The ART of tomography

Timothy G. Feeman Villanova University
2016 NCAA Men's Basketball National Champions

In 1979, the Nobel Prize for Medicine and Physiology was awarded jointly to **Allan McLeod Cormack** and **Godfrey Newbold Hounsfield**, two pioneering scientist-engineers whose work, in the 1960s and early 1970s, led to the development of **computerized axial tomography**, popularly known as the CAT or CT scan.

Cormack, then a Professor at Tufts University, in Massachusetts, developed algorithms based on the Fourier transform to create an image from X-ray data.

Hounsfield, a research scientist at EMI Central Research Laboratories in the United Kingdom, designed the first operational CT scanner as well as the first commercially available model. (Side note: EMI, or Electric and Musical Industries Ltd., published the EMI record label, among whose artists were the Beatles.)

About 70 million CT scans are performed each year in the United States.

A typical CT scan is generated from a set of about 30000 X-rays beams, consisting of 150 to 200 beams in each of 180 directions.

Beer's Law of X-ray behavior: *The rate of change of intensity per millimeter of an (ideal) X-ray beam is (locally) proportional to the intensity of the beam.*

Expressed as a differential equation: $$\frac{dI}{dx} = -f(x) \cdot I(x)$$

The DE is separable. Thus, $$\int_{x_0}^{x_1} f(x)\,dx = \ln\left(\frac{I_0}{I_1}\right)$$

What we can measure: An X-ray emission/detection machine (the scanner) measures the values of I_0 and I_1. Hence, we can compute $\int_{x_0}^{x_1} f(x)\,dx$, the integral of the (unknown) attenuation coefficient function along the path of the X-ray.

What we want to know: The value of f at each point depends on the nature of the medium located there. Thus, the idea behind the CT scan is that we can determine the value of f if we know the values of its line integrals along many different lines. This is an *inverse problem.*

There are two principal approaches to reconstructing an image from the X-Ray data.

Fourier transforms were used in the seminal work of Cormack and are used in the algorithms of most CT scanners today. Fourier transform methods begin with a continuous theory, culminating in the so-called **filtered back-projection formula**, which is then modeled using discrete methods.

However, the first CT scanner, designed in the late 1960s by Godfrey Hounsfield, used an approach grounded in linear algebra to generate an image from the machine readings. Algorithms that adopt this point of view are known as **algebraic reconstruction techniques**, or ART, for short. In this talk, we look at a few basic mathematical elements of ART.

ART techniques are especially useful for

- non-destructive material testing — abrupt changes in the density of the material being scanned require image reconstruction methods that can provide high contrast;
- SPECT (single-photon emission computerized tomography) and PET (positron emission tomography) — difficulty in measuring the attenuation can render transform methods less reliable than usual;
- incomplete data collection — the range of the scan is restricted in order to limit the patient's exposure. This includes digital breast tomosynthesis (DBT).

ART treats the problem of image reconstruction as a discrete problem. Any image that we produce will be constructed inside a rectangular grid of picture elements, or pixels.

As a simple model, suppose we have a thin slice of material (the medium) in the shape of a square. The square is divided into a 3-by-3 rectangular grid of smaller squares each of which is either black or white. Each white square absorbs "1 unit" of X-ray energy while the black squares do not absorb X-ray energy. (So the white squares act like bone, say, and the black ones act like air.)

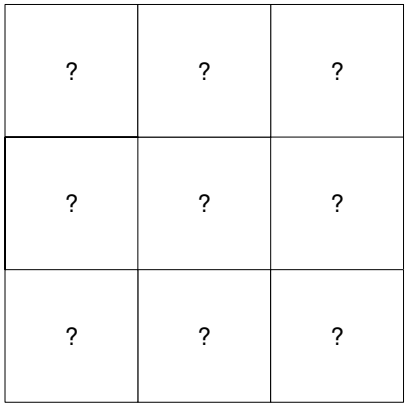


Figure 1: We use scan data to assign a B/W color value to each square.

$$\text{Scan of Row \#1} = 2 \implies \text{2 W, 1 B in Row \#1}$$
$$\text{Scan of Col \#1} = 1 \implies \text{1 W, 2 B in Col \#1}$$

What can we say so far?

Continuing in this way:

$$\begin{aligned} \text{Scan of Row \#2} = 2 &\implies ??? \\ \text{Scan of Row \#3} = 1 &\implies ??? \\ \text{Scan of Col \#2} = 2 &\implies ??? \\ \text{Scan of Col \#3} = 2 &\implies ??? \end{aligned}$$

What are the possible configurations? How many White squares? How many Black squares? What additional information might help to pinpoint a unique shading pattern?

Here is one possible configuration consistent with these measurements.

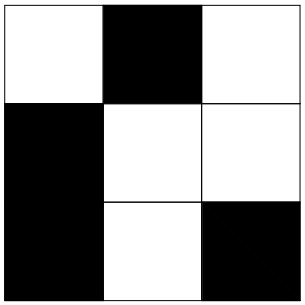


Figure 2: This grid of white and black squares has a prescribed X-ray energy absorption for each row and column.

For a highly entertaining elaboration on this model for thinking about CT scans, see Dewdney, A. K., How to resurrect a cat from its grin, in Mathematical Recreations, *Scientific American* **263**(3) (September 1990), 174–177.

ART treats the problem of image reconstruction as a discrete problem. Any image that we produce will be constructed inside a rectangular grid of picture elements, or pixels.

To form an image, a specific color value is assigned to each pixel. For instance, the color value assigned to a given pixel might be a *grayscale* value, a number between 0 (= black) and 1 (= white), that represents the density or attenuation coefficient of the matter in the sample at the location of the given pixel.

ART techniques use a system of constraints derived from the machine readings to compute these color values.

So, suppose an image is to be constructed in a K-by-K grid of pixels. Each pixel is really a small square in the plane.

For convenience, number the pixels like so: 1 through K from left to right across the top row; $K+1$ through $2K$ across the second row; and so on until, in the bottom row, we find pixels numbered $(K-1)K+1$ through K^2.

Next, assign the (for now unknown) color value x_k to the k^{th} pixel.

Now suppose that an X-ray beam following the line ℓ passes through the region of our image.

- The scanner tells us the change in intensity of the beam during its journey.
- According to *Beer's law*, the change in intensity as the beam passes through any one pixel is equal to the color value of that pixel times the length of the intersection of the beam's path ℓ with the pixel.
- Thus, the total change in intensity of the beam is equal to the sum of the intensity changes over the separate pixels. (Note that any one X-ray beam will hit relatively few pixels.)

So: let $\{\ell_j : j = 1, \ldots, J\}$ be the collection of lines corresponding to the X-ray beams in a full CT scan, and let p_j denote the measured change in the intensity of the beam along line ℓ_j.

Next: denote by $r_{j\,k}$ the length of the intersection of the line ℓ_j with pixel number k, for $j = 1, \ldots, J$ and $k = 1, \ldots, K^2$. In principle, these values are easy to compute. (*Caveat:* If we allow finite-width X-ray beams, rather than zero-width, then this computation becomes more complicated.)

Thus: we get the system of equations

$$p_j = \sum_{k=1}^{K^2} r_{j\,k}\, x_k \quad \text{for } j = 1, \ldots, J. \tag{1}$$

This is a system of J linear equations in K^2 unknowns (x_1, ..., x_{K^2}). Typically, both J and K^2 are on the order of 10^4 or 10^5, so **the system is large.**

A full CT scan might include 200 X-ray measurements at each of 180 different directions, for a total of 36000 equations in the system. A grid of 160×160 pixels gives 25600 unknowns and an over-determined system. To get an image with higher resolution, though, we may want to use a grid of 256×256 pixels, or 65536 unknowns. This results in a system that is heavily under-determined.

If the system of equations is over-determined, with more equations than unknowns, then the system likely does not have an exact solution. If the system is under-determined, with more unknowns than equations, then there may be infinitely many solutions, only one of which could possibly be the correct solution. In any case, due to errors in the emission/detection measurements, the equations are only estimates to begin with. So the system is not likely to have an exact solution.

Any particular line ℓ_j passes through relatively few of the pixels in the grid, on the order of K out of K^2 total pixels. Thus, most of the values $r_{j\,k}$ are equal to 0, meaning that the coefficient matrix $(r_{j\,k})$ for this system is large but *sparse*. With typical values of J and K, fewer than one percent of the entries in the coefficient matrix of the system are nonzero.

Kaczmarz's method

Kaczmarz's method is a geometric algorithm for approximating a solution to a linear system $A\mathbf{x} = \mathbf{p}$.

If we denote the jth row of a matrix A by $\mathbf{r}_j$, and the jth coordinate of a vector $\mathbf{p}$ by p_j, then the system $A\mathbf{x} = \mathbf{p}$ amounts to having $\mathbf{r}_j \bullet \mathbf{x} = p_j$ for every value of j. Kaczmarz's method works by producing a sequence of vectors each of which satisfies one of the individual equations $\mathbf{r}_j \bullet \mathbf{x} = p_j$.

The first research article to explore the application of algebraic reconstruction techniques to medical imaging appeared in 1970. The principal technique employed therein for the reconstruction of images turned out to be the same as Kaczmarz's method.

Affine spaces are central to the geometry of Kaczmarz's method.

Definition. For a fixed n-dimensional vector $\mathbf{r}$ and a number p, the *affine space* $\mathcal{S}_{\mathbf{r},\,p}$ is defined by

$$\mathcal{S}_{\mathbf{r},\,p} = \{\mathbf{x} \in \mathbb{R}^n \,:\, \mathbf{r} \bullet \mathbf{x} = p\}\ .$$

Example: Every line ℓ in the plane is an affine space.

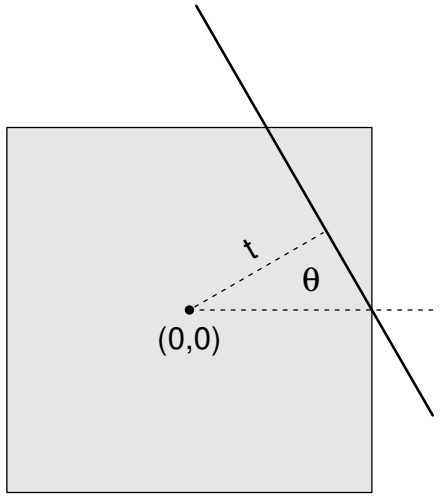


Figure 3: The line is $\mathcal{S}_{\mathbf{r},\,t}$ for $\mathbf{r} = \langle\, \cos(\theta),\, \sin(\theta)\,\rangle$.

Choose θ and t so that $\mathbf{r} = \langle\, \cos(\theta),\, \sin(\theta)\,\rangle$ is perpendicular to ℓ and $(t\cos(\theta),\, t\sin(\theta))$ lies on ℓ. Then $\ell = \{\mathbf{x} = \langle\, t\cos(\theta) - s\sin(\theta),\, t\sin(\theta) + s\cos(\theta)\,\rangle \mid -\infty < s < \infty\}$. Compute

$$\mathbf{r} \bullet \mathbf{x} = t\cos^2(\theta) - s\sin(\theta)\cos(\theta) + t\sin^2(\theta) + s\cos(\theta)\sin(\theta) = t(\cos^2(\theta) + \sin^2(\theta)) = t\,.$$

(If $t = 0$, then ℓ passes through the origin and, so, is a subspace of $\mathbb{R}^2$.)

Example: Every plane in 3-space is an affine space.

The equation of a plane has the form $ax + by + cz = d$. But this is the same as saying that the plane consists of (the terminal points of) those vectors $\mathbf{x} = \langle x, y, z \rangle$ such that $\langle a, b, c \rangle \bullet \mathbf{x} = d$.

The vector $\langle a, b, c \rangle$ is in fact perpendicular to this plane.

Geometrically, the vector $\mathbf{r}$ is orthogonal to each affine space $\mathcal{S}_{\mathbf{r}, p}$.

We can see this concretely for the two examples just discussed. More generally, this works just like the point-normal form for the equation of a plane. In other words, suppose $\mathbf{x}_0$ and $\mathbf{x}_1$ are two vectors in $\mathcal{S}_{\mathbf{r}, p}$. This really means that the terminal points of these vectors lie in a certain geometric figure (e.g., a line or a hyperplane). So, the vector $\mathbf{x}_h = \mathbf{x}_1 - \mathbf{x}_0$ lies in this geometric set. But $\mathbf{x}_h$ satisfies the homogeneous equation $\mathbf{r} \bullet \mathbf{x}_h = 0$. Thus, $\mathbf{r}$ is perpendicular to the affine space.

(Note that $\mathbf{x}_1 = \mathbf{x}_h + \mathbf{x}_0$ in the foregoing argument. It follows that every affine space can be viewed as a copy of a linear *subspace* that has been shifted by a fixed vector. This should remind us of the "general solution equals particular solution plus general homogeneous solution" formulation for a non-homogeneous linear differential equation or a non-homogeneous system of linear equations.)

Definition: Affine projection. Given a vector $\mathbf{u}$ and an affine space $\mathcal{S}_{\mathbf{r},p}$ for some vector $\mathbf{r}$ and some number p, the *affine projection* of $\mathbf{u}$ in $\mathcal{S}_{\mathbf{r},p}$ is the vector $\mathbf{u}_*$ in $\mathcal{S}_{\mathbf{r},p}$ that is *closest* to $\mathbf{u}$ amongst all vectors in $\mathcal{S}_{\mathbf{r},p}$.

Now, in order to move from $\mathbf{u}$ to the closest point in the affine space, it is evident that we should move *orthogonally* to the affine space. According to our previous observations, this means that we should move in the direction of the vector $\mathbf{r}$ itself. Thus, the vector $\mathbf{u}_*$ that we seek should have the form $\mathbf{u}_* = \mathbf{u} - \lambda\mathbf{r}$ for some number λ.

Substituting $\mathbf{u}_* = \mathbf{u} - \lambda\mathbf{r}$ into the equation $\mathbf{r} \bullet \mathbf{u}_* = p$ and solving for λ yields

$$\lambda = \frac{(\mathbf{r} \bullet \mathbf{u}) - p}{\mathbf{r} \bullet \mathbf{r}} \ .$$

Thus, the *affine projection* $\mathbf{u}_*$ of the vector $\mathbf{u}$ in the affine space $\mathcal{S}_{\mathbf{r},p}$ is given by

$$\mathbf{u}_* = \mathbf{u} - \left(\frac{(\mathbf{r} \bullet \mathbf{u}) - p}{\mathbf{r} \bullet \mathbf{r}} \right) \mathbf{r} \ . \tag{2}$$

Kaczmarz's method

Now we put our knowledge of affine spaces to work. Recall that our goal is to find an approximate solution to a linear system $A\mathbf{x} = \mathbf{p}$. Again denote the jth row of A by $\mathbf{r}_j$ and the jth coordinate of $\mathbf{p}$ by p_j. Then each of the equations $\mathbf{r}_j \bullet \mathbf{x} = p_j$ describes an affine space. Kaczmarz's method proceeds by starting with an initial guess at a solution, a vector of prospective color values, and then computing the affine projection of this initial guess in the first affine space in our list. This projection is then projected to the next affine space in the list, and so on until we have gone through the entire list of affine spaces. This constitutes one "iteration" and the result of this iteration becomes the starting point for the next iteration.

In detail, the method proceeds as follows.

algorithm for Kaczmarz's method:

(i) Select a starting "guess" for $\mathbf{x}$; call it $\mathbf{x}^0$.

(ii) Next set $\mathbf{x}^{0,0} = \mathbf{x}^0$.

(iii) The inductive step is this: Once the vector $\mathbf{x}^{0,j-1}$ has been determined, define

$$\mathbf{x}^{0,j} = \mathbf{x}^{0,j-1} - \left(\frac{\mathbf{x}^{0,j-1} \bullet \mathbf{r}_j - p_j}{\mathbf{r}_j \bullet \mathbf{r}_j} \right) \mathbf{r}_j \,. \tag{3}$$

We have used the affine projection formula (2).

(iv) Note that if the matrix A has J rows, then the vectors $\mathbf{x}^{0,1}$, $\mathbf{x}^{0,2}$, ..., $\mathbf{x}^{0,J}$ will be computed.

(v) Once $\mathbf{x}^{0,J}$ has been computed, define $\mathbf{x}^1 = \mathbf{x}^{0,J}$ and begin the process again starting with $\mathbf{x}^1$. That is, now set $\mathbf{x}^{1,0} = \mathbf{x}^1$ and compute the vectors $\mathbf{x}^{1,1}$, $\mathbf{x}^{1,2}$, ..., $\mathbf{x}^{1,J}$, as in (3).

(vi) Then let $\mathbf{x}^2 = \mathbf{x}^{1,J}$ and repeat the process starting with $\mathbf{x}^{2,0} = \mathbf{x}^2$.

(vii) Stop when we've had enough!

Example. Apply Kaczmarz's method to the system consisting of just the two lines $x + 2y = 5$ and $x - y = 1$. So $\mathbf{r}_1 = \langle\, 1,\, 2 \,\rangle$, $\mathbf{r}_2 = \langle\, 1,\, -1 \,\rangle$, $p_1 = 5$, and $p_2 = 1$. With the initial guess $\mathbf{x}^0 = \langle\, 0.5,\, 0.5 \,\rangle$, the diagram on the left in Figure 4 shows that the solution $\langle\, 7/3,\, 4/3 \,\rangle$ is quickly found.

However, when we include the third line $4x + y = 6$ (so $\mathbf{r}_3 = \langle\, 4,\, 1 \,\rangle$ and $p_3 = 6$), then the successive iterations settle into a triangle pattern, shown in the diagram on the right in the figure.

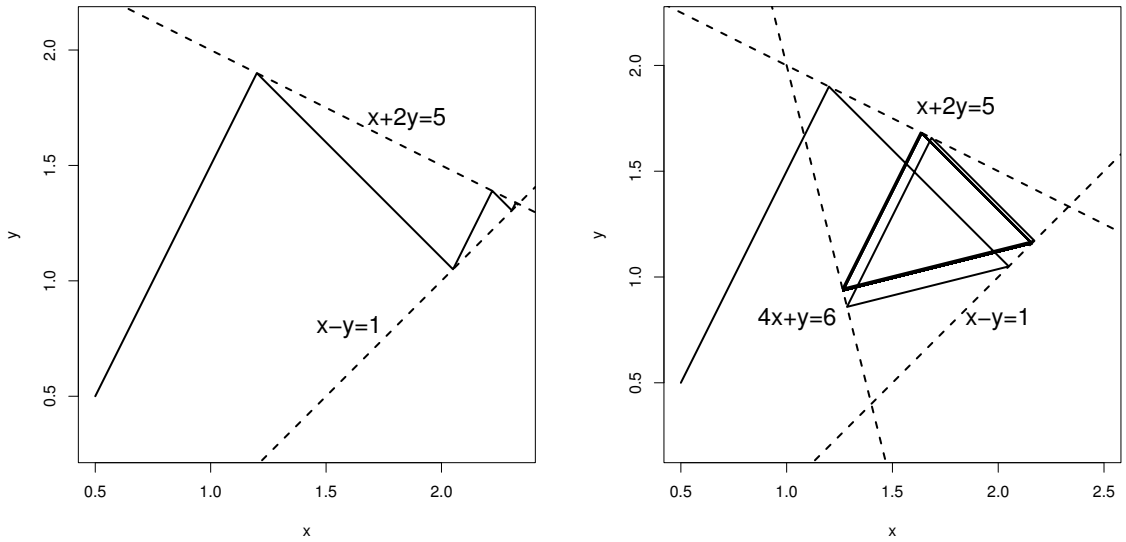


Figure 4: On the left, Kaczmarz's method converges quickly to the point of intersection of two lines. On the right, the successive iterations settle into a triangular pattern for a system of three lines.

The successive vectors $\mathbf{x}^0$, $\mathbf{x}^1$, $\mathbf{x}^2$, ... should get closer to a vector that satisfies the original system $A\mathbf{x} = \mathbf{p}$. However, the convergence may be quite slow; many cycles would be needed to get a good approximant. If the system has no solution, then the vectors might settle into a specific pattern or even exhibit "chaotic" behavior.

In an implementation of Kaczmarz's method, one can choose to save only the last estimate of the solution, after all cycles have been completed. Alternatively, one can generate a matrix where each successive row is an intermediate estimate, either from a single equation or from a full cycle. We present both ideas here.

```
##Kaczmarz method: save next estimate from each eqn.
kacz.v0=function(mat,rhs,ncycles){
neq=nrow(mat)
V=matrix(double((neq*ncycles+1)*ncol(mat)),
neq*ncycles+1,ncol(mat))
V[1,]=rep(0.5,2)
for (i in 1:ncycles){
for (j in 1:neq){
V[neq*(i-1)+j+1,]=ifelse(mat[j,]==0,V[neq*(i-1)+j,],
V[neq*(i-1)+j,]-((sum(V[neq*(i-1)+j,]*mat[j,])-
rhs[j])/sum(mat[j,]*mat[j,]))*mat[j,])}}
list(V=V)}
##For Figure 09_1
R1=matrix(c(1,2,1,-1),2,2,byrow=T)#2 lines
p1=c(5,1)
V1=kacz.v0(R1,p1,2)$V
R2=matrix(c(1,2,1,-1,4,1),3,2,byrow=T)#3 lines
p2=c(5,1,6)
V2=kacz.v0(R2,p2,3)$V
```

```
#plot fig09_1a
plot(V1[,1],V1[,2],type="l",lwd=2,xlab="x",ylab="y")
plot(V2[,1],V2[,2],type="l",asp=1,xlab="x",ylab="y",lwd=2)


##Kaczmarz's method; save only most recent estimate
kaczmarz.basic=function(mat,rhs,numcycles){
numeq=nrow(mat)
sol.est=rep(0.5,ncol(mat))#initial "guess" of all 0.5s
for (i in 1:numcycles){
for (j in 1:numeq){
sol.est=ifelse(mat[j,]==0,sol.est,sol.est-
((sum(sol.est*mat[j,])-rhs[j])/sum(mat[j,]*mat[j,]))*mat[j,])
}}list(sol.est=sol.est)}
```

Before looking at an image created using Kaczmarz's method, we state, without proof, the main convergence theorem for this algorithm. (For a proof see Natterer's book.)

Theorem. If the linear system $A\mathbf{x} = \mathbf{p}$ has at least one solution, then Kaczmarz's method converges to a solution of this system. Moreover, if $\mathbf{x}^0$ is in the range of A^T, then Kaczmarz's method converges to the solution of minimum norm.

This convergence result is generally not relevant in medical imaging, where the linear systems encountered tend to be indeterminate and where, in any case, it is computationally feasible to run only a few iterations of Kazcmarz's method.

ART in medical imaging

Phantoms One method for testing the accuracy of a particular image reconstruction algorithm is simply to apply the algorithm to data taken from an actual human subject. The drawback of this is that usually we don't know exactly what we ought to see in the reconstructed image. So there is no way to determine the accuracy of any particular image.

To get around this, we can apply algorithms to data taken from a physical object whose internal structure is known. That way, we know what the reconstructed image ought to look like. However, there may be errors in the data collection. In turn, these errors may lead to errors in the reconstructed image. We will not be able to distinguish these flaws from errors caused by the algorithm itself.

To resolve this dilemma, Shepp and Logan introduced the concept of a *mathematical phantom* — a simulated test subject whose structure is defined mathematically. Thus, *no errors occur in collecting the data.* When an algorithm is applied to produce a reconstructed image of the phantom, *all inaccuracies are due to the algorithm.* This makes it possible to judge the quality of an algorithm.

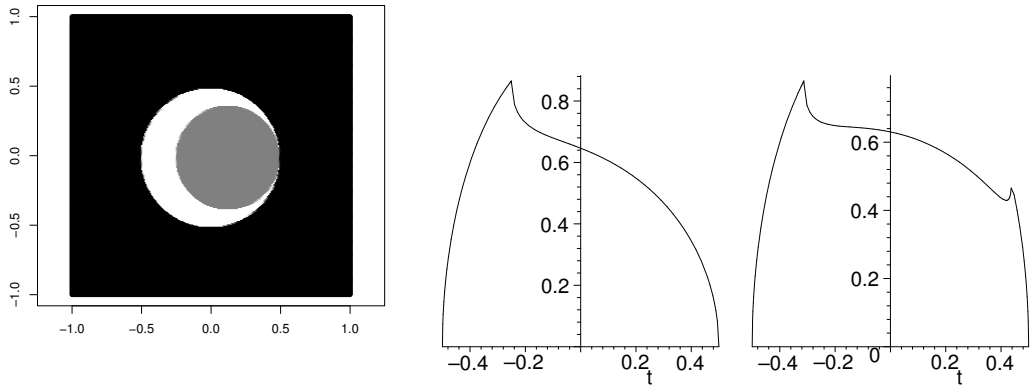


Figure 5: The figure shows the graph of a crescent-shaped phantom alongside graphs of its Radon transform at angles 0 and $\pi/3$.

The following images are reconstructions of the crescent-shaped phantom illustrated above. For each image, five full iterations of Kaczmarz's method were applied with an initial vector $\mathbf{x}^0$ having every coordinate equal to 0.5. (In other words, the starting point was taken to be a neutral grey image.)

The first image uses a 40×40 grid. The phantom was sampled using an angle increment of $\pi/60$ with 41 X-rays at each angle, for a total of 2460 X-rays. The corresponding system of equations is over-determined.

The second image has 10000 pixels and a total of 9090 X-ray beams (101 beams at each of 90 angles). This system is under-determined.

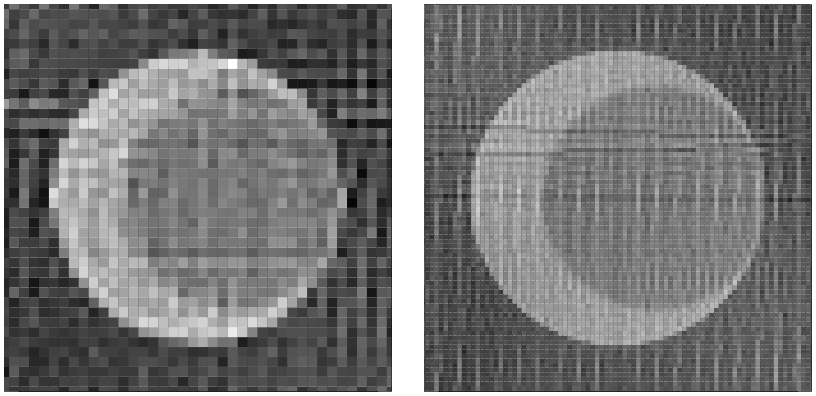
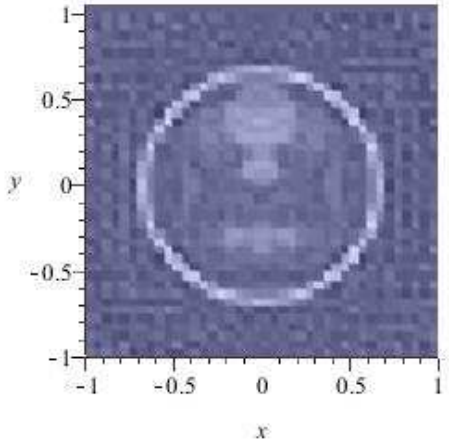


Figure 6: Kaczmarz's method applied to a crescent-shaped phantom: On the left, a 40×40 image grid and 2460 values of the Radon transform; on the right, a 100×100 image grid and 9090 values of the Radon transform.

Two more examples: (i) 40×40 grid; 1200 X-ray beams.



(ii) 100×100 grid; 9090 X-ray beams.

Some computational concerns:

- *The sheer size of the system of equations:* Each value r_{jk} represents the length of the intersection of one of the X-ray beams with one of the pixel squares in the image grid. Most of these values are 0s. The formula for computing $\mathbf{x}^{k,j}$ from $\mathbf{x}^{k,j-1}$ only alters $\mathbf{x}^{k,j-1}$ in those components that correspond to the pixels through which the jth beam passes. To streamline the computation of $\mathbf{x}^{k,j-1} \bullet \mathbf{r}_j$, we could store the locations of the nonzero entries of $\mathbf{r}_j$.

- *The rate of convergence:* Adjacent X-ray beams transmitted in the same direction will intersect many of the same pixels. Thus, the corresponding affine spaces $\{\mathbf{x} \bullet \mathbf{r} = p\}$ will be almost parallel. This might tend to slow down the convergence of the algorithm and increase the number of iterations required to get an acceptable image. One way to address this is to use more and smaller pixels, though this adds to the processing time.

Variations of Kaczmarz's method

Perhaps the most commonly employed variation of Kaczmarz's method involves the introduction of so-called *relaxation parameters* in the crucial step of the algorithm. Specifically, for each j and k, let λ_{jk} satisfy $0 < \lambda_{jk} < 2$ and take

$$\mathbf{x}^{k,\,j} = \mathbf{x}^{k,\,j-1} - \lambda_{jk} \cdot \left(\frac{\mathbf{x}^{k,\,j-1} \bullet \mathbf{r}_j - p_j}{\mathbf{r}_j \bullet \mathbf{r}_j} \right) \mathbf{r}_j \,. \tag{4}$$

For $\lambda_{jk} = 1$ this is the same as before. For $0 < \lambda_{jk} < 1$, the vector $\mathbf{x}^{k,\,j-1}$ is projected only part of the way to the affine space $\mathcal{S}_{\mathbf{r}_j,\,p_j}$. When $1 < \lambda_{jk} < 2$, the vector $\mathbf{x}^{k,\,j-1}$ is projected to the other side of $\mathcal{S}_{\mathbf{r}_j,\,p_j}$. Note that, if $\lambda_{jk} = 2$, then the vector $\mathbf{x}^{k,\,j}$ is just the reflection of $\mathbf{x}^{k,\,j-1}$ across $\mathcal{S}_{\mathbf{r}_j,\,p_j}$ and there is no improvement in the proximity to a solution. This is the reason for the restriction $\lambda_{jk} < 2$. In fact, the usual requirement is that the value of λ_{jk} be bounded away from 0 and 2; that is, there should be numbers α and β such that $0 < \alpha \leq \lambda_{jk} \leq \beta < 2$ for all j and k.

The additional control over the projection offered by the relaxation parameters can be used to facilitate finding an acceptable approximate solution to an indeterminate system.

Using R: Relaxation parameters can be introduced into the computer implementation of Kaczmarz's method with a simple modification to the procedure in **??**. For instance, the relaxation parameter can be chosen randomly at each step, like so.

```
#Kaczmarz's method with relaxation
kaczmarz.relax=function(mat,rhs,numcycles){
numeq=nrow(mat)
solB=rep(0.5,ncol(mat))
for (j in 1:numcycles){
for (i in 1:numeq){
solB=ifelse(mat[i,]==0,solB,solB-runif(1,min=0.5,max=1.5)*
((sum(solB*mat[i,])-rhs[i])/sum(mat[i,]*mat[i,]))*mat[i,])}}
list(solB=solB)}
```

Then continue as before to compute the coefficient matrix and so on.

Alternatively, we can replace the original system of equations with a system of inequalities, like so. For each j, select a (small) positive number ε_j and consider the inequalities

$$p_j - \varepsilon_j \leq \mathbf{r}_j \bullet \mathbf{x} \leq p_j + \varepsilon_j \ .$$

A solution to this *feasibility problem* is a vector $\mathbf{x}^*$ that lies in close proximity to some collection of affine spaces, instead of lying at their intersection. Geometrically, if we think of an affine space as a higher-dimensional plane sitting inside a many-dimensional space, then vectors in proximity to an affine space form a higher-dimensional "slab," with thickness $2 \cdot \varepsilon_j$. The solution vector $\mathbf{x}^*$ would lie inside these slabs.

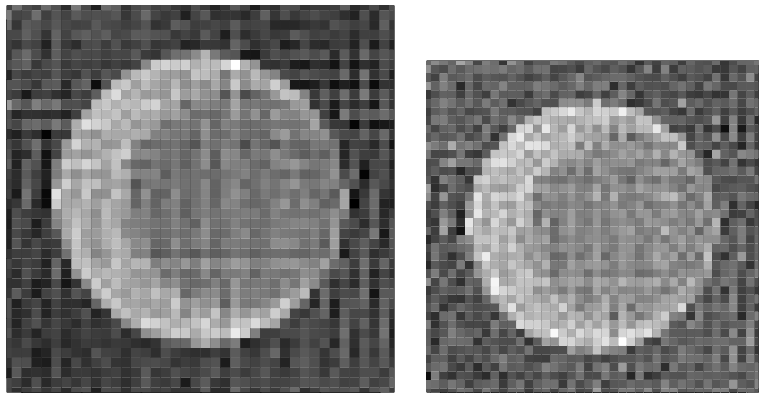


Figure 7: Variations of Kaczmarz's method: On the left, the relaxation parameter at each step is a number between 0.5 and 1.5 chosen randomly from a uniform distribution; on the right, random Gaussian noise has been added to the values of the Radon transform.

The articles by Censor and by Gordon, Bender, and Herman offer more details of these methods. A general approach to feasibility problems that includes Kaczmarz's method and its variants can be found in the comprehensive article by Bauschke and Borwein.

References

1. Bauschke, H. H., and J. M. Borwein, On projection algorithms for solving convex feasibility problems, *SIAM Review* **38** (1996), 367–426.
2. Censor, Y., Row-action methods for huge and sparse systems and their applications, *SIAM Review* **23** (1981), 444–464.
3. Censor, Y., Finite series-expansion reconstruction methods, *Proc. IEEE* **71** (1983), 409–419.
4. Cormack, A. M., Representation of a function by its line integrals, with some radiological applications I, II, *J. Appl. Phys.* **34** (1963), 2722–2727; **35** (1964), 2908–2912.
5. Dewdney, A. K., How to resurrect a cat from its grin, in Mathematical Recreations, *Scientific American* **263**(3) (September 1990), 174–177.
6. Gordon, R., R. Bender, and G. T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, *J. Theoretical Biology* **29** (1970), 471–481.
7. Hounsfield, G. N., Computerized transverse axial scanning tomography, *British J. Radiology* **46** (1973), 1016–1022.
8. Hounsfield, G. N., A method of and apparatus for examination of a body by radiation such as X or gamma radiation, The Patent Office, London, (1972) Patent Specification 1283915.
9. Natterer, F., *The Mathematics of Computerized Tomography*, Classics in Applied Mathematics **32**, SIAM, Philadelphia, 2001.
10. Shepp, L. A., and B. F. Logan, The Fourier reconstruction of a head section, *IEEE Trans. Nucl. Sci.* **NS-21** (1974), 21–43.