个角度上都可以探测出或者通过模型计算出位置模板辐射剩余能量，则称该模型具有好的精度。

定义 2 稳定性：模型的稳定性是用于定量描述在数学模型下CT系统对已知模板的信息（位置，几何形状，吸收率等）反映的准确程度的量。若模型的稳定性为 $\alpha \geq 0$ ，不大于某个人为设定的常值 β ，则认为模型是稳定的。

$$\alpha = h/S * \sum \bar{I}/I$$

其中

- h ：为形状对应系数，且恒不小于1；
- α ：模型稳定性；

- β : 稳定性判别值, 设其值为0.0077;
- S : 为模板占据的面积;
- \bar{I} : 为某个单位上模型反投影得到的吸收率;
- I : 为对应单位上的实际的吸收率。

7.1 旋转中心坐标精度

在问题一中, 进行CT系统的参数标定时, 为求解旋转中心坐标, 我们采用人工计数测量单元的个数的方法计算距离。因为附件二的数据是离散的, 非连续的, 因此 a 和 b (椭圆投影长度最长或最短时投影中心, 即椭圆圆心横坐标的绝对值) 会存在较大的误差。将投影矩阵进行不同密度程度的插值后, 相邻数据之间的差值减小, 结果的精确程度会提高, 原理近似于定积分的逼近思想。通过讨论插值密度对 a 和 b 的影响, 来理解插值密度对于求解旋转中心坐标精度的影响。

当插值后的间距分别为0.6 ~ 1时, 即插值后的行数在512 ~ 852之间。它们的值的变化与中心坐标的变化成正比。可以从表格中看到, a 的变化是随着间距的减小, 即插值密度变大, a 的精度越大, b 的趋势也相同。所以得出结论: 当插值越密, a 和 b 的精度越大, 求解出来的中心坐标越准确。

a 和 b 变化如表5所示:

表 5: 插值运算对精度的影响					
间距	1	0.9	0.8	0.7	0.6
a	233.7143	248	291	331	389
精度 a	0	0.06112	0.2451	0.4162	0.6644
b	223	261	278.5	248	281
精度 b	0	0.1704	0.2489	0.1121	0.2601

7.2 探测器间距精度

在问题一中, 为标定探测器单元间距, 根据公式 (1) 的计算, 需要光源只照到小圆的每一列数据中的等间距或者连续的四组值, 为方便计算, 采取了方便抽样原则, 虽然进行了检验, 但数据本身存在的系统误差和选取数据过程中造成的误差累积成较大的误差。

为更加精确的计算间距, 采用简单随机抽样的思想, 利用Matlab编程 (代码在附录中) 在0-180中产生5个数, 确定列数, 选取该列第一个值, 再产生1-10 中1个数, 确定每组4个数的位置公差, 从而产生五组样本, 每组样本包含4 个数, 进行计算, 得到其对应的单元距离, 并且求取这五组值得平均值, 做比较。可以看到平均的单位距离比大部分组的精确度要大一些, 所以在求取单元距离时要尽可能随机选取较多组进行计算, 求其平均值。在这一问题中, 样本的随机性越好, 样本的数量越多, 则误差将会越小, 从而提高精确度。

如下表所示:

表 6: 插值运算对精度的影响

单元距离（保留4位有效数字）	0.2770	0.2768	0.2767	0.2770	0.2766
精确度（保留5位有效数字）	0.00114	0.00039	0.00026	0.00116	1.74498E-09
平均单元距离（为显示其精确性不作保留）	0.276806379				
精确度（为显示其精确性不作保留）	0.00059558				

7.3 噪点对稳定性的影响

定义 3 噪点：是指在函数 $iradon$ 运行过程中产生的误差。

问题二（三）中使用 $iradon$ 函数求解反投影，由于此函数在各个步骤中会产生一些误差，所以误差积累会形成较大偏差影响结果^[1]，从而影响装置的稳定性。所以得到的数据即图像每一点的衰减程度存在一定误差，需要对其进行去噪处理，重建后的图像矩阵，进行取其绝对值与本身的平均值处理，即负数归零处理，再消去矩阵中小于其最大值的0.001的所有值。最后把去噪前和去噪后的图像对比，可以得到更加精确的衰减程度，即其吸收率。

使用 $Matlab$ 运行函数,结果如图10,11所示（代码见附录中,数据见附件中）。去噪后的图片更为清晰分明，有利于系统稳定性，与被测物体实际情况的诊断。

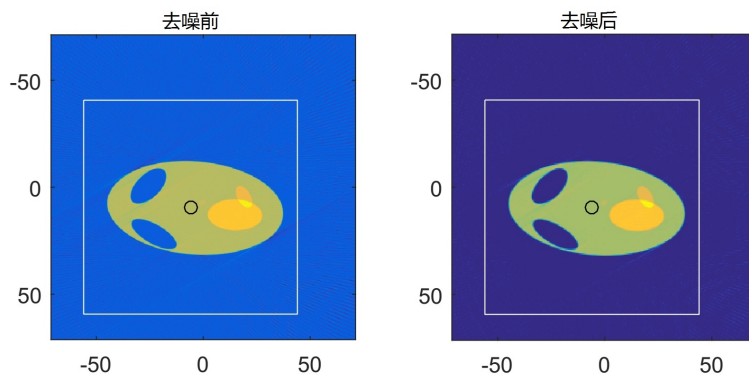


图 10: 问题二示意图

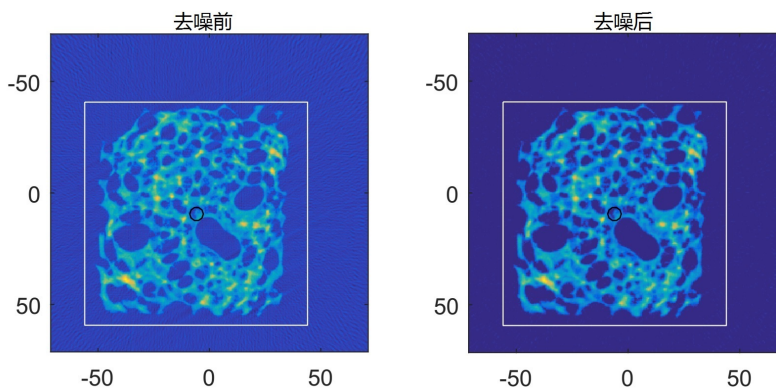


图 11: 问题三示意图

7.4 自定义参数模板

首先是中心位置，仍设置在 $(-9.2683mm, 5.9483mm)$ 由于这一点对于一个模板来说是具有普通性的一个点，所以确定为这一点，更可以检验模型的普适性；旋转角度则首先设定为均匀分布，即 $0 \sim 360$ 度上均匀分布；探测器的间距仍为 $0.2767mm$ ，也体现了一般性；对于模板吸收强度,我们进行了大的调整,图示如下（深蓝色线条表示此处吸收率较强）：

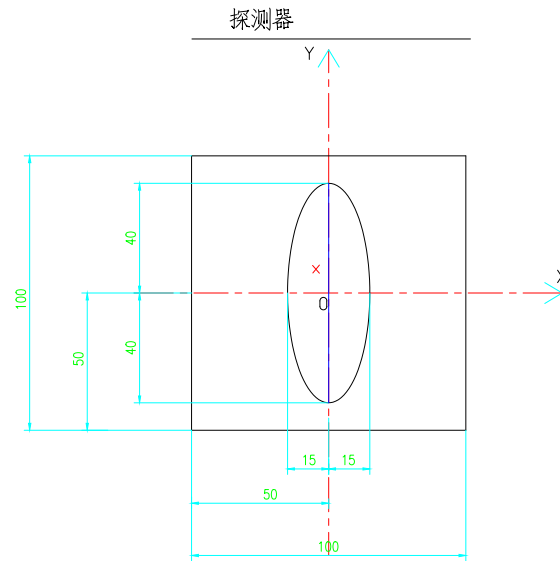


图 12: 新模板吸收信息图

对于已知吸收强度的图形投影计算，用 $iradon$ 函数的反函数 $radon$ 函数^[2]，具体形式如下：

$$R = radon(I, \theta)$$

注：

- R : 投影矩阵;
- θ : 一个标量值，等于相邻两个扫描角度的间隔;
- I : 模板图像;

利用 $radon$ 函数，导入新模板吸收率数据为 I ,对 θ 取标量，循环带入求取其投影数据。编写 $Matlab$ 代码实现，并作接受信息图如下图所示：

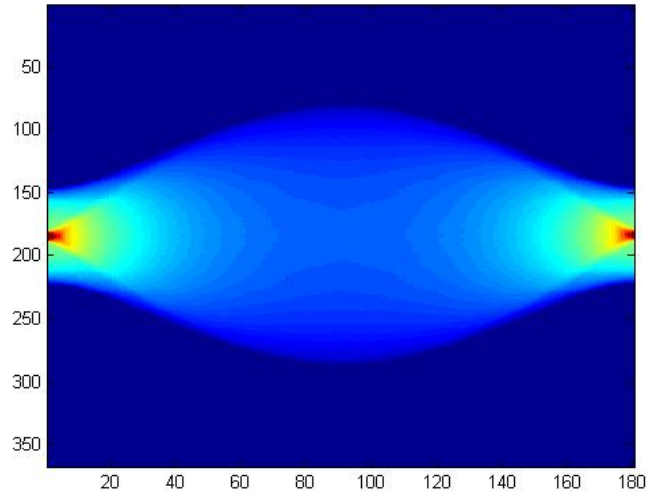


图 13: 新模板接受信息图

通过上述方法得到新模板接受信息数据，同问题二，三理，使用 $iradon$ 函数，求解反投影图，并进行去噪操作可以得到如下图形：

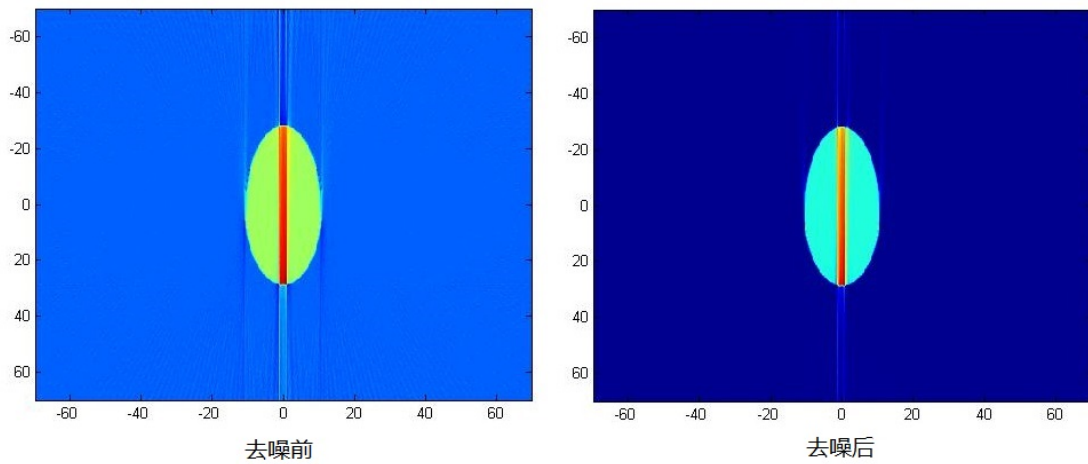


图 14: 去噪前后示意图

由上述方法去噪后可以求出， $\alpha = 0.0058$ ，且 $\alpha < \beta$ ，所以，可以验证该模型是稳定的。

八 模型的评价

8.1 模型优点

该模型运用了几何分析的方法，较为简便的计算得到了 CT 系统的中心位置，180组数据对应的角度。其中，在利用 $Matlab$ 编程时，还应用到了矩阵的运算等代数方法。求问题二及问题三时，为方便简单是用了 $iradon$ 函数模型，利用滤波反投影，直接具体的计算出了物体的各个点的吸收率，并利用去噪函数，对个阶段产生的误差最后形成的误差累计，进行消除，得到更为真实的吸收率。

8.2 模型缺点

- 在稳定性分析去噪时，规定把矩阵最大值的0.001以下的数据清零，这个去噪的精度只有三位小数；
- 计算探测器单元距离时，由于只抽取了其中的两组样本值，结果会存在误差；
- 问题一中假设180个角度中总存在两个特殊角度，即光源分别垂直于椭圆长轴和短轴，以此计算旋转中心的坐标，这个假设存在一定的主观性；

参考文献

- [1] 毛小渊, 二维CT图像重建算法研究, [J]南昌航空大学,2016.6.
- [2] <http://blog.sina.com.cn/s/blog_12ff84a8a0102waut.html>,iradon 函数与radon 函数.
- [3] 叶海霞, 工业CT窄角扇束卷积反投影并行图像重建研究, [J] 重庆大学,2003.
- [4] 薛红, 反向直线扫描CT 图像重建算法研究, [J] 重庆大学,2016.
- [5] 于源, CT投影数据不完全的代数重建算法研究与实现, [J] 东北大学,2009.
- [6] 罗海, CT 图像重建及运动伪影, [J]中国科学技术大学,2011.5.

附 录

A 问题一

1.1 附件2作图

```
x=[1:1:180];  
y=[1:1:512];  
z=xlsread('C:\Desktop\CUMCM2017Problems\A\A1.xlsx');  
surf(x,y,z)
```

A1.xlsx数据是附件二的数据。

1.2 δ 的求解

计算

```
[k,b,s]=solve('(16-(b/(k^2+1)^0.5-s)^2)/(16-b^2/(k^2+1))=(5.3196/1.2430)^2',  
'(16-(b/(k^2+1)^0.5-2*s)^2)/(16-(b/(k^2+1)^0.5-s)^2)=(7.2887/5.3196)^2',  
'(16-(b/(k^2+1)^0.5-3*s)^2)/(16-(b/(k^2+1)^0.5-2*s)^2)=(8.7194/7.2887)^2',  
'k','b','s')
```

验算

```
[k,b,s]=solve('(16-(b/(k^2+1)^0.5-s)^2)/(16-b^2/(k^2+1))=(7.1045/5.0431)^2',  
'(16-(b/(k^2+1)^0.5-2*s)^2)/(16-(b/(k^2+1)^0.5-s)^2)=(8.5784/7.1045)^2',  
'(16-(b/(k^2+1)^0.5-3*s)^2)/(16-(b/(k^2+1)^0.5-2*s)^2)=(9.7355/8.5784)^2',  
'k','b','s')
```

1.3 寻找最宽最窄列代码

```
% function code3()  
  
A=xlsread('S:\A.xls',2);  
number=sum(A>0,1);  
num=number';  
numMax=max(num);  
numMin=min(num);  
x=find(num(:,1)==numMax);%最胖的列数  
  
x1=length(x);%胖的列数有几组?  
for i=1:x1  
x2=x(i);  
a=find(A(:,x2)>0);%最胖的这一列的大于零的数的行数  
a1=length(a);%最胖的数的行数的个数  
a2(1,i)=(a(1)+a(a1))/2;  
end  
a3=mean(a2)
```

```

y=find(num(:,1)==numMin);%最瘦的列数
y1=length(y);
for i=1:y1
y2=y(i);
b=find(A(:,y2)>0);%最瘦的这一列的大于零的数的行数
c=find(b>=100);
b=b(c);
b1=length(b);%最瘦的数的行数的个数
b2(1,i)=(b(1)+b(b1))/2;
end
b3=mean(b2)

```

1.4 180个角度计算代码

```

B1=[];B2=[];B3=[];
z=xlsread('S:\data.xlsx',1);
k=length(z);
for i=1:k
a=abs(z(i)-257);
syms c2 c1 A;
s=vpasolve([5.9482==A*(-1)*9.2681
+c2 (A*45+0+c1)^2==4*4*(A^2+1) (c1-c2)^2
==(A*A+1)*(a*0.276661)^2],[c2 c1 A]);
d1=double(s.c1);d2=double(s.c2);d3=double(s.A);
if length(d1)*length(d2)*length(d3)<24
d1=[d1;0;0];d2=[d2;0;0];d3=[d3;0;0];
end;
B1=[B1 d1];
B2=[B2 d2];
B3=[B3 d3];
B3=real(B3);

end
t=atan(B3);

```

B 问题四

2.1 增加精度

```
clear
```

```

clc
z=xlsread('C:\A.xls',2);
for i=1:180

    t1=[1:1:512];
    tt=[1:0.6:512];    %改变插值的密度
    y2(:,i)=interp1(t1,z(:,i),tt);

end;

```

2.2 五组随机样本产生

```

y=180*rand(1,5);
y
x=10*rand(1,1);
x

```

2.3 问题二(三)去噪前后两图形代码

```

%适用于问题1,2,3
%探测器间距
d = 0.2778;
%中心相对圆点位置
xc = -33.5*d;yc = 21.5*d;
%导入图形
T = load('1.dat',,);%1, 2,3,4
%规范化
T = [zeros(150,180); T; zeros(150,180)];
imagesc(phantom)%检验
%作出物品
figure
img = iradon(T,[0:179]-60);
n = size(img,1);
[x, y] = meshgrid([-n/2:n/2]*d);%修改
%原图像
imagesc(x(1,:), y(:,1), img)
hold on
%除杂后图像
% img1=(imgg+abs(imgg))/2;dd=max(max(img1));
img1(find(img1)<=0.001*dd)=0; imagesc(x(1,:), y(:,1),img1);hold on

%确定原点和托盘位置
plot(-yc,-xc,'ok')

```



```

xp = [-50 50 50 -50 -50]';
xp=xp-yc;
yp = [-50 -50 50 50 -50]';
yp=yp-xc;
plot(xp,yp,'-w')
axis image

```

2.4 问题四去噪前后两图形代码

```

A=load('4.dat');
x=[];for theta=0:179
    B=radon(A,theta);x=[x B];
end;
xx=x;xx=[zeros(179,180);xx;zeros(180,180)];
xxx=iradon(xx,[0:179]);
figure(1);imagesc(x);figure(gcf);
figure(2);imagesc(xx);figure(gcf);
n=size(xxx,1);
[x,y]=meshgrid([-n/2:n/2]*0.2778);
figure(3);imagesc(x(1,:),y(:,1),xxx)
size(xxx)

img1=(xxx+abs(xxx))/2;
dd=max(max(img1));
img1(find(img1)<=0.001*dd)=0;
figure(4);imagesc(x(1,:), y(:,1),img1);

```