1

Some Fundamental Issues in Computational Mathematics

Ronald DeVore
Department of Mathematics
University of South Carolina
Columbia, SC 29208
Email: devore@math.sc.edu

Abstract

We enter a discussion as to what constitutes the 'foundations of computational mathematics'. While not giving a definition, we give examples from image/signal processing and numerical computation where foundational issues have helped to 'correctly' formulate problems and guide their solution.

1.1 The question

While past chair of the organization Foundations of Computational Mathematics (FOCM), I was frequently asked what is the meaning of 'foundations of computational mathematics'. Most people understand what computational mathematics is. So the question really centers around the meaning of 'foundations' in this context. Even though I have thought about this quite a while, I would not dare to try to give a precise definition of foundations – I am sure it would be picked apart. However, I would like in this presentation to give some examples where the adherence to fundamental questions has helped to shape the formulation of computational issues and more importantly contributed to their solution. The examples I choose in signal/image processing and numerical methods for PDEs are of course related to my own research. I am sure

[0] This work has been supported by the Office of Naval Research Contract Nr. N0014-91-J1343, the Army Research Office Contract Nr. DAAD 19-02-1-0028, and the National Science Foundation Grant DMS0221642.

there are many other stories of the type I put forward that are waiting to be told.

The first of the three examples that I will discuss is that of image compression. This subject has grown rapidly over the last decade with an important infusion of ideas from mathematics especially the theories of wavelets and nonlinear approximation. The main topic to be addressed here is how we can decide which algorithms for compression are optimal.

A somewhat related topic will concern Analog to Digital (A/D) conversion of signals. This is an area that is important in consumer electronics. The story here centers around trying to understand why engineers do A/D conversion in the way they do, which by the way is very counterintuitive to what a first mathematical analysis would suggest.

Finally, I discuss adaptive methods for solving PDEs. This is an extremely important area of numerical computation in our quest to solve large problems to higher and higher resolution. The question to be answered is how can we know when an adaptive method is optimal in its performance.

1.2 Image compression

Digital signal processing has revolutionized the storage and transmission of audio and video signals as well as still images, in consumer electronics and in more scientific settings (such as medical imaging). The main advantage of digital signal processing is its robustness: although all the operations have to be implemented with, of necessity, not quite ideal hardware, the a priori knowledge that all correct outcomes must lie in a very restricted set of well separated numbers makes it possible to recover them by rounding off appropriately.

Every day, millions of digitized images are created, stored, and transmitted over the Internet or using other mediums. A grey scale image is an array (matrix) of pixel values. It is already important to have a mathematical model for what these pixel values represent. We shall view the pixel array as arising in the following fashion. We have a light intensity function f defined on a continuum Ω. For simplicity we assume that $\Omega := [0,1]^2$ and that f takes values in $[0,1)$ (the latter can be achieved by simple renormalization). Digitization corresponds to two operations: averaging and quantization. We take a tiling of Ω into squares Q and associate to each square Q the average intensity

$$f_Q := \frac{1}{|Q|}\int_I f(x)\,dx,$$

where $|Q|$ denotes the Lebesgue measure of Q. The pixel values p_Q are derived from the numbers $f_Q \in [0,1)$ by quantization. We write f_Q in its binary expansion

$$f_Q = \sum_{j=1}^{\infty} b_j(f_Q)2^{-j}$$

and define the pixel value $p_Q := \sum_{j=1}^{m} b_j(f_Q)2^{-j}$. Typical choices of m are $m = 8$ (one byte per pixel) or $m = 16$. The array $I := I(f) := (p_Q)$ of pixel values is a digitization of f. The accuracy at which I resolves f depends on the fineness of the tiling and the accuracy of the quantization (i.e. size of m). We do not really know f. We only see it through the digitized image $I(f)$. In practice, the pixel values p_Q are corrupted by noise but we shall ignore this in our discussion since we are aiming in a different direction.

We see that a digitized image in its raw form is described by mN bits where N is the number of squares in the tiling. *Lossy compression* seeks to significantly reduce this number of bits used to represent f at the expense of some loss in the fidelity of resolution. Hopefully, this loss of fidelity is not perceptible. There are two parts to a lossy compression scheme. The *encoder* assigns to each pixel array I a bitstream $B(I)$. A *decoder* gives the recipe for changing any given bitstream B back into a pixel array. After encoding and then decoding the resulting pixel values $\bar{p}_Q$ will generally not be the same as the original p_Q. Some fidelity is lost.

One can imagine, given the practical importance of the compression problem, that there are a ton of encoding/decoding schemes. How can one decide from this myriad of choices which is the best? Engineers have introduced a number called the PSNR (Peak Signal to Noise Ratio) which measures the performance of a given encoding/decoding on a given digitized image I. It is not necessary to give its precise definition here but simply mention that it measures the least squares distortion $((\#I)^{-1}\sum_Q [p_Q - \bar{p}_Q]^2)^{1/2}$ as a function of the number of bits. Here $\#(I)$ is the number of pixels. A new encoding scheme is tested by its performance (PSNR) on a few test images – the Lena image being the most widely used.

Now, there is a fundamental question here. Should the quality of a compression algorithm be determined by its PSNR performance on a few test images? Given a collection of 2^k images, we can encode them all with k bits per image by simply enumerating them in binary. So on a mathematical level this test of performance is quite unsatisfactory.

What has any of this to do with "foundations of computational mathematics". Well, we cannot have a decidable competition among

compression algorithms without a clear and precise formulation of the compression problem. This is a foundations question that rests on two issues that we must clarify. The first is the *metric* we are going to use to compare two images (for example, the original and the compressed image). The second is the class of images we wish to compress. We shall briefly discuss these issues.

1.2.1 The metric

We have already mentioned the PSNR which is based on the ℓ_2 metric. In our view of images as functions, this corresponds to the $L_2(\Omega)$ function metric. Is this the obvious choice? By no means. This choice seems to be more a matter of convenience and tradition. It is easy to solve optimization problems in the L_2 metric.

Certainly the choice of metric must depend on the intended application. In some targeted applications such as feature extraction and image registration, the least squares metric is clearly not appropriate and is better replaced by metrics such as L_∞ or maximum gradient.

Most compression is directed at producing visually pleasing images which cannot be distinguished from the original by the human eye. Thus, we can speak about the metric of the human visual system. The problem is that this vague notion is useless mathematically. Our goal would be to derive a mathematical metric which is a good model for the human visual system. There are some mathematical models for human vision which may be useful in directing our pursuit but little is agreed upon.

So, at this stage, we are left with using simple mathematical metrics such as the $L_p(\Omega)$ norms, $0 < p \leq \infty$, or certain smoothness norms. The point we wish to make here is not so much as to which metric is better but rather that any serious mathematical comparison of compression algorithms must at the outset agree on the metric used to measure distortion. Once this is decided we can go further.

1.2.2 Model classes of images

Once we have chosen a mathematical metric in which to measure distance between images, the question turns to describing the class of images that we wish to compress. This is a subject that spurs many interesting debates. We will touch on this only briefly and in a very prejudicial way.

There are two main models for images: the stochastic and the deterministic. Stochastic models are deeply embedded in the engineering and information theory communities influenced in a large part by Shannon's theory for optimal encoding. Deterministic models take the view we have presented of an image as a function defined on a continuum. We have begun by assuming only that an image is a bounded function. This is too broad of a class of functions to serve as the description of the images we wish to compress. Images have more structure.

One deterministic view of an image function f is that it is a sum of fundamental components corresponding to edges, texture, and noise. For example the famous model of Mumford and Shah [17] views the image as a sum $f = u + v$ of a component u of Bounded Variation (BV) and a component v in L_2. The component u is not an arbitrary BV function but rather has gradient given by a measure supported on a one dimensional set (corresponding to the edges in the image) and an L_1 part (corresponding to smooth regions in the image). The L_2 component v captures deviations from this model.

There are many variants of the Mumford–Shah model. These are beautifully described in the lecture notes of Meyer [14] – a must read. We wish to pick up on only one point of Meyer's exposition. Even when one settles on the functional nature of the two components u and v in the image, there are infinitely many ways to write $f = u + v$ depending on how much energy one wishes to put in each of these components. This is completely analogous to K-functional decompositions used in proving theorems on interpolation of operators. One needs to look at this totality of all such decompositions to truly understand f. For example, consider the case where we simply look for decompositions of $f = u + v$ where $u \in \mathrm{BV}$ and $v \in L_2$. We can give a quantitative description of these decompositions through the K-functional

$$K(f,t) := K(f,t;L_2,\mathrm{BV}) := \inf_{f=u+v} \|v\|_{L_2} + t|u|_{\mathrm{BV}}, \quad t > 0, \qquad (1.1)$$

where the $|\cdot|_{\mathrm{BV}}$ is the BV seminorm. For any fixed $t > 0$, the optimal decomposition in (1.1) tries to balance the two terms. Thus for t small it puts more energy into the BV component and less into the L_2 component. The rate of decrease in $K(f,t)$ as $t \to 0$ tells how nice f is with respect to this model.

The role of the K-functional is to distinguish between images. Certainly some images are more complex than others and more apt to be more difficult to compress. The rate at which a K-functional tends to

0 as $t \to 0$ measures this complexity. Thus, we can use the K-functional to separate images into classes K_α which are compact sets in our chosen metric. When classical metrics such as L_p norms are used, then these classes correspond to finite balls in smoothness spaces. In other words, using appropriate K-functionals, we can obtain a strata of image classes K_α reflecting the complexity of images.

1.2.3 Optimal encoding and Kolmogorov entropy

Suppose now that we have decided on the metric to be used to measure the distortion between two images and suppose we also have our model classes K_α for the classification of images. We shall assume that the metric is given by a quasi-norm $\|\cdot\| := \|\cdot\|_X$ on a topological linear space X. Each of the sets $K = K_\alpha$ is assumed to be a compact subset in the topology given by $\|\cdot\|$.

Recall that an encoder E for K is a mapping that sends each $f \in K$ into a bitstream $B(f) := B_E(f)$. Associated to E is a decoder D which takes any bitstream B and associates to it an element DB from X. Thus given $f \in K$, $\bar{f} := DEf = D(B_E(f))$ is the compressed image given by this encoding-decoding pair. This means that the distortion in the performance of this encoding on a given f is

$$d_E(f) := \|f - \bar{f}\| = \|f - DEf\|. \tag{1.2}$$

Of course, we are interested in the performance of this encoding not on just one element $f \in K$ but on the entire class. This leads us to define the *distortion for the class* K by

$$d_E(K) := \sup_{f \in K} d_E(f). \tag{1.3}$$

This distortion also depends on the decoder which we do not indicate. (One could become more specific here by always choosing for the given encoder E and set K the best decoder in the sense of minimizing the distortion (1.2).) To measure the complexity of the encoding we use

$$\#(E) := \#(E(K)) := \sup_{f \in K} \#(B(f)) \tag{1.4}$$

which is the maximum number of bits that E assigns to any of the elements of K.

We are interested in a competition among encoders/decoders to determine the optimal possible encoding of these classes. Suppose that we are

given a bit budget n; this means we are willing to allocate a maximum of n bits in the encoding of any of the elements of K. Then,

$$d_n(K) := \inf_{\#(E)\leq n} d_E(K) \tag{1.5}$$

is the minimal distortion that can be obtained for the class K with this bit budget n.

There is a mathematical description, called *Kolmogorov entropy*, that completely determines the optimal performance that is possible for an encoding of a given class K. Since K is compact in $\|\cdot\|$, for any given ϵ there is a collection of balls $\mathcal{B}(f_i,\epsilon)$, $i=1,\dots,N$, of radius ϵ centered at $f_i \in X$, such that

$$K \subset \bigcup_{i=1}^{N} \mathcal{B}(f_i,\epsilon). \tag{1.6}$$

The smallest number $N_\epsilon(K)$ of balls that provide such a cover is called the *covering number* of K. The Kolmogorov entropy of K (in the topology of X) is then given by

$$H_\epsilon(K) := \log N_\epsilon(K) \tag{1.7}$$

where here and later log always refers to the logarithm to the base 2. We fix K and think of $H_\epsilon(K)$ is a function of ϵ. It gives a measure of the massivity of K. The slower $H_\epsilon(K)$ tends to infinity as $\epsilon \to 0$ the more thin is the set K.

We can reverse the roles of ϵ and the entropy $H_\epsilon(K)$. Namely, given a positive integer n, let

$$\epsilon_n(K) := \inf\{\epsilon : \ H_\epsilon(K) \leq n\}. \tag{1.8}$$

The $\epsilon_n(K)$ are called the entropy numbers of K; they tend to zero as $n \to \infty$. The faster they tend to zero, the smaller the set K. Notice that an asymptotic behavior $H_\epsilon(K) = \mathcal{O}(\epsilon^{-1/\alpha})$ is equivalent to $\epsilon_n(K) = \mathcal{O}(n^{-\alpha})$.

The two notions of optimal distortion and entropy numbers are identical:

$$\epsilon_n(K) = d_n(K). \tag{1.9}$$

The proof is easy. Suppose E is an optimal encoder using n bits (if no such optimal encoder exists one modifies the following argument slightly). For each bitstream $B = B(f)$, $f \in K$, let $f_B := D(B)$ which is an element of X. Then, taking $\epsilon = d_n(K)$, we have $f \in \mathcal{B}(f_B,\epsilon)$. Since

there are at most 2^n distinct bitstreams $B(f)$, $f \in K$, we obtain that $H_\epsilon \le n$ and hence $\epsilon_n(K) \le \epsilon = d_n(K)$. We can reverse this inequality as follows. Suppose that n is given and $\epsilon = \epsilon_n(K)$. We assume that $H_\epsilon(K) \le n$. (We actually only know $H_\rho(K) \le n$ for all $\rho > \epsilon$ so that our assumption is not necessarily valid but we can easily modify the argument when $\epsilon_n(K)$ is not attained.) Let $\mathcal{B}(f_i, \epsilon)$, $i = 1, \dots, H_\epsilon(K)$, be a minimal covering for K with balls of radius ϵ. We associate to each i the binary bits in the binary representation of i. We define the encoder E as follows. If $f \in K$, we choose a ball $\mathcal{B}(f_i, \epsilon)$ that contains f (this is possible because these balls cover K) and we assign to f the bits in the binary representation of i. The decoder takes the bitstream, calculates the integer i which has these bits in its binary expansion, and assigns the decoded element to be the center of the ball $\mathcal{B}_i$. This encoding has distortion $\le \epsilon = \epsilon_n(K)$ and so we have $d_n(K) \le \epsilon_n(K)$.

The above discussion shows that the construction of an optimal encoder with distortion ϵ for the set K is the same as finding a minimal covering for K by balls of radius ϵ. Unfortunately, such coverings are usually impossible to find. For this reason, and others illuminated below, this approach is not very practical for encoding. On the other hand, it gives us a benchmark for the performance of encoders. If we could find an encoder which is nearly optimal for all the classes K of interest to us, then we could rest assured that we have done the job in the context in which we have framed the problem. We shall discuss in the next section how one could construct an encoder with these properties for a large collection of compact sets in standard metrics like L_p, $1 \le p \le \infty$.

1.2.4 Wavelet bases and compact subsets of L_p

A set K is compact in L_p provided that the modulus of smoothness

$$\omega(f,t)_p := \sup_{|h| \le t} \|\Delta_h(f,\cdot)\|_{L_p(\Omega)}, \quad t > 0 \tag{1.10}$$

for all of the elements $f \in K$ have a continuous majorant ω_K

$$\sup_{f \in K} \omega(f,t)_p \le \omega_K(t) \tag{1.11}$$

where $\omega_K(0) = 0$. The rate at which ω_K tends to zero at 0 measures the compactness of K. Thus the natural compact sets in L_p are described by common smoothness conditions. This leads to the Sobolev and Besov smoothness spaces. For example, the Besov spaces are defined by conditions on the higher order moduli of smoothness of $f \in L_p$. We denote

these Besov spaces by $B_q^s(L_p(\Omega))$ where p is the L_p space in which we are measuring smoothness. The parameter $s > 0$ gives the order of smoothness much like the number of derivatives. The parameter $0 < q \leq \infty$ is a fine tuning parameter which makes subtle distinctions between these spaces. We do not make a precise description of these spaces at this juncture but we shall give a description of these spaces in a moment using wavelet bases.

The reader is probably by now quite familiar with wavelet bases. We shall limit ourselves to a few remarks which will serve to describe our notation. When working on the domain $\mathbb{R}$, a wavelet basis is given by the shifted dilates $\psi_\lambda := \psi(2^j \cdot -k)$, $\lambda = (j,k)$, of one fixed function ψ. When moving to $\mathbb{R}^d$, one needs the shifted dilates of a collection ψ^e of $2^d - 1$ functions; the parameter e is usually indexed on the set E of nonzero vertices of the unit cube $[0,1]^d$. Thus the wavelets are indexed by three parameters $\lambda = (j,k,e)$ indicated frequency (j), location (k) and type (e). When working on a finite domain, two adjustments need to be made. The first is that the range of j is from $j_0 \leq j < \infty$. The coarsest level $j = j_0$ corresponds to scaling functions; all other j correspond to the actual wavelets. For notational convenience, we shall take $j_0 = 0$ in what follows. The second adjustment is that near the boundary some massaging has to be made in defining ψ_λ.

Thus, a wavelet basis on a finite domain Ω in $\mathbb{R}^d$ is a collection $\Psi = \{\psi_\lambda : \lambda \in \mathcal{J}\}$ of functions ψ_λ. The indices λ encode scale, spatial location and the type of the wavelet ψ_λ. We will denote by $|\lambda|$ the *scale* associated with ψ_λ. We shall only consider compactly supported wavelets, i.e., the supports of the wavelets scale as follows

$$S_\lambda := \operatorname{supp} \psi_\lambda, \quad c_0 2^{-|\lambda|} \leq \operatorname{diam} S_\lambda \leq C_0 2^{-|\lambda|}, \tag{1.12}$$

with c_0 and $C_0 > 0$ absolute constants. The index set $\mathcal{J}$ has the following structure $\mathcal{J} = \mathcal{J}_\phi \cup \mathcal{J}_\psi$ where $\mathcal{J}_\phi$ is finite and indexes the scaling functions on the fixed coarsest level 0. $\mathcal{J}_\psi$ indexes the "true wavelets" ψ_λ with $|\lambda| > 0$. From compactness of the supports we know that at each level, the set $\mathcal{J}_j := \{\lambda \in \mathcal{J} : |\lambda| = j\}$ is finite. In fact, one has $\#\mathcal{J}_j \sim 2^{jd}$ with constants depending on the underlying domain.

There is a natural tree structure associated to wavelet bases. A node in this tree corresponds to all $\lambda = (j,k,e)$, $e \in E$, with j,k fixed. In the case the domain is $\mathbb{R}^d$, each such node has 2^d children corresponding to the indices $(j+1, 2(k+e))$ where $e \in \{0,1\}^d$. In other words, the children all occur on the next dyadic level. In the case of Haar functions, the supports of the wavelets corresponding to children are contained

in those corresponding to a given parent. This is modified on domains because only some of the indices are used on the domain.

Wavelet bases have many remarkable properties. The first that we want to pick up on is that Ψ is an *unconditional basis* for many function spaces X. Consider first the case that $X = L_2(\Omega)$. Then every $f \in L_2(\Omega)$ has a unique expansion $f = \sum f_\lambda \psi_\lambda$ and there exist some constants c and C independent of f such that

$$c\|(f_\lambda)_{\lambda\in\mathcal{J}}\|_{\ell_2} \le \|\sum_{\lambda\in\mathcal{J}} f_\lambda\psi_\lambda\|_{L_2(\Omega)} \le C\|(f_\lambda)_{\lambda\in\mathcal{J}}\|_{\ell_2}. \tag{1.13}$$

In the case of L_p spaces, $p \neq 2$, the norm $\|f\|_{L_p(\Omega)}$ is not so direct and must be made through the square function. However, if we normalize the basis in L_p, $\|\psi_\lambda\|_{L_p(\Omega)} = 1$, then the space B_p of functions $f = \sum_{\lambda\in\mathcal{J}} f_\lambda\psi_\lambda$ satisfying

$$\|f\|_{B_p} := \|(f_\lambda)\|_{\ell_p} \tag{1.14}$$

is very close to $L_p(\Omega)$ and can be used as a *poor man's* substitute in many instances. By the way, B_p is an example of a Besov space $B_p = B_p^0(L_p)$ where the smoothness order is zero.

Besov spaces in general have a simple description in terms of wavelet coefficients. If $f = \sum_{\lambda\in\mathcal{J}} f_\lambda\psi_\lambda$ with the ψ_λ normalized in L_p, $\|\psi_\lambda\|_{L_p(\Omega)} = 1$, then

$$\|h\|_{B_q^s(L_p(\Omega))} := \begin{cases} \left(\sum_{j=0}^\infty 2^{jsq}\left(\sum_{|\lambda|=j}|f_\lambda|^p\right)^{q/p}\right)^{1/q}, & 0 < q < \infty, \\ \sup_{j\ge 0} 2^{js}\left(\sum_{|\lambda|=j}|f_\lambda|^p\right)^{1/p}, & q = \infty. \end{cases} \tag{1.15}$$

Suppose that we fix $1 \le p \le \infty$ and agree to measure the distortion of images in the $L_p(\Omega)$ norm. Which of the Besov spaces are embedded in $L_p(\Omega)$ and which are compactly embedded? This is easily answered by the Sobolev embedding theorem. The unit ball of the Besov space $B_q^s(L_\tau(\Omega))$ is a compact subset if and only if $\frac{1}{\tau} < \frac{s}{d} - \frac{1}{p}$. Notice that this condition does not depend on q. When $\frac{1}{\tau} = \frac{s}{d} - \frac{1}{p}$ (the so-called *critical line* in the Sobolev embedding) then the Besov space $B_q^s(L_\tau(\Omega))$ is embedded in $L_p(\Omega)$ for small enough q but these embeddings are not compact.