Physica Scripta. Vol. T61, 38–43, 1996

Mathematics for Computer Tomography

Bertil Gustafsson

Uppsala University, Department of Scientific Computing, Box 120, S-751 04 UPPSALA, Sweden

Received August 14, 1995

Abstract

Computerized tomography requires not only fast computers, but also analysis of mathematical models and construction of numerical algorithms. Classical mathematical theory is combined with modern numerical analysis to form the basis for efficient implementation on fast computers.

The solution of the inverse problem of finding the image from given X-ray projections is theoretically obtained by the inverse Radon transform. Since only a finite number of projections are available, some approximation must be found, and this leads to a discrete counterpart of the continuous problem. There are three major classes of numerical solution methods: the Algebraic Reconstruction Method, the Filtered Back projection Method and the Direct Fourier Method. Much research is devoted to making the methods faster and more robust. The first one was used for the original tomography machine, the second one is used on almost all current machines in use. The third one has great potential for the future, since almost all computation is done by using the fast discrete Fourier transform.

We shall describe the basic mathematical problem in computer tomography and the computational methods mentioned above for solving it. In particular we shall emphasize the special difficulties that are built into the problem. However, this is not a review article. Instead, it is intended to describe the influence of modern numerical methods on a fundamental problem of great significance for the society. We shall also indicate how computerized tomography has initiated new important research in central fields of numerical analysis, that can be used for problems in many other applications.

1. Introduction

The use of computational methods have become a very important tool in science and modern engineering with significant effects on the society. Perhaps the computerized tomography is one of the top candidates for the most spectacular use of computational methods. It has lead to a breakthrough for X-ray diagnosis leading to dramatically improved possibilities for medical treatment of severe illness. Here the computer uses fast algorithms to produce the X-ray images in everyday clinical work. Computer tomography has also many other applications in science and engineering, nondestructive material testing being one of them.

Computational methods require fast and reliable computers, and the development of such machines has sky-rocketed during the last decades. But the fundamental basis behind computation is mathematics and numerical analysis. Computational tomography is one of the best examples where these two disciplines have interacted in a most fruitful way to play a very important role. Classical mathematical theory developed long time ago was combined with modern numerical analysis to form the foundation for the first computerized tomography system. It was patented in 1972 and lead to the Nobel Prize in medicine 1979. One of the constructors was G. N. Hounsfield, who was responsible for the mathematical/numerical part of the system. There is no Nobel Prize in mathematics, but this one in medicine shows that mathematics is also appreciated, at least in its applied form.

In this paper we shall give a brief description of the basic mathematical problem and the numerical methods that are used for reconstructing the image. We shall not describe in detail the various improved methods and tricks that are being worked on today, but rather describe the key problems. Furthermore we shall describe some of the characteristics of modern numerical methods that come into play for the problem at hand.

We shall restrict ourselves to the basic two-dimensional problem with parallel X-rays. For more details about this case, as well as about the more general case including fan-beam projections and three-dimensional tomography, we refer to the literature, for example the book of Herman [7] and of Natterer [15]. In the bibliography, further books and articles are given [1, 2, 4, 8–14] without special reference in the presentation that follows.

2. The reconstruction problem

When an X-ray beam passes through matter, the energy is attenuated. Higher density gives a stronger attenuation, and this is the basis for all X-ray diagnoses. Let $f(s)$ denote the density along the straight line L between the source and the detector as shown in Figure 1.

If the energy at the source is E_0, the energy E_1 at the detector is

$$E_1 = E_0 e^{-\int_L f(s)\mathrm{d}s}.$$

By taking the logarithm of both sides, we get

$$\int_L f(s)\mathrm{d}s = \ln \frac{E_0}{E_1}. \tag{1}$$

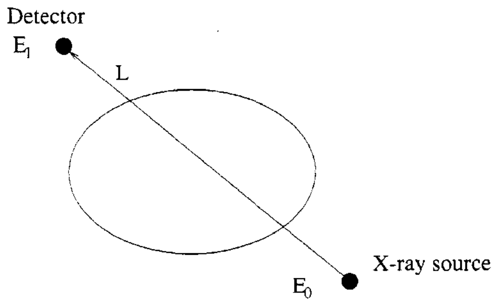

Fig. 1. X-ray projection.

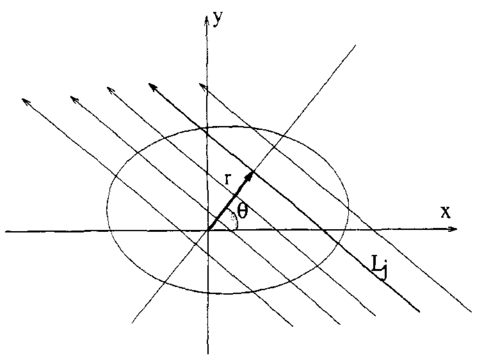

Fig. 2. Polar coordinate system r, θ.

The value $p = \ln(E_0/E_1)$ is measured in the detector (after calibration). The fundamental problem is that we would like to determine $f(s)$, but we know only p. There are infinitely many functions $f(s)$ that satisfies the equation (1). More information is needed.

Considering the two-dimensional problem and the density function $f(x, y)$ in the plane, one X-ray beam gives rise to one equation

$$\int_{L_j} f(x, y)\mathrm{d}s = p_j,$$

where $\int_{L_j}$ denotes the line integral along the line L_j. If sufficiently many such integrals along different lines L_j are known, it seems reasonable to assume that there is enough information to obtain at least a good approximation of $f(x, y)$. The question is how this approximation should be obtained. This is what image reconstruction is about.

We shall only discuss the case with parallel beams, and introduce a polar coordinate system (r, θ) according to Figure 2.

For a given angle θ and a given distance r from the origin, the measured value for the corresponding beam is the line integral of $f(x, y)$ along the beam path. By introducing the Dirac delta-function $\delta(x)$, we can formally write

$$p(r, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - r)\mathrm{d}x\mathrm{d}y.$$

The right hand side is the Radon transform $[\mathcal{R}f]$ (r, θ) of f, and it represents all line integrals of f. Assuming that $p(r, \theta)$ is known, we would like to find

$$f(x, y) = [\mathcal{R}^{-1}p](x, y),$$

where $\mathcal{R}^{-1}$ is the inverse Radon transform. The solution of this problem was given by Radon 1917 [16], and it can be written as

$$f(x, y) = \frac{1}{2\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} \frac{\dfrac{\partial p}{\partial r}}{x\cos\theta + y\sin\theta - r}\,\mathrm{d}r\mathrm{d}\theta. \tag{2}$$

This is an important formula, but unfortunately it doesn't give us the solution in a practical situation. We don't have an infinite number of measurements at our disposal, i.e. $p(r, \theta)$ is known only at a finite number of points. And even if we had, it wouldn't be possible to give an explicit computable expression for the integral. But we can hope that a good approximation of $f(x, y)$ can be obtained by *discretizing* the integrals. This is where numerical analysis comes in.

We shall briefly describe the three most common methods for computing the solution. All of them represent a finite number of operations on p that can be seen as a discrete approximation of $\mathcal{R}^{-1}$.

There is actually still another basic difficulty associated with the problem and the operator $\mathcal{R}^{-1}$. The measured projections p are never exact, and we must expect errors in the solution. However, as for most other problems in mathematics, one expects that small errors in the data produces only small errors in the solution. If that is the case, one calls the problem *well posed*, or *well conditioned*. Unfortunately, in our case this is not true. One can prove that small errors δp in the data may produce large errors in the solution f. Such a problem is called *ill posed* or *ill conditioned*. We shall not give the exact mathematical formulation of this property here, it requires some extra mathematical concepts that are beyond the scope of this article. However, we must be aware, that the numerical approximation is bound to have the same type of ill-conditioning built in, at least if the number of pixels is large, such that the discrete problem is close to the continuous one. In practice the errors in the solution have an oscillatory character, and it is possible to design methods which have a smoothing property for this type of functions.

Before going into the next section, we shall discuss the importance of finding an algorithm with low operation count. Assume that we have an image represented by $N \times N$ pixels. For most machines of today, $N = 512$. As will be shown in the next section, one solution method leads to a linear system of equations for all the N^2 unknown f-values, which we order in a vector f. In the real situation we have more measurements than unknowns, but for the sake of simplicity we assume here that they are equal in number. Then we have a system

$$\mathbf{A}f = p,$$

where $\mathbf{A}$ is a very large square matrix. Already Gauss knew how to solve such a system. But his method, which is still used for small systems, requires $N^6/3$ arithmetic operations (a.o.). But $512^6/3 \approx 6 \cdot 10^{15}$, and even with a supercomputer, this leads to computing times where the proper time unit is years. Furthermore, the memory requirement is enormous. We shall see, that with modern methods, the operation count can be reduced to $\mathcal{O}(N^3)$, which gives reasonable computing times. However, in the search for machines that give the answers in seconds, such that it can be used in real time, one would like to reduce the computing time further. If a certain algorithm of today gives the answer in 5 minutes, a reduction by a factor N would take the time down to half a second. For three-dimensional problems, super-fast algorithms are of course even more important.

3. The algebraic reconstruction method (ARM)

In this section we shall denote the number of pixels by J, where in most cases $J = N^2$. The number of projections is denoted by I, where in general $I > J$. The mathematical formulation is based on the representation of the image $f(x, y)$ in terms of the *basis images* $\{b_j(x, y)\}_{j=1}^{J}$ defined by

$$b_j(x, y) = \begin{cases} 1 \text{ if } (x, y) \in \text{pixel no. } j \\ 0 \text{ else.} \end{cases}$$

The image is written as

$$f(x, y) = \sum_{j=1}^{J} c_j b_j(x, y),$$

where the coefficients c_j are to be determined. The projections $\{p_i\}_{i=1}^{I}$ are known, and if R_i is the projection operator defined by

$$R_if = \int_{L_i} f(x, y)\mathrm{d}s$$

along the line L_i, we get the system

$$R_if = \sum_{j=1}^{J} c_j R_i b_j(x, y) = p_i, \ i = 1, 2, \ldots, I.$$

Since $b_j(x, y)$ are known, the numbers

$$a_{ij} = R_i b_j(x, y)$$

can be computed once and for all, and form the rectangular matrix $\mathbf{A} = \{a_{ij}\}$. Since $b_j(x, y)$ is zero almost everywhere, most of the elements a_{ij} are zero, leading to a sparse matrix $\mathbf{A}$. With $\boldsymbol{c}$ and $\boldsymbol{p}$ denoting the vectors with elements c_j and p_i respectively, we have a linear system of equations

$$A\boldsymbol{c} = \boldsymbol{p}.$$

In general there is no solution of this system, since there are more equations than there are unknowns. Hence, the best we can hope for, is an approximation of the solution. This is a very common situation in many applications, and the usual procedure is to find the least-square solution. With the norm of an I-element vector v defined by

$$\|v\| = \left(\sum_{i=1}^{I} v_i^2\right)^{1/2},$$

we solve the optimization problem

$$\min_{\boldsymbol{c}} \| \mathbf{A}\boldsymbol{c} - \boldsymbol{p} \| .$$

The solution is theoretically given by a new linear system of equations

$$\mathbf{A}^{\mathrm{T}} \mathbf{A}\boldsymbol{c} = \mathbf{A}^{\mathrm{T}} \boldsymbol{p}, \tag{3}$$

which has a square coefficient matrix. However, even if we now have a unique solution, there is still an inherent difficulty with this approach. It was mentioned in Section 2 that the original problem is ill posed. This fact, together with the properties of the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, makes the problem (3) ill posed. This means that a small perturbation of $\boldsymbol{p}$ may cause a big perturbation of the solution $\boldsymbol{c}$. The next step is therefore to modify the problem once more, such that we get a more robust solution method. This is called *regularization.*

One form of regularization is to reformulate the optimization problem into

$$\min_{\boldsymbol{c}} (\| \mathbf{A}\boldsymbol{c} - \boldsymbol{p} \|^2 + \gamma \| \boldsymbol{c} \|^2), \tag{4}$$

where γ is a constant. The choice of this parameter γ is a crucial part of this approach.

Another form of regularization can be described for any linear system of equations that is ill conditioned. Given the system

$$\mathbf{Q}\boldsymbol{x} = \boldsymbol{b} \tag{5}$$

(with change of notation) we consider *iterative methods* for solving it. Iterative methods are particularly efficient when the matrix $\mathbf{Q}$ is sparse. A simple but common form for such methods is

$$\boldsymbol{x}^{(j+1)} = \mathbf{B}\boldsymbol{x}^{(j)} + \mathbf{C}\boldsymbol{d}, \ j = 0, 1, \ldots, \tag{6}$$

with $\boldsymbol{x}^{(0)}$ being the initial guess. If the eigenvalues of $\mathbf{B}$ are less than one in magnitude, and

$$(\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}\boldsymbol{d} = \mathbf{Q}^{-1}\boldsymbol{b}, \tag{7}$$

then the solution $\boldsymbol{x}^{(j)}$ converges to the true solution as $j \to \infty$. If the iterations are carried out to convergence, the ill-conditioning is of course still present. However, by making the reasonable assumption that $\mathbf{B}$ and $\mathbf{C}$ are matrices with a norm that is bounded and not very large, the iterates $\boldsymbol{x}^{(j)}$ cannot be sensitive to perturbations for small j. For example, if a perturbation $\delta\boldsymbol{d}$ is introduced, the corresponding error $\delta\boldsymbol{x}^{(1)}$ can be estimated by

$$\| \delta\boldsymbol{x}^{(1)} \| \le \| \mathbf{C} \| \cdot \| \delta\boldsymbol{d} \| .$$

By inserting this in the iteration formula for $\boldsymbol{x}^{(2)}$, we get

$$\| \delta\boldsymbol{x}^{(2)} \| \le (\| \mathbf{B} \| + 1) \| \mathbf{C} \| \cdot \| \delta\boldsymbol{d} \| .$$

This procedure can be continued for increasing j. The error constant multiplying $\| \delta\boldsymbol{d} \|$ will be a polynomial in $\| \mathbf{B}\|$ and $\| \mathbf{C}\|$ of degree j. Thus, as long as j is not too large, the error will be of reasonable size. On the other hand, in order to be near the true solution to the unperturbed problem, a sufficient number of iterations must be carried out. A typical error curve looks like the one shown in Figure 3.

The goal of the analysis for these problems is to find the value j_0 that gives the smallest error, and much research in numerical analysis of today is devoted to this.

The optimization problem (4) is a special case of more general forms. The expression to be minimized can be more general and there may be various forms of side conditions. This includes *entropy maximization* methods and also *maximum likelihood* methods, where the formulation is derived from a probability theoretical approach. A common feature for all these methods based on the a-priori discretization of the picture, is that the final algebraic criterion can be formulated as a large linear system of equations with a sparse coefficient matrix. These systems are usually solved by iterative methods, and we shall discuss such methods a little more from a general point of view.

When constructing iterative methods, the primary goal is to obtain convergence to the solution, or rather to an approximate solution, as fast as possible. For one-step methods

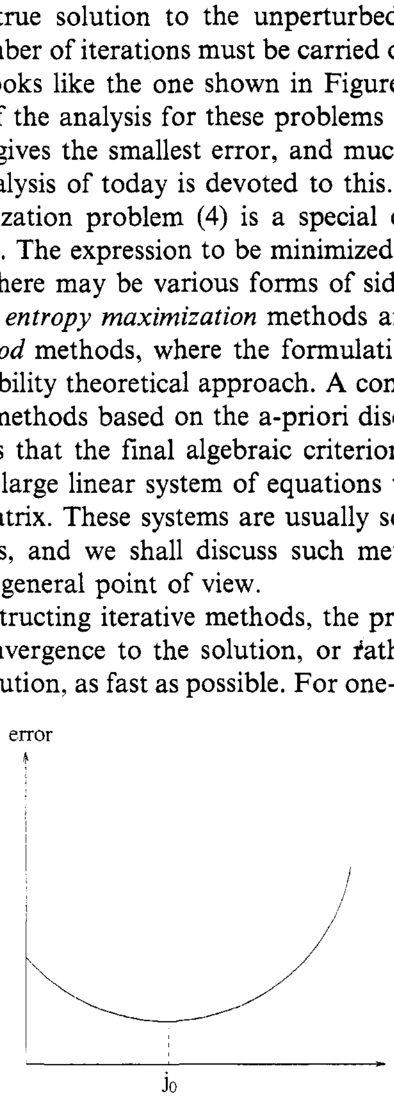

Fig. 3. Error as a function of iteration number.

of type (6), the convergence is determined by the size of the eigenvalues of **B**. Smaller magnitude gives faster convergence. However, the consistency requirement (7) introduces certain restrictions, that often forces some of the eigenvalues to be located close to the unit circle.

The most popular iterative method today is the conjugate gradient method, where each iteration is dependent on all previous values $\boldsymbol{x}^{(j)}$. We refer to the book by Golub-van Loan [5] for the exact algorithm. Also for this method, the location of the eigenvalues of the matrix **Q** in (5) determines the convergence rate. Under certain conditions one can prove that convergence to the true solution is obtained in m steps, where the eigenvalues are clustered at m points. Usually m is very large, and one needs to work with a different matrix. This leads to *preconditioned iterative methods*. The idea is to multiply the system (5) by a suitable matrix:

$$\mathbf{D}^{-1}\mathbf{Q}\boldsymbol{x} = \mathbf{D}^{-1}\boldsymbol{b}. \tag{8}$$

The conjugate gradient method is then applied to the new preconditioned system.

There are two requirements on the new system.

1. A system $\mathbf{D}\boldsymbol{y} = \boldsymbol{d}$ must be easy to solve.
2. $\mathbf{D}^{-1}\mathbf{Q}$ has clustered eigenvalues.

The first condition is to ensure that each step in the algorithm is fast. The second one ensures that few iterations are required for convergence. The ideal case would be if we could get all eigenvalues clustered at one point. However, that would require that we can invert **Q**, which is exactly the difficult problem we want to solve. Instead we look for a matrix **D** that is close to **Q**, but easy to invert. Then there is good hope that the eigenvalues are clustered, and furthermore, that they are contained in a small part of the complex plane.

An example (not immediately applicable to the reconstruction problem, see [6]) is given in Figure 4, where the spectrum of a certain matrix **Q** is shown.

The matrix has the order $N^2 \times N^2$, $N = 32$, and the N^2 eigenvalues are all different and widely spread. By modifying the matrix slightly, it takes a form where the system $\mathbf{D}\boldsymbol{y} = \boldsymbol{d}$ can be solved in $\mathcal{O}(N^2)\log_2 N$ a.o.. The spectrum of the preconditioned matrix $\mathbf{D}^{-1}\mathbf{Q}$ is shown in Figure 5.

The eigenvalues are now collected in 32 clusters, i.e. we have gained a factor N. The complete algorithm now takes $\mathcal{O}(N)$ iterations, i.e. $\mathcal{O}(N^3 \log_2 N)$ a.o., which even for larger N leads to acceptable computing times.

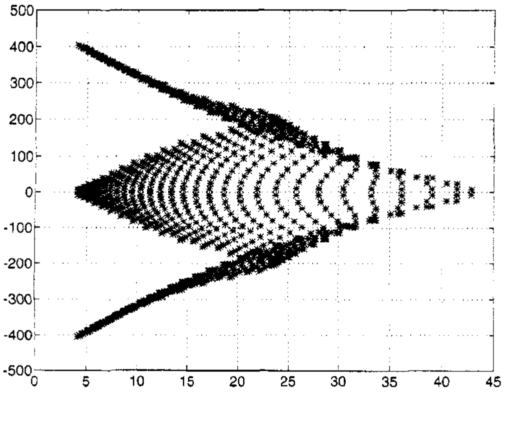

Fig. 4. Eigenvalues of original matrix.

The search for good preconditioners and iterative methods is one of the research directions for the tomography problem. If it will be successful, taking also the regularization into account, the ARM may well make a comeback into commercial machines.

4. The filtered back-projection method (FBM)

Let us again consider Figure 2 and the projections

$$p(r, \theta) = \int_L f(x, y)\mathrm{d}s, \tag{9}$$

where L is uniquely defined by the coordinates r, θ. We consider p as known for all r, θ. Define the Fourier transform of $p(r, \theta)$ with respect to r by

$$\hat{p}\,(\omega, \theta) = \int_{-\infty}^{\infty} p(r, \theta)e^{-i2\pi\omega r}\mathrm{d}r, \tag{10}$$

where ω is the Fourier variable. A straightforward calculation using (9), (10) shows that the image $f(x, y)$ can be expressed as

$$f(x, y) = \int_0^{\pi}\int_{-\infty}^{\infty} \hat{p}\,(\omega, \theta)|\omega|e^{i2\pi\omega r}\mathrm{d}\omega\mathrm{d}\theta,$$

where $r = x\cos\theta + y\sin\theta$. Based on this expression, the image is obtained in four steps.

1. Fourier transform of p with respect to r.
2. Multiplication by $|\,\omega\,|$.
3. Inverse Fourier transform of $\hat{p}\,|\,\omega\,|$.
4. The integral with respect to θ.

Since $p(r, \theta)$ is known only at discrete points (r_i, θ_j), we use the discrete Fourier transform and its inverse. Given any sequence of numbers $\{u_j\}_{j=0}^{N-1}$, it is given by

$$\hat{u}_\omega = \frac{1}{N}\sum_{j=0}^{N-1} e^{-i2\pi\omega j/N}u_j,$$

$$u_j = \frac{1}{N}\sum_{\omega=0}^{N-1} e^{i2\pi\omega j/N}\hat{u}_\omega.$$

A direct evaluation of these sums term by term takes $8N^2$ real a.o., which for large N takes unrealistic computing time. However, with the Fast Fourier Transform (FFT) where the algebraic operations are combined in a different way, the operation count is reduced to $5N\log_2 N$ a.o.. The introduction of the FFT by Cooley and Tukey 1965 [3] is probably the one algorithm that has had the greatest impact on numerical computation ever. In almost all areas there are problems where the discrete Fourier transform is a central tool, and for typical applications, the computing time goes down from hours to seconds by using the FFT.

Let us now take a closer look at the FBM algorithm. Consider the angle $\theta = \theta_j$ as fixed, and assume that there are $M = \mathcal{O}(N)$ discrete parallel projections $p(r_i, \theta)$ for each fixed θ. Step 1 and 3 takes $\mathcal{O}(M\log_2 M)$ a.o., step 2 takes $\mathcal{O}(M)$ a.o.. The integral in step 4, often referred to as back-projection, requires a numerical approximation. Let x, y be the Cartesian coordinates, and $q(r, \theta)$ the result of step 3. If $\Delta\theta$ denotes the difference in angle between different projections, then

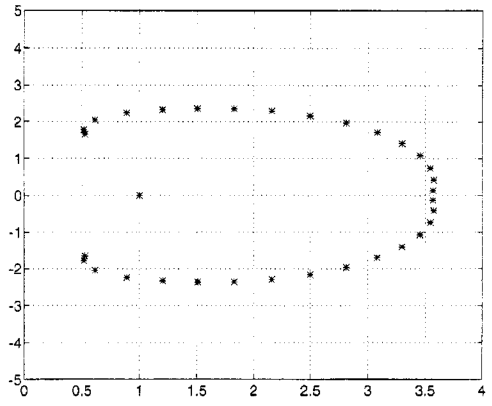

Fig. 5. Eigenvalues of preconditioned matrix.

$$f(x, y) = \int_0^\pi q(x \cos \theta + y \sin \theta, \theta) d\theta \approx \frac{\pi}{\Delta\theta} \sum_j q(x \cos \theta j + y \sin \theta_j, \theta_j),$$

where the integral has been approximated by a Riemann sum. Consider a pixel located at (x, y), and draw a line L_i through (x, y) perpendicular to the r-coordinate line. The line will pass between r_i and r_{i+1}, and we get the required value $q(\tilde{r}_i, \theta_j)$ by interpolation, see Figure 6.

The image $f(x, y)$ gets the contribution

$$\frac{\pi}{\Delta\theta} q(\tilde{r}_i, \theta_j)$$

for each angle θ_j. Obviously, all points along L_i will get the same contribution, and this computation requires $O(N)$ a.o. for each line L_i. In all we need to do $O(N^2)$ a.o. for each θ_j. We have $O(N)$ angles θ_j, hence, the total operation count is $O(N^3)$ for step 4. This is the dominating part of the algorithm, since the first 3 steps take only $O(N^2 \log_2 N)$ a.o..

However, there is one advantage with FBM when compared to ARM. Once the first set of projections at $\theta = \theta_1$ is complete, the computing cycle 1–4 above can begin, and it can be completed independently of the next set of projections at $\theta = \theta_2$. Let t_p be the time for the physical process of obtaining one set of projections, and let t_c be the time for the corresponding computing cycle. Then the image is complete at time

$$T \approx N \max(t_p, t_c).$$

If $t_c \leq t_p$, no other algorithm can beat FBM. Presently t_c is not small enough to fulfill this condition. If the speed of computers will increase at a faster rate than the speed of X-ray machines, then the condition will finally be fulfilled.

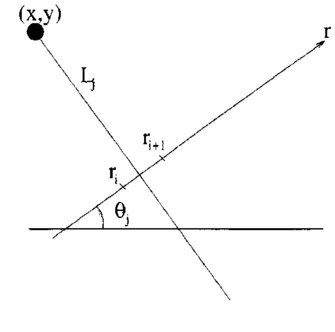

Fig. 6. Projection line in polar grid.

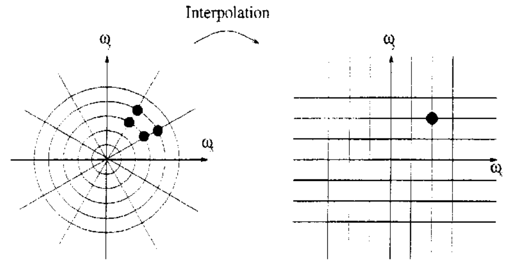

Fig. 7. Interpolation from polar to Cartesian grid.

However, there are of course also other considerations to take into account when developing algorithms, quality of the image being the most important one.

5. Direct Fourier methods (DFM)

There is still another way of reformulating the Radon transformation solution. Again we assume that the projections $p(r, \theta)$ in (9) are given, and that $\hat{p}(\omega, \theta)$ is the Fourier transform (10) with respect to r. Then a straightforward calculation shows that *the Projection Slice Theorem*

$$\hat{p}(\omega, \theta) = \hat{f}(\omega \cos \theta, \omega \sin \theta) \tag{11}$$

holds. At a first glance it seems as if the solution is now easily obtained by an inverse two-dimensional Fourier transformation. However, in order to use the FFT for the actual computation, we need to have the wave-numbers ω_x, ω_y available on a rectangular grid. By (11) they are now given on a grid corresponding to a polar coordinate system, and interpolation is required, see Figure 7.

If this interpolation can be done accurately enough with a few number of operations per point in the rectangular grid, then the DFM is very fast. The FFT and its inverse takes $O(N^2 \log_2 N)$ a.o., and the interpolation takes only $O(N^2)$ a.o.. The FFT is available, and interpolation is a simple technique that has been around long before computers were available. So, why is it that this method is not used for any commercial CT-machines? The main reason is that the interpolation takes place in Fourier space. We shall discuss a simple one-dimensional model problem that explains the inherent difficulty. Figure 8 shows a simple sine-wave, sin x, on the interval $[0, 2\pi]$, and a grid with $N = 16$ points.

If the values at the grid-points are connected with straight lines corresponding to linear interpolation, we get very accurate results. Consider now the Fourier sine-representation

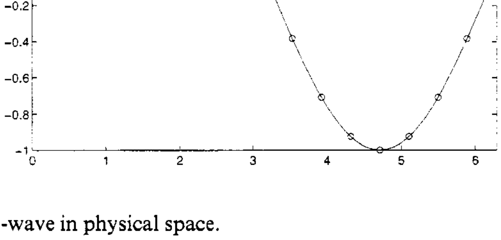

Fig. 8 Sine-wave in physical space.

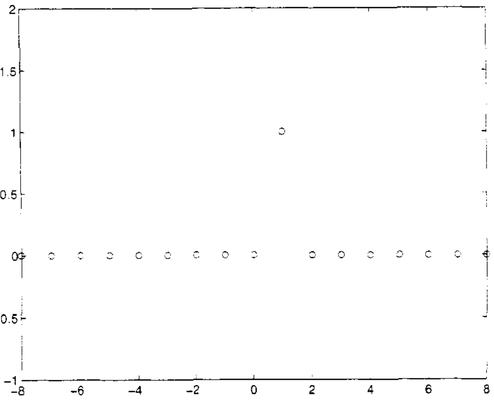

Fig. 9. Sine-wave in Fourier space.

$$f(x) = \sum_{\omega=-N/2}^{N/2} a_\omega \sin(\omega x)$$

in wave-number space in Figure 9.

The coefficients are zero for all ω except $\omega = 1$, where it is one. Assume now that we interpolate to get new values at the half-points $\omega = \pm 1/2, \pm 3/2, \ldots$. Again using linear interpolation, we get the new function

$$\tilde{f}(x) = \frac{1}{2}\sin\left(\frac{1}{2}x\right) + \frac{1}{2}\sin\left(\frac{3}{2}x\right).$$

This function is shown in Figure 10.

Except for the part near $x = 0$, it is completely wrong. The reason for this is that a smooth function in physical space is in general not smooth in Fourier space.

Our example is a special case of the following more general problem. Consider the function

$$f(x) = \sum_{\omega=-\infty}^{\infty} \hat{f}(\omega) e^{i2\pi\omega x},$$

where $\hat{f}(\omega) = 0$ for large $|\omega|$. Interpolate the coefficients linearly by

$$\hat{g}(\omega + 1/2) = \frac{1}{2}(\hat{f}(\omega) + \hat{f}(\omega + 1)),$$

and consider the resulting function in physical space

$$g(x) = \sum_{\omega=-\infty}^{\infty} \hat{g}(\omega + 1/2) e^{i2\pi(\omega+1/2)x}.$$

We have

$$g(x) = \sum_{\omega=-\infty}^{\infty} \frac{1}{2}(\hat{f}(\omega) + \hat{f}(\omega+1)) e^{i\pi x} e^{i2\pi\omega x} = \frac{1}{2} e^{i\pi x} \sum_{\omega} \hat{f}(\omega) e^{i2\pi\omega x}$$

$$+ \frac{1}{2} e^{-i\pi x} \sum_{\omega} \hat{f}(\omega + 1) e^{i2\pi(\omega+1)x} = \cos(\pi x) f(x).$$

Obviously, the new function obtained by interpolation in Fourier space is good in the neighborhood of $x = 0$, where $\cos(\pi x)$ is near one, but not elsewhere.

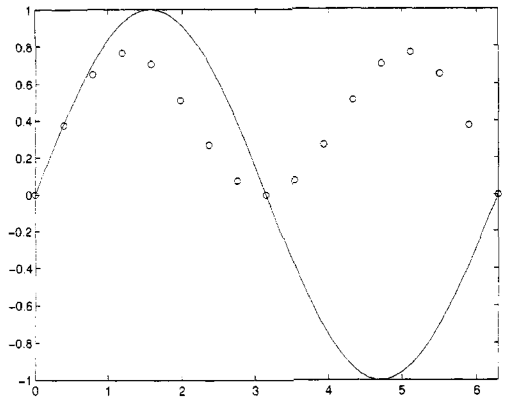

Fig. 10. New "sine-wave" after interpolation in Fourier space.

The result for the simple model problem studied here can be transferred to the two-dimensional image reconstruction problem. We cannot have methods that give accurate images in part of the domain, but not in others.

The problem of finding good interpolating methods in Fourier space is a topic for current research. If it can be solved, there is a good possibility that future CT-machines will be based on the direct Fourier method.

Acknowledgment

I would like to thank Lina Hemmingsson for making the figures in this article.

References

1. Censor, Y., Proc. IEEE **71**, 409 (1983).
2. Censor, Y., Elfving, T. and Herman, G. (Guest Eds) Linear Algebra Appl. **130**, 1 (1990).
3. Cooley, J. W. and Tukey, J. W., Math. Comp. **19**, 297 (1965).
4. Deans, S., "The Radon transform and some of its applications", (J. Wiley & Sons 1983).
5. Golub, G. H. and van Loan, C., "Matrix Computations", 2nd ed, (The John Hopkins University Press 1989).
6. Hemmingsson, L., Toeplitz preconditioners with block structure for first-order PDEs, Numer. Lin. Alg. Appl., To appear.
7. Herman, G., "Image Reconstruction from Projections. The Fundamentals of Computerized Tomography", (Academic Press, New York, 1980).
8. Herman, G. (Guest Ed), Proc. IEEE **71**, 291 (1983).
9. Herman, G. and Natterer, F. (Eds), "Mathematical Aspects of Computerized Tomography", (Springer-Verlag, Berlin, 1981).
10. Herman, G., Louis, A. K. and Natterer, F. (eds), "Mathematical Methods in Tomography", (Proc. Oberwolwach, Springer, 1990).
11. Herman, G., Tuy, H., Langenberg, K. and Sabatier, P., "Basic Methods of Tomography and Inverse Problems", (Adam Hilger, Bristol, 1987).
12. Lewitt, R. M., Proc. IEEE, **71**, No. 3, (1983).
13. Magnusson, M., "Linogram and other direct Fourier methods for tomographic reconstruction", Linköping Studies in Science and Technology. Dissertations No. 320, (1993).
14. Magnusson, M., Danielsson, P. E. and Edholm, P., Artefacts and remedies in direct Fourier tomographic reconstruction. Proc. IEEE Med. Imag. Conf., Orlando, pp. 1138–1140, (1992).
15. Natterer, F., "The mathematics of computerized tomography", (J. Wiley & Sons, Chichester 1986).
16. Radon, J., "Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten", Ber. Verh. Sächs. Akad. der Wissensch., Math.-Phys. **69**, pp 262–267, (1917).