Mon. Not. R. astr. Soc. (1984) **211**, 111–124

Maximum entropy image reconstruction: general algorithm

J. Skilling and R. K. Bryan* *Department of Applied Mathematics and Theoretical Physics, Silver Street, Cambridge CB3 9EW*

Accepted 1984 May 14. Received 1984 May 14; in original form 1982 July 23

Summary. Maximum entropy is an optimal technique of image reconstruction, widely applicable in astronomy and elsewhere. We present a general-purpose algorithm, capable of generating maximum entropy images from a wide variety of types of data.

1 Introduction

Maximum entropy is being increasingly widely used as a general and powerful technique for reconstructing positive images from noisy and incomplete data. In astronomy, it has been used throughout the electromagnetic spectrum for radio aperture synthesis (Gull & Daniell 1978; Scott 1981), for optical deconvolution (Frieden & Swindell 1976; Frieden & Wells 1978; Bryan & Skilling 1980), for X-ray imaging (Gull & Daniell 1978; Willingale 1981) and for gamma-ray imaging (Skilling, Strong & Bennett 1979), and for eclipse mapping of accretion discs (Horne 1982). The technique can also be used in other fields such as structural molecular biology (Bryan *et al.* 1983) and medical tomography (Minerbo 1979; Kemp 1980). As well as producing images of optimal quality, maximum entropy can also be used to re-calibrate poorly known parameters such as phases or instrumental drifts (Scott 1981). A review of many of these applications is given in Skilling (1981).

Many of these papers, and others referred to therein, contain comparisons between reconstructions by conventional methods and by maximum entropy. We believe that these comparisons clearly demonstrate the superiority of maximum entropy for producing optimum general-purpose restorations of images from incomplete and noisy data.

In this paper we are concerned with presenting the rationale and details of a robust and efficient algorithm for computing maximum-entropy images which has been developed in Cambridge. This algorithm deals routinely with images of up to a million or more pixels, and with dynamic ranges well in excess of 10000. It can be applied to many different problems with the minimum of changes, by rewriting a few computer subroutines which define the transforms between image and data.

In Section 2 of the paper, a suitable form of the maximum entropy criterion is set up.

* Present address: European Molecular Biology Laboratory, 6900 Heidelberg, Germany.

Section 3 is devoted to a survey of algorithms and develops the general-purpose algorithm we recommend. Section 4 summarizes the ingredients of this successful program.

2 The maximum entropy criterion

An image can be regarded as a set of positive numbers $f_1, f_2, \ldots, f_N$ which are to be determined, and on which the entropy

$$S(f) = -\sum_{j=1}^{N} p_j \log p_j, \qquad p_j = f_j/\Sigma f \tag{1}$$

is defined. Use of this form of entropy (Shannon 1948) in the context of image reconstruction is originally due to Frieden (1972).

The theoretical foundation of the maximum entropy method in data analysis is that this method is the *only* consistent way of selecting a single image from the very many images which fit the data. Shore & Johnson (1980, 1983) proved this axiomatically, a more readable account of their ideas has been given by Gull & Skilling (1983), and a proof of intermediate formality appears in Livesey & Skilling (1984). These papers show that maximum entropy is the only method which does not introduce correlations in the image, beyond those which are required by the data.

Entropy can also be justified in information-theoretic terms. Given an image radiating with intensity pattern f_j, the entropy measures the number of bits of information needed to localize the position j of a single radiated photon. Maximizing S, subject to observational constraints, involves seeking a maximally non-committal answer to the fundamental question 'Where would the next photon come from?' (Skilling & Gull 1984).

The practical merit of maximizing entropy is that the resulting image has minimum configurational information, so that there must be evidence in the data for any structure which is seen, and the displayed structure is uniquely easy to comprehend. Also, the physically important requirement of positivity is automatically invoked, since the entropy does not even exist if any of the f_j are negative. Numerically, it is far easier to ensure positivity via a single, smooth function such as S, than via N separate inequality constraints $f_j \geqslant 0$.

The observational constraints on permitted reconstructions come from data D_k related in some known way to the image, and subject to some form of noise. Thus, for additive noise,

$$D_k = R_k(f) + n_k \sigma_k \tag{2}$$

where R is the response function of the observing equipment, σ_k is the standard error on datum k and n_k is a random variable of zero mean and unit variance. In any linear experiment, $R_k(f) = \Sigma R_{kj} f_j$ so that R becomes a matrix. For example, in interferometry, $R_k(f)$ would be the Fourier component of f corresponding to spacing k, so that R_{kj} would be a Fourier matrix. Likewise in deconvolution, R_{kj} would be a convolution matrix, (usually of Toeplitz form).

Naively, one might attempt to recover f from the data by applying R^{-1}, but this fails in principle whenever the data are incomplete, because R^{-1} is not uniquely defined. It also fails in practice whenever R^{-1} is badly conditioned, as in most deconvolution problems. Inverse filters alleviate this difficulty by modifying R^{-1} to make it better conditioned, albeit incorrect in the sense that retransforming by R will not reproduce the original data. What one can do correctly, using R itself instead of its inverse, is eliminate those f which are inconsistent with the data. The data can do no more than this.

The formal observational constraint on reconstructions f is set up by comparing the *actual* (noisy) data D_k with the simulated data

$$F_k = R_k(f) \tag{3}$$

which would be obtained (in the absence of noise) if the pattern being observed were indeed represented by the numbers f. A reconstruction f is said to be *feasible* if the simulated data agree with the actual data to within the noise. Feasible reconstructions are those which are not contradicted by the data. Note that the comparison is made only with the data points actually measured. There is no assumption, implicit in inverse filter methods, that unmeasured data values are zero.

A single constraint statistic $C(f)$, usually chi-squared (Ables 1974; Gull & Daniell 1978) is used to measure the misfit;

$$C(f) = \chi^2 = \Sigma(F_k - D_k)^2/\sigma_k^2 \tag{4}$$

where the summation is over the observed k. Different choices for $C(f)$ are also possible (Bryan & Skilling 1980), and may be preferable in certain circumstances. For other forms of noise, e.g. Poisson, an appropriately modified statistic $C(f)$ should be used.

Statistical analysis indicates some upper bound C_{aim} to the values which C can plausibly take. For chi-squared, the largest acceptable value at 99 per cent confidence is about $(M + 3.29\sqrt{M})$, where M is the number of observations. It is much easier to use a single statistic than to attempt to fit each separate datum, both because this avoids an unwieldy proliferation of Lagrange multipliers and because one can construct alternative statistics $C(f)$ that tolerate occasional individual errors of several standard deviations. For example one can ignore extreme outliers in the calculated noise residuals (Bryan & Skilling 1980; Burch, Gull & Skilling 1983).

The strict maximum entropy criterion requires one to select that particular feasible image which has the greatest entropy. One maximizes S subject to $C \leqslant C_{\text{aim}}$. If the unconstrained maximum of S satisfies this constraint, then this will be the maximum entropy solution – the data are too noisy for any information to be extracted. Otherwise the solution will lie on the boundary $C = C_{\text{aim}}$ and we have an optimization problem with an equality constraint

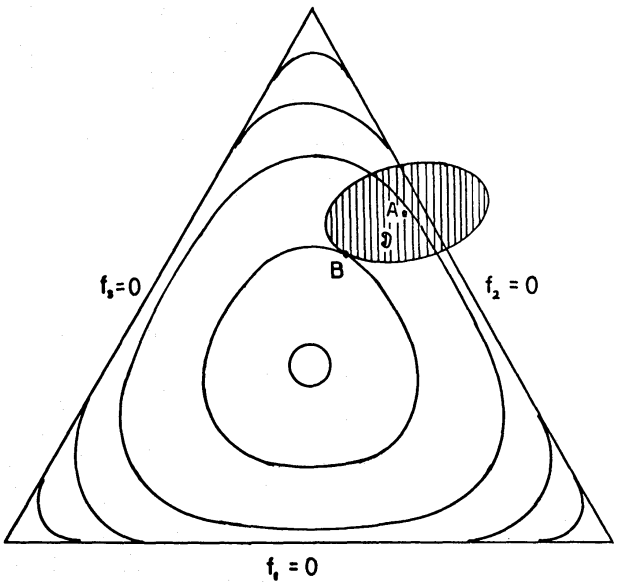

Figure 1. The S criterion and the χ^2 statistic in f-space for a 3-cell map normalized to $\Sigma f = 1$. S surfaces are convex and χ^2 surfaces are ellipsoids. A is the image which fits the data exactly; B is the maximum-entropy image.

to solve (Fig. 1). As usual for such problems, a Lagrangian function $Q = S - \lambda C$ can be set up, and the solution will lie at an extremal of Q for some value of the Lagrange multiplier λ. From this, f can be determined via

$$\log A - \log f_j = (\lambda \Sigma f)\, \partial C/\partial f_j, \qquad A = \exp(\Sigma p_i \log f_i). \tag{5}$$

A is a weighted mean of the f_j which can be interpreted as the default value to which f_j will tend if there are no data pertaining to cell j ($\partial C/\partial f_j = 0$).

In many applications, interferometry being typical, the total flux Σf has a special status, as indeed it does in the entropy itself. The maximum entropy criterion may accordingly be modified to maximize S subject to $C = C_{\text{aim}}$ and to some constraint on Σf. This will be found at an extremal of $S - \mu \Sigma f - \lambda C$, where μ is an extra multiplier which has the operational effect of changing A. One can either choose μ, and hence A, to fit some given value of total flux Σf, or, more simply, one can use A itself as a user-defined default intensity or 'sky background'. This is what the astronomer usually wants. Formally, this is equivalent to modifying the entropy to

$$S = -\sum_j f_j [\log (f_j/A) - 1] \tag{6}$$

whose derivatives are

$$\partial S/\partial f_j = \log A - \log f_j, \qquad \partial^2 S/\partial f_i \partial f_j = -\delta_{ij}/f_j. \tag{7}$$

It is this latter form (6) of S which is used in this paper, although the algorithms which are presented can easily be modified to cope with the strict form of S, or indeed with other modifications of it.

For any linear experiment, the surfaces of constant chi-squared (Fig. 1) are convex ellipsoids in N-dimensional image space. Since the entropy surfaces are strictly convex, the maximum entropy reconstruction is unique.

In this section we have set up the maximum entropy method as an equality constrained optimization problem. It is important that the numerical algorithm used solve the problem is reliable, in the sense that it should produce the correct solution, or give a definite indication of failure. There is no point in using an algorithm which terminates after a certain number of iterations if there are no tests for fitting the data and for maximizing the entropy. In Section 3 we survey various algorithms, and develop a general purpose algorithm which we believe satisfies these criteria.

3 Maximum entropy algorithms

Entropy being intrinsically non-linear, the computational problem is one of constrained non-linear optimization. The problem is also large-scale, since an image contains N elements f_j to be determined, and N may be a million or more. The number M of observations may be many thousand. The image and data may be considered as vectors in N or M-dimensional linear spaces, even if they represent physical arrays in 1, 2, or even 3 dimensions. It follows immediately from the size of the problem that vector operations such as scalar products, adds, multiplies etc. are allowed in N-dimensional image-space and M-dimensional data-space, but matrix operations of $O(N^2)$ or $O(M^2)$ are prohibited. Consequently for these large-scale images, it is not possible to use methods such as Newton-Raphson iteration, which has been employed successfully on smaller problems (Frieden 1972).

The non-linear nature of the problem also forces the algorithm to be iterative. This necessarily involves repeated passage from image-space to data-space via the response function R_{kj}. Each trial image, for example, must be transformed to the corresponding dataset F_k before its constraint statistic C can be evaluated. There must also be repeated feedback from data-space to image-space, which again involves R. For example the gradient $\partial C/\partial f_j$, which is useful in determining how to adjust a trial image in order to fit the data better, is calculated by

$$\partial C/\partial f_j = \sum_k (\partial F_k/\partial f_j)(\partial C/\partial F_k) = \sum_k R_{kj} 2(F_k - D_k)/\sigma_k^2. \tag{8}$$

The summation is now on the first index of R_{kj}, showing that feedback is obtained through the *transpose* of R (*not* its inverse, which may well not exist).

All the algorithms discussed in this paper are based purely on vector operations in image-space and in data-space, together with image-data transformations by R or its transpose. Normally, the major computational overhead is in the image-data transformations, and these should be coded efficiently. Thus an interferometric problem, for which R is a Fourier transform, should be coded via a Fast-Fourier-Transform routine, using $O(N \log N)$ operations.

In the discussion of algorithms which follows, Section 3.1 discusses a potentially promising algorithm which nevertheless proved insufficiently powerful. Sections 3.2, 3.3, 3.4, 3.5 develop the basic ideas for a successful technique. This is presented in Section 3.6 (which sets up the image-space structures needed) and in Section 3.7 (which gives the procedure for control of these structures).

3.1 THE 'INTEGRAL EQUATION'

Gull & Daniell (1978) attempted to find the solution of (5) by directly maximizing $Q = S - \lambda C$ at fixed λ. They used the iterative form

$$f_j^{(n+1)} = A \exp\left[-\lambda \partial C(f^{(n)})/\partial f_j\right] \tag{9}$$

where (n) denotes the nth iterate. This procedure had the attractive feature that successive iterates were all automatically positive, because of the exponential function. Also, the algorithm allowed high values of f to develop in relatively few iterations, again because of the exponential. This was of considerable importance in Gull & Daniell's astronomical applications, because the dynamic range of their images was often $O(1000)$.

Unfortunately, the exponential also introduced instability into the iteration and they had to smooth successive iterates by setting

$$f_j^{(n+1)} = (1-p) f_j^{(n)} + pA \exp\left[-\lambda \partial C(f^{(n)})/\partial f_j\right]. \tag{10}$$

Even then, the behaviour of the algorithm was erratic and unstable, especially at high values of λ, and the proportion p often had to be reduced so severely that the algorithm effectively stopped. This instability had also been noted by Willingale (1979).

Nevertheless, the work of Gull & Daniell was of crucial importance in demonstrating the possibility of computing high-resolution maximum-entropy images for a variety of different experiments.

3.2 STEEPEST ASCENTS

The simplest approach is to maximize $Q = S - \lambda C$ by steepest ascents, using

$$f_j^{(n+1)} = f_j^{(n)} + x\partial Q(f^{(n)})/\partial f_j \tag{11}$$

for suitable x. The catastrophic disadvantage of this method is that in almost every case, whenever x is sufficiently large to enable high values f_j of the image to develop significantly, there are also cells with negative $\partial Q/\partial f_j$ at which f_j becomes significantly negative. The entropy is not defined when $f_j < 0$, so that after each iteration any negative f's must be reset to small positive values. If many f's become negative, the effect is to stop the algorithm making progress towards a maximum.

3.3 CONJUGATE GRADIENTS

The standard way of improving a steepest ascent algorithm is to use the conjugate gradient technique (Fletcher & Reeves 1964) or a variant of it (e.g. Polak 1971; Powell 1976, 1977). At the nth iteration, instead of using ∇Q itself as a direction in which to look for a maximum of Q, one uses only that part $\mathbf{e}^{(n)}$ of ∇Q which is conjugate to the previous directions ($\mathbf{e}^{(r)}$, $r = 1, 2, \ldots, n-1$), defined by $\mathbf{e}^{(n)T} \cdot \nabla\nabla Q \cdot \mathbf{e}^{(r)} = 0$. In the terminology of this paper, the technique seeks a maximum of Q over the points $\mathbf{f}^{(n)} + x\mathbf{e}^{(n)}$ lying along a search direction $\mathbf{e}^{(n)}$, where

$$\mathbf{e}^{(n)} = -\nabla Q + \beta \mathbf{e}^{(n-1)} \tag{12}$$

with $\beta = |\nabla Q^{(n)}|^2 / |\nabla Q^{(n-1)}|^2$ or a formal equivalent thereof. The coefficient β in the search direction is derived on the assumption that Q has constant curvature. However, Q in maximum entropy is highly non-quadratic, and it may not be sensible to assume that curvature information can be carried forward for several iterates.

Even so, as noted by Wernecke & d'Addario (1977) for a similar problem, conjugate gradients afford a considerable improvement over steepest ascents, although the algorithm remains plagued by negative values of f_j, and still concentrates too much on small values.

3.4 SEARCH DIRECTIONS FOR THE UNCONSTRAINED PROBLEM (FIXED λ)

The conjugate gradient technique attempts to build up information about the $N \times N$ Hessian matrix $\nabla\nabla Q$ by using successive vectors ∇Q, calculated at successive points $f^{(n)}$. It then uses a specific linear combination of the various ∇Q as a search direction along which Q is maximized either by a fixed coefficient based on a Newton-Raphson increment or by an exact line search.

In the maximum entropy problem the main computational cost of this lies in generating the successive vectors ∇Q, each of which requires an image-data transformation R followed by its transpose to calculate ∇C (equation 8). Scalar products between the vectors are much quicker to compute. Accordingly, *one can gain considerable extra flexibility at negligible extra computational cost by constructing, not merely one line along which to search, but rather a full subspace spanned by several vectors.*

Let the vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r$ ($r < 10$ say) be these base vectors. Then, within the subspace so spanned, one may construct a quadratic model

$$\tilde{Q}(x) = Q_0 + Q_\mu x^\mu + \tfrac{1}{2} H_{\mu\nu} x^\mu x^\nu \tag{13}$$

for the value of Q at increment $\delta f = x^{\mu}\mathbf{e}_{\mu}$. Note the covariant and contravariant indices: Greek indices refer to subspace quantities. The components of the model are

$$Q_{\mu} = \mathbf{e}_{\mu}^{T} \cdot \nabla Q \quad \text{and} \quad H_{\mu\nu} = \mathbf{e}_{\mu}^{T} \cdot \nabla\nabla Q \cdot \mathbf{e}_{\nu}, \tag{14}$$

chosen to agree with the local gradient and curvature components of $Q(f)$ itself. Then, within the subspace, $\tilde{Q}$ is maximized at

$$x^{\mu} = -(H_{\mu\nu})^{-1} Q_{\nu}, \tag{15}$$

the evaluation of which involves the trivially quick task of solving r simultaneous equations. The resulting value of $\tilde{Q}$, moreover, is greater than could have been obtained directly from conjugate gradients, because such a point is necessarily included in the subspace spanned by the $\mathbf{e}_{\mu}$.

As suggested above, $H_{\mu\nu}$ is obtained from the current and the previous $r-1$ evaluates of ∇Q. However, this presupposes that the curvature of $Q(f)$ remains nearly constant even when f is incremented $r-1$ times. That is an extremely severe restriction on the increment length $|\delta f|$, since in the maximum entropy problem the curvature of Q is dominated by $1/f_j$ terms from the entropy, and these are very sensitive to small changes whenever a particular f_j is small. It is better to evaluate the search directions at the present position f as

$$\mathbf{e}_1 = \nabla Q, \qquad \mathbf{e}_2 = \nabla\nabla Q \cdot \nabla Q, \ldots, \qquad \mathbf{e}_r = (\nabla\nabla Q)^{(r-1)} \cdot (\nabla Q). \tag{16}$$

Admittedly this involves the conventionally unorthodox step of discarding information from previous iterates, but all our attempts to use old curvature information resulted in marked reductions in algorithm efficiency.

3.5 ENTROPY METRIC

Even with models which are completely updated at each iteration, some limit must be placed on the difference δf between successive iterates, as the quadratic model will still be inaccurate at large distances. One should maximize $\tilde{Q}(x)$ subject to $|\delta f|^2 \leqslant l_0^2$ for some l_0.

The precise form of the distance limit bears closer investigation. So far, the main disadvantage of the search-direction algorithms has been their tendency to allow negative values of f. A distance limit $\Sigma(\delta f_i)^2 \leqslant l_0^2$ alleviates this, but at the cost of drastically slowing the attainment of high values. However, the distance limit can be modified to overcome this defect. Logarithmic modification $\Sigma(\delta f_i/f_i)^2 \leqslant l_0^2$ is too severe on low values, and the intermediate form

$$\sum_i (\delta f_i)^2/f_i \leqslant l_0^2 \tag{17}$$

is a good practical compromise. It discriminates in favour of allowing high values to change more than low ones, but not excessively so. The actual value of l_0^2 should be $O(\Sigma f)$ on dimensional grounds, and values around $0.1\,\Sigma f$ to $0.5\,\Sigma f$ are useful in practice.

Using a distance in this form is equivalent to putting a metric

$$g_{ij} = 1/f^{i} (i = j) \quad \text{and} \quad g_{ij} = 0 (i \neq j) \tag{18}$$

onto image-space (note covariant and contravariant indices). But this is just minus $\nabla\nabla S$ (Bryan 1980)! This metric is far simpler and more convenient than the Hessian metric

$g_{ij} = \partial^2 Q/\partial f_i\, \partial f_j$ normally used (Sargent 1974) in variable-metric non-linear optimization problems.

Using $-\nabla\nabla S$ as the metric is the single most important key to the development of a robust algorithm. With a non-Cartesian metric, the gradient direction $\nabla Q = \partial Q/\partial f^j$ appears initially in covariant form. In order to increment the contravariant vector f^i, its index must be raised by g^{ij}, giving $\mathbf{f}^i \partial Q/\partial \mathbf{f}^i$ [componentwise multiplication, henceforth represented by $f(\nabla Q)$] as the basic contravariant search direction. Furthermore, the second derivative matrix $\nabla\nabla Q = \partial^2 Q/\partial \mathbf{f}^i \partial \mathbf{f}^j$ must likewise be premultiplied by g^{mi} if it is to map contravariant vectors onto contravariant vectors. This gives the revised set of search directions

$$\mathbf{e}_1 = f(\nabla Q), \quad \mathbf{e}_2 = f(\nabla\nabla Q)\, f(\nabla Q), \ldots, \tag{19}$$

We note here that the integral equation (9) can be written as

$$f_j^{(n+1)} = A \exp\left[\partial Q(f^{(n)})/\partial f_j - \partial S(f^{(n)})/\partial f_j\right]$$
$$= f_j^{(n)} \exp\left[\partial Q(f^{(n)})/\partial f_j\right].$$

On expanding the exponential to first order, we obtain

$$f_j^{(n+1)} - f_j^{(n)} = f_j^{(n)}\, \partial Q(f^{(n)})/\partial f_j.$$

So, for small increments, the integral equation is equivalent to a steepest ascent optimization using the entropy metric.

3.6 SEARCH DIRECTIONS FOR THE CONSTRAINED PROBLEM

One difficulty with maximizing Q is that of λ, which still has to be iterated to fit $C = C_{\mathrm{aim}}$. This double iteration is clumsy and inefficient. However, the use of different values of λ involves using different proportions of S and C in Q, which suggests using *two* models in the subspace, one for S and the other for C, and attempting somehow to *solve the actual problem of maximizing* S *subject to* $\mathrm{C} = \mathrm{C_{aim}}$ *directly without using* λ *explicitly*. This would also be a more general approach, since there are non-linear experiments R for which the entropy maximum on the constraint surface may not be obtainable by maximizing Q, whatever value of λ is chosen. Examples in which the desired extremal of Q is not a maximum are given in Bryan (1980) and Livesey & Skilling (1984).

The subspace itself would be constructed from

(1) 2 directions $f(\nabla S)$ and $f(\nabla C)$,

(2) 4 directions, $f(\nabla\nabla S)$ and $f(\nabla\nabla C)$ operating on (1),

(3) 8 directions, $f(\nabla\nabla S)$ and $f(\nabla\nabla C)$ operating on (2),

and so on. But $f(\nabla\nabla S)$ is minus the identity operator and nothing new is obtained by operating with it. The search directions reduce to

$$\begin{aligned}
&(1)\, f(\nabla S), \qquad f(\nabla C)\\
&(2)\, f(\nabla\nabla C)\, f(\nabla S), \qquad f(\nabla\nabla C)\, f(\nabla C)\\
&(3)\, f(\nabla\nabla C)\, f(\nabla\nabla C)\, f(\nabla S), \qquad f(\nabla\nabla C)\, f(\nabla\nabla C)\, f(\nabla C)
\end{aligned} \tag{20}$$

and so on. It is worth repeating that the operator $\nabla\nabla C$, though formally a matrix, can be applied by vector operations allied to image-data transformations.

The factors of f^i in the search directions discriminate in favour of high values, and this helps to keep all the values positive. In fact it is rare for any cell to be sent negative when these search directions are properly controlled. Protection against stray negative values is still needed, but it does not slow the algorithm and is no longer a source of difficulty.

The family (20) of directions is sufficiently powerful that the first four enable most practical problems to be solved. Indeed even this level of complexity is usually unnecessary, as the third and fourth directions can normally be replaced by a single difference combination, giving just three search directions

$$\begin{aligned} \mathbf{e}_1 &= f(\nabla S) \\ \mathbf{e}_2 &= f(\nabla C) \\ \mathbf{e}_3 &= |\nabla S|^{-1} f(\nabla\nabla C) f(\nabla S) - |\nabla C|^{-1} f(\nabla\nabla C) f(\nabla C). \end{aligned} \tag{21}$$

Here the entropy metric is used to define the lengths

$$|\nabla S| = [\Sigma f^i (\partial S/\partial f^i)^2]^{1/2}, \quad \nabla C = [\Sigma f^i (\partial C/\partial f^i)^2]^{1/2}. \tag{22}$$

With these three (or four or more) search directions, quadratic models for S and C

$$\tilde{S}(x) = S_0 + S_\mu x^\mu - \tfrac{1}{2} g_{\mu\nu} x^\mu x^\nu, \quad \tilde{C}(x) = C_0 + C_\mu x^\mu + \tfrac{1}{2} M_{\mu\nu} x^\mu x^\nu \tag{23}$$

where

$$S_\mu = \mathbf{e}_\mu^T \cdot \nabla S, \quad g_{\mu\nu} = \mathbf{e}_\mu^T \cdot \mathbf{e}_\nu, \quad C_\mu = \mathbf{e}_\mu^T \cdot \nabla C, \quad M_{\mu\nu} = \mathbf{e}_\mu^T \cdot \nabla\nabla C \cdot \mathbf{e}_\nu \tag{24}$$

are constructed in the subspace

$$\mathbf{f}^{(\text{new})} = \mathbf{f} + x^\mu \mathbf{e}_\mu = \mathbf{f} + \delta\mathbf{f} \tag{25}$$

parameterized by x, within which the length-squared of the increment $\delta\mathbf{f}$ is

$$l^2 = g_{\mu\nu} x^\mu x^\nu. \tag{26}$$

It is rather remarkable that such a small space can capture enough of the structure of a non-linear optimization problem in a million dimensions.

3.7 CONTROL PROCEDURES

Control of the algorithm now passes into the subspace, in order to determine suitable coefficients x^μ for the search directions. The problem in the subspace becomes one of optimizing a quadratic function subject to quadratic constraints. Although the control procedure involves substantial programming, the computation time involved is negligible in comparison with practical image-data transformations.

3.7.1 Diagonalization in the subspace

This preliminary step simplifies the algebra. First, the base vectors $\mathbf{e}_\mu$ are normalized by scaling the model parameters and the metric tensor $g_{\mu\nu}$ is diagonalized. The algorithm can now be protected against linear dependence of the search directions. Such dependence shows up as one or more unusually small eigenvalues of $g_{\mu\nu}$. Components of the model along the corresponding eigenvector(s) may reflect rounding errors rather than true structure, and such eigenvectors are discarded, reducing the subspace to that part spanned by eigenvectors having significant eigenvalues.

With the remaining eigenvectors rescaled to make the metric Cartesian, the distinction between covariant and contravariant indices disappears. Further simplification is effected by diagonalizing the revised form of $M_{\mu\nu}$ to give

$$\tilde{S}(x) = S_0 + S_\mu x_\mu - \tfrac{1}{2} x_\mu x_\mu$$
$$\tilde{C}(x) = C_0 + C_\mu x_\mu + \tfrac{1}{2}\gamma_{(\mu)} x_\mu x_\mu \qquad (27)$$
$$l^2 = x_\mu x_\mu$$

where the $\gamma_{(\mu)}$ are the eigenvalues of $M_{\mu\nu}$, and all symbols are defined with respect to the new base vectors.

Apart from the protection against linear dependence, the above procedure is merely the simultaneous diagonalization of $g_{\mu\nu}$ and $M_{\mu\nu}$. Much hangs on accurate diagonalization in the subspace (especially for badly conditioned problems), and it is essential to diagonalize accurately. This is a standard linear algebra problem in a space of low dimension, and reliable algorithms are available for this purpose.

3.7.2 Basic control

The aim of the control procedure is to maximize $\tilde{S}$ over $\tilde{C} = C_{\text{aim}}$ subject to a distance constraint $l^2 \leqslant l_0^2 (\simeq 0.1\Sigma f$ to $0.5\Sigma f)$. Unfortunately, this may be impossible at first. For very many applications, C is a convex (elliptical) function of f, for which all eigenvalues $\gamma_{(\mu)}$ are positive. There is then a minimum value

$$\tilde{C}_{\text{min}} = C_0 - \tfrac{1}{2}\gamma_{(\mu)}^{-1} C_\mu C_\mu \qquad (28)$$

which $\tilde{C}$ can attain in the subspace. Clearly one should not attempt to aim below $\tilde{C}_{\text{min}}$, regardless of the value of C_{aim}. In fact, even attempting to reach values as low as $\tilde{C}_{\text{min}}$ is inappropriate, since the resulting x is then determined purely by the structure of $\tilde{C}$ and not at all by $\tilde{S}$. It is better to set the more modest aim

$$\tilde{C}_{\text{aim}} = \max\left(\tfrac{2}{3}\tilde{C}_{\text{min}} + \tfrac{1}{3} C_0, C_{\text{aim}}\right) \qquad (29)$$

which is always accessible.

The various maxima of $\tilde{S}$ over different values of $\tilde{C}$ may be parameterized by the Lagrange multiplier α in

$$\tilde{Q} = \alpha\tilde{S} - \tilde{C} \qquad (30)$$

(re-defining Q using α instead of λ). Maximizing $\tilde{Q}$ yields

$$x_\mu = (\alpha S_\mu - C_\mu)/(\gamma_{(\mu)} + \alpha) \qquad (31)$$

in which α is chosen to fit $\tilde{C} = \tilde{C}_{\text{aim}}$. The required range for α is the positive range

$$\alpha_{\text{min}} < \alpha < \infty \qquad (32)$$

assigning positive weight to the entropy. α_{min} is normally zero (for positive definite $\nabla\nabla C$), at which C takes its minimum value $\tilde{C}_{\text{min}}$. If, for non-positive $\nabla\nabla C$, any eigenvalues γ are negative, α_{min} becomes max $(-\gamma_{(\mu)})$ at which the increment x_μ diverges. The upper limit $\alpha = \infty$ corresponds to unconstrained maximization of $\tilde{S}$ irrespective of $\tilde{C}$. The value of $\tilde{C}(x)$ increases monotonically in α, for α in the allowed range, so that a simple chop suffices to iterate α towards $\tilde{C} = \tilde{C}_{\text{aim}}$.

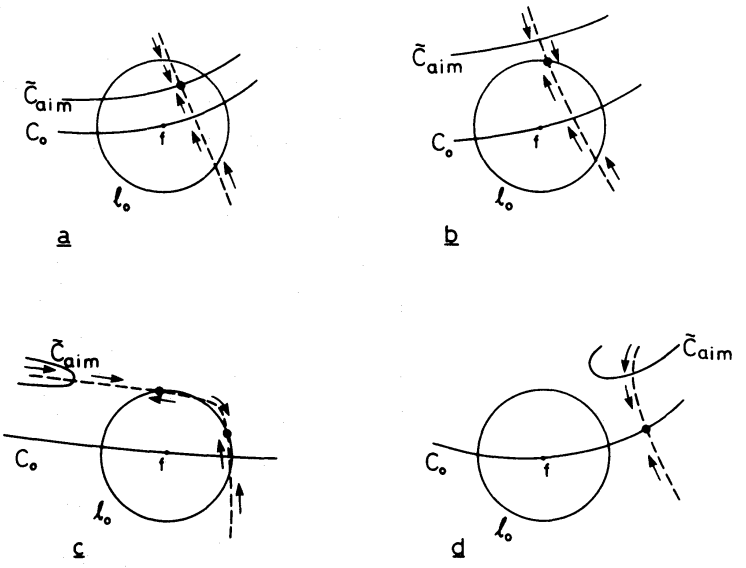

Figure 2. Operation of α-chop in the subspace. The maximum-entropy trajectory, parameterized by α, is shown dashed. The circle centred on the current image f marks the maximum allowed distance l_0. Arrows indicate the direction induced by the chop. Results x are shown as filled circles •. (a) Unique iterate $\tilde{C} = \tilde{C}_{aim}$, $l < l_0$; (b) Unique iterate $\tilde{C} > \tilde{C}_{aim}$, $l = l_0$. (c) Ambiguous iterate $\tilde{C} > \tilde{C}_{aim}$, $l = l_0$. (d) Too distant iterate $\tilde{C} = \tilde{C}_0$, $l > l_0$.

The resulting x may, however, be too large to satisfy the distance constraint. To protect against this, the chop in α is redirected towards $\tilde{C} = C_0$ whenever α gives an increment with too large l^2. The rationale for this form of protection is that maximizing $\tilde{S}$ over the *existing* value C_0 is likely to give a closer iterate than attempting to reach a *different* value $\tilde{C}_{aim}$.

The α-chop normally behaves as in Figs 2a or 2b, and in any case it must always give a result in the range $\tilde{C}_{aim} \leqslant \tilde{C} \leqslant C_0$. Because the distance l is *not* monotonic in α, this can lead to an ambiguity in the result of the chop (Fig. 2c), the result produced depending on the particular values of α actually used in the chop. Nevertheless, the ambiguity is harmless in that either answer for x gives a useful iterate. More seriously, the algorithm may be unable to find *any* sufficiently close value of x (Fig. 2d), especially if the current image f is far from a maximum-entropy image.

3.7.3. Distance penalty

If the α-chop cannot find a sufficiently close value of x, the distance constraint must be introduced explicitly into the maximization via a second Lagrange multiplier P, giving

$$\tilde{Q} = \alpha\tilde{S} - \tilde{C} - Pl^2, \qquad P = \text{distance penalty} \geqslant 0. \tag{33}$$

This is maximized at

$$x_\mu = (\alpha S_\mu - C_\mu)/(P + \gamma_{(\mu)} + \alpha). \tag{34}$$

Thus P can be interpreted as an artificial increase of each eigenvalue $\gamma_{(\mu)}$ of C, giving a revised form

$$\tilde{C}_P(x) = C_0 + C_\mu x_\mu + \tfrac{1}{2}(P + \gamma_{(\mu)})\, x_\mu x_\mu \tag{35}$$

which takes larger values than $\tilde{C}$ itself. $\tilde{C}_P$ is also more convex than $\tilde{C}$, and the maximization of $\tilde{S}$ becomes better conditioned.

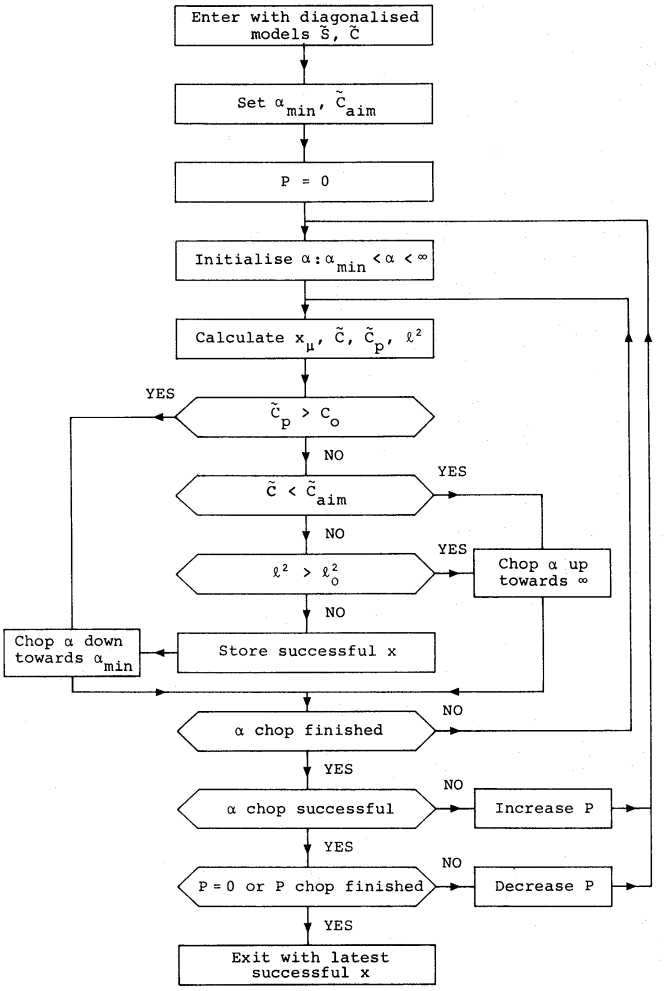

Figure 3. Flow-chart of control procedure.

There are several ways of proceeding from here: we suggest the following. With a distance penalty invoked, α is chopped towards $\tilde{C} = \tilde{C}_{\rm aim}$ as required, but the chop is redirected towards $\tilde{C}_P = C_0$ whenever the distance is too large. Redirecting on $\tilde{C}_P$ rather than on $\tilde{C}$ helps the algorithm because $\tilde{C}$ itself is always less than $\tilde{C}_P$, and hence will make useful progress down towards $\tilde{C}_{\rm aim}$. For sufficiently large penalty P, the α-chop must be able to reach a result satisfying

$$\tilde{C}_{\rm aim} \leqslant \tilde{C} < \tilde{C}_P \leqslant C_0 \tag{36}$$

and the smallest such P is used to give the final result x. A flow-chart of this algorithm is shown in Fig. 3.

All that remains is to increment f by the multiples x_μ of the search directions, whilst protecting against stray non-positive values. The algorithm is complete.

In general, the algorithm proceeds by reducing C at each iteration, whilst keeping close to the maximum of S for the current value of C, until $\tilde{C}_{\rm aim}$ is obtained. Then S is increased, C necessarily remaining at $\tilde{C}_{\rm aim}$, until it is sufficiently close to the maximum that the termination criterion of Section 4 is satisfied.

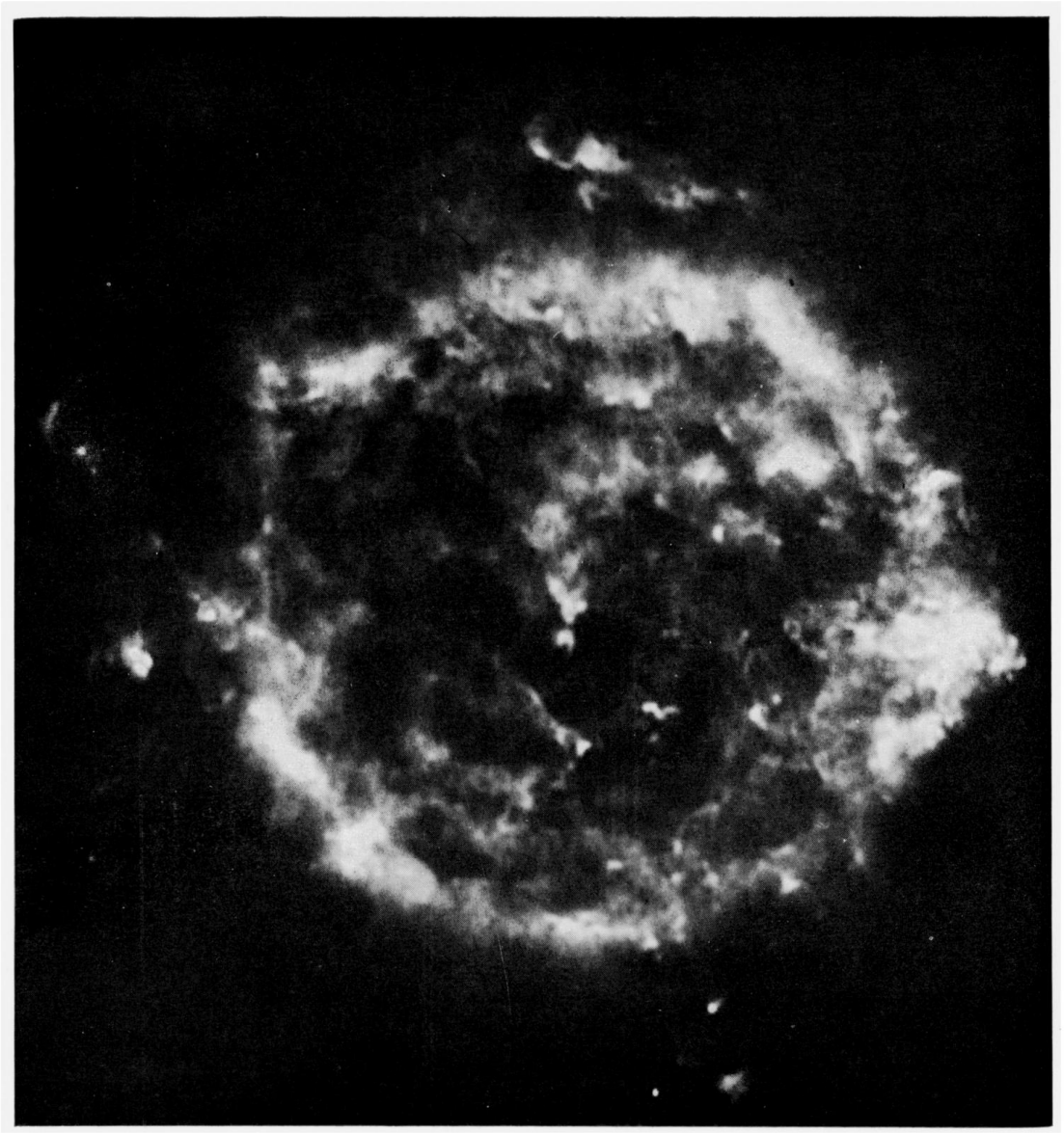

Plate 1. Million-pixel maximum entropy reconstruction of Cas A, computed by the algorithm presented in this paper.

[*facing page* 122]

4 Discussion

Our first conclusion is that it is operationally effective to subsume large numbers of individual constraints (such as positivity of each pixel and acceptable fits to experimental datasets) into single functions like $S(f)$ and $C(f)$. This enables practical algorithms to be developed for large problems.

There are then four main ingredients in the maximum-entropy algorithm recommended in this paper. They are

(1) The use of a subspace of search directions instead of merely using line searches,

(2) Updating fully at each iteration, and *not* attempting to carry information forward, especially information on rapidly changing high derivatives,

(3) The entropy metric, in which the entropy function itself is used to define distances,

(4) Controlling the algorithm directly on the constraint C, and not on its Lagrange multiplier λ.

These ideas may also find use in other non-linear maximization problems. The resulting program has proved highly robust and powerful. It supersedes earlier versions which have been used successfully in image enhancement (Bryan & Skilling 1980; Daniell & Gull 1980), gamma-ray astronomy (Skilling *et al.* 1979), medical tomography (Kemp 1980), and latterly in other fields too. For example, to deconvolve an optical image of the galaxy M87, the algorithm presented here took half the number of iterations of an earlier algorithm (essentially that of Section 3.4) using the same number of search directions (Bryan & Skilling 1980; Bryan 1980). The saving in CPU time was not as great, since more sophisticated search directions, requiring more image-data transforms, are used here. *The operation of the algorithm is always checked* by displaying the value of

$$\mathrm{TEST} = \frac{1}{2}\left|\frac{\nabla S}{|\nabla S|} - \frac{\nabla C}{|\nabla C|}\right|^2 . \tag{37}$$

This measures the degree of non-parallelism between ∇S and ∇C, which is zero for a true maximum entropy image. Usually there is no difficulty in reaching TEST < 0.1 or so at the correct value of C, which demonstrates that the correct, unique maximum-entropy reconstruction has been attained. We think it important that programs purporting to produce maximum-entropy images should make this test of their operation.

Fortran implementations of the algorithm routinely perform maximum-entropy calculations on images up to a million pixels, and there seems no bar in principle to still larger sizes. An example is the 1024×1024 maximum-entropy image of the supernova remnant Cas A (Plate 1), reconstructed from Cambridge 5-km telescope observations at 5 GHz (Gull & Brown, reported in Skilling 1981 and Tuffs 1984): a conventional reconstruction from these data was given by Bell (1977). Almost regardless of size, the algorithm takes something like 20 iterations to reconstruct an image from an experiment with signal-noise of about 100:1. Each iteration of the recommended three search-direction version involves 6 image-data transformations, so that the maximum-entropy reconstruction is about 100 times slower than simple linear reconstructions.

Acknowledgments

During much of the development work reported here, one of us (RKB) was in receipt of a SRC research studentship. Several colleagues, particularly S. F. Gull and G. J. Daniell who introduced us to maximum entropy, also helped to focus our ideas.

References

Ables, J. G., 1974. *Astr. Astrophys. Suppl.*, **15**, 383.
Bell, A. R., 1977. *Mon. Not. R. astr. Soc.*, **179**, 573.
Bryan, R. K., 1980. *PhD thesis*, University of Cambridge.
Bryan, R. K., Bansal, M., Folkhard, W., Nave, C. & Marvin, D. A., 1983. *Proc. Nat. Acad. Sci., USA*, **80**, 4728.
Bryan, R. K. & Skilling, J., 1980. *Mon. Not. R. astr. Soc.*, **191**, 69.
Burch, S. F., Gull, S. F. & Skilling, J., 1983. *Comp. Vis. Graph. Im. Proc.*, **23**, 113–128.
Daniell, G. J. & Gull, S. F., 1980. *IEE Proc. (E)*, **5**, 170.
Fletcher, R. & Reeves, C. M., 1964. *Comp. J.*, **7**, 149.
Frieden, B. R., 1972. *J. opt. Soc. Am.*, **62**, 511.
Frieden, B. R. & Swindell, W., 1976. *Science*, **191**, 1237.
Frieden, B. R. & Wells, D. C., 1978. *J. opt. Soc. Am.*, **68**, 93.
Gull, S. F. & Daniell, G. J., 1978. *Nature*, **272**, 686.
Gull, S. F. & Skilling, J., 1983. *IAU/URSI Symposium on Indirect Imaging*, Sydney, Australia.
Horne, K. D., 1982. *PhD thesis*, California Institute of Technology.
Kemp, M. C., 1980. *International Symposium on Radionuclide Imaging, IAEA-SM-247*, **128**, Heidelberg.
Livesey, A. K. & Skilling, J., 1984. *Acta Crystallogr. A.*, in press.
Minerbo, G., 1979. *Comp. Graph. Im. Proc.*, **10**, 48.
Polak, E., 1971. *Computational methods in optimisation*, p. 44, Academic Press, New York.
Powell, M. J. D., 1976. *Math. Program.*, **11**, 42.
Powell, M. J. D., 1977. *Math. Program.*, **12**, 141.
Sargent, R. W. H., 1974. *Numerical methods for constrained optimisation*, p. 149, eds Gill, P. E. & Murray, W., Academic Press, London.
Scott, P. F., 1981. *Mon. Not. R. astr. Soc.*, **194**, 23P.
Shannon, C. E., 1948. *Bell System Tech. J.*, **27**, 379 & 623.
Shore, J. E. & Johnson, R. W., 1980. *IEEE Trans.*, **IT-26**, 26.
Shore, J. E. & Johnson, R. W., 1983. *IEEE Trans.* **IT-29**, 942.
Skilling, J., 1981. *Workshop on Maximum Entropy Estimation and Data Analysis*, University of Wyoming, Reidel, Dordrecht, Holland, in press.
Skilling, J. & Gull, S. F., 1984. *SIAM Proc. Am. Math. Soc.*, **14**, 167.
Skilling, J., Strong, A. W. & Bennett, K., 1979. *Mon. Not. R. astr. Soc.*, **187**, 145.
Tuffs, R. J., 1984. *PhD thesis*, University of Cambridge.
Wernecke, S. J. & d'Addario, L. R., 1977. *IEEE Trans*, **C-26**, 351.
Willingale, R., 1979. *PhD thesis*, University of Leicester.
Willingale, R., 1981. *Mon. Not. R. astr. Soc.*, **194**, 359.