A FORTRAN routine to estimate a function of two variables from its autocorrelation *

M. Nieto-Vesperinas ^a, F.J. Fuentes ^b, R. Navarro ^c and M.J. Perez-Illarbe ^d

^a Instituto de Ciencia de Materiales, CSIC, Facultad de Ciencias, Universidad Autonoma, Cantoblanco, Madrid 28049, Spain

^b Instituto de Astrofísica de Canarias, 38071 La Laguna, Tenerife, Spain

^c Instituto de Optica, CSIC., Serrano, 121, Madrid, 28006, Spain

^d Departamento de Automática, Universidad Pública de Navarra, Pamplona, Spain

Received 6 November 1991; in revised form 19 April 1993

We describe a simulated annealing routine that yields an estimate of a function of two variables from its autocorrelation. This is equivalent to retrieving the phase from the modulus in Fourier space.

PROGRAM SUMMARY

Title of program: ANNE

Catalogue number: ACPC

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Licensing provisions: none

Computer: VAX 9210

Operating system: MS-DOS 3.3

Programming language used: FORTRAN

Memory required to execute with typical data: 4 Mbytes

No. of bits in a word: 32

No. of lines in distributed program, including test data, etc.: 491

Keywords: phase retrieval, image processing, simulated annealing

Nature of physical problem

Estimation of a function of two variables from its autocorrelation.

Method of solution

Simulated annealing optimization algorithm for finding the zero of a cost function which is constructed from a system of non-linear equations [1,2]

Restrictions on the complexity of the problem

Functions with no more than 64×64 sampling points.

Typical running time

9 minutes for a 32×32 array.

References

- [1] S. Kirkpatrick, C.D. Gelatt and M. Vecchi, Science 220 (1983) 671.
- [2] M. Nieto-vesperinas, R. Navarro and F.J. Fuentes, J. Opt. Soc. Am. A 5 (1988) 30.

Correspondence to: M. Nieto-Vesperinas, Instituto de Ciencia de Materiales, CSIC, Facultad de Ciencias, Universidad Autonoma, Cantoblanco, Madrid 28049, Spain.

* Work supported by the CICYT.

LONG WRITE-UP

1. Introduction

The estimation of a function of two variables from its autocorrelation (which in Fourier space amounts to determining the phase from the amplitude of the Fourier transform), has been previously attempted with partial success only [1,2]. The main difficulty arises from the stagnation of the optimization algorithm in local minima of the corresponding cost function. The complexity of the problem, and therefore the number of local minima, exponentially increases with the size of the array representing the samples of the function to be determined [3], thus making impracticable any of the standard global optimization methods. Specifically, attempts by means of an important class of Newton methods, namely, the Levenberg–Marquardt (LM) algorithm [4] were previously made [3]. An improved code of the LM method given by subroutine LMSTR of the MINPACK-1 library [5] was used. However, it was unable to solve the problem for arrays with more than 6×6 sampling values. These results suggest that the application of Newton, or other descent method, seems hopeless. Even ad hoc algorithms as those of refs. [1,2] often stagnate, although they have manifested to be more powerful than any of the standard global optimization methods. This is so because Fienup's algorithms, like the algorithm to be presented here, allow the cost function to increase sometimes and, thus, one is not likely to get stuck in the first relative minimum encountered.

In refs. [6,7] a *simulated annealing* algorithm for estimating a function of two variables from its autocorrelation was established. It proved to be a very flexible and powerful method. More robust than any other procedure established so far; even though at the expense of a large amount of computing time. At present, this time factor seems to be the main limitation of the algorithm since its convergence is guaranteed and it always yielded a good estimate. The procedure was successfully applied to image reconstruction in astronomy [8].

2. Description

Given the autocorrelation $Q(x, y)$ of a real non-negative function $g(x, y)$ of finite support D :

$$Q(x, y) = \iint g(\xi, \eta) g(\xi + x, \eta + y) d\xi d\eta. \quad (1)$$

The algorithm finds an estimate of $g(x, y)$. It has been shown [9–11] that for functions g of two or more independent variables, the solution to (1) is generally unique, apart from a trivial 180° rotation.

By discretizing the functions Q and g at sampling intervals δx and δy , the problem is equivalent to solving the set of non-linear equations:

$$Q_{ij} = \sum_{m=1}^M \sum_{n=1}^M g_{mn} g_{m+i-M, n+j-M}, \quad \begin{aligned} 1 \leq i \leq 2M-1, \quad 1 \leq m+i-M \leq M, \\ 1 \leq j \leq 2M-1, \quad 1 \leq n+j-M \leq M, \end{aligned} \quad (2)$$

g_{mn} being the M^2 samples $g(\delta x_m, \delta y_n)$ of $g(x, y)$ and Q_{ij} representing the $(2M-1)^2$ sampling points $Q(\delta x_i, \delta y_j)$ of $Q(x, y)$. Since $Q(x, y)$ is even, the set (2) contains $M^2 + (M-1)^2$ independent equations from which the M^2 unknowns g_{mn} are to be found.

The estimated solution to (2) is obtained by means of a *simulated annealing* algorithm that minimizes the cost function, written in the k th iteration as

$$F^{(k)} = \left(\sum_{i=1}^{2M-1} \sum_{j=1}^{2M-1} r_{ij}^{(k)^2} \right)^{1/2}, \quad i = 1, 2, \dots, 2M-1, \quad j = 1, 2, \dots, 2M-1, \quad (3)$$

where the residuals are

$$F_{ij}^{(k)} = Q_{ij} - \sum_{m=1}^M \sum_{n=1}^M g_{mn}^{(k)} g_{m+i-M, n+j-M}^{(k)}, \quad \begin{aligned} 1 \leq i \leq 2M-1, \quad 1 \leq m+i-M \leq M, \\ 1 \leq j \leq 2M-1, \quad 1 \leq n+j-M \leq M. \end{aligned} \quad (4)$$

Details on the foundations of this method can be found in refs. [12–16]. The procedure is based on a theorem stating that if $F(x)$ is a well-behaved function of the N -vector x with a global minimum at the point x^* in a domain S of R^N , then [14–16]:

$$x^* = \min_{T \rightarrow 0} \frac{\int_S x e^{-F(x)/T} dx}{\int_S e^{-F(x)/T} dx}. \quad (5)$$

x^* is then obtained by calculating the “averages” of x with “density of probability” $e^{-F(x)/T}$ at each “temperature” T following a Monte Carlo procedure developed by Metropolis et al. [13]. This is done by generating at each fixed value of the free parameter T an irreducible Markov chain with probability $e^{-F(x)/T}$ such that in the long run the transition probabilities that increase $F(x)$ equal those that decrease $F(x)$, i.e. at that point the configuration of x remains at “equilibrium”. The global minimum x^* is then the limit as $T \rightarrow 0$ of the successive equilibrium configurations of x obtained at each T .

The algorithm is then carried out as follows: Starting with an initial guess function $g_{mn}^{(0)}$, whose corresponding cost function will be $F^{(0)}$, one fixes the value of T . Then one perturbs all the elements of the array $g_{mn}^{(0)}$, one at a time. The perturbation of each element is done by an amount h chosen from a sequence of computer generated random numbers uniformly distributed in the interval $(-\alpha, \alpha)$, α being the *scale of perturbation*. The value of α must be chosen with care, if too large the corresponding cost function F will highly oscillate, or will yield forbidden changes, and if too small the evaluation of the reconstruction will be too slow. The performance of the algorithm is enhanced if α is allowed to decrease as $T \rightarrow 0$ according to a certain schedule. For instance: $\alpha_m = e^{-(m-1)}$ with the index m increasing one unit every seven steps of variation in T . On the other hand, the rate of decay of T may be chosen to be linear, for instance, $T_{n+1} = 0.75 T_n$.

Every time that an element (or sampling value) is perturbed, the cost function $F^{(k)}$ of the result-

ing new array $g_{mn}^{(k)}$ is evaluated. Then one computes the difference

$$\Delta F = F^{(k)} - F^{(k-1)}, \quad (6)$$

where $F^{(k-1)}$ is the cost function of the object prior to the corresponding perturbation of the element. If $\Delta F \leq 0$, the change in the element is accepted and one proceeds to perturb the next element. If $\Delta F > 0$, the change is accepted with probability $\exp(-\Delta F/T)$. This amounts to taking a number n from a sequence of random numbers uniformly distributed in $(0, 1)$. If n is less than $\exp(-\Delta F/T)$, the perturbation is accepted and one passes on to the next sampling value of the function $g(x, y)$. If n is greater than or equal to $\exp(-\Delta F/T)$, the change is not accepted and one proceeds with the next sample of $g(x, y)$. All sampling values of $g(x, y)$ are scanned in this way at a fixed value of T (say, T_n), until “equilibrium” is reached, namely, when on the average the number of trials with increasing E equals the number of trials with decreasing F . Then T is lowered to the value T_{n+1} and the procedure is started again. The larger T , the higher is the probability of accepting a change raising F . Since F is allowed to increase, one is not likely to get stuck in the nearest local minimum. Eventually, when one is near the end of the algorithm at very low values of T , only changes lowering F are accepted.

Both α and T must be slowly lowered in order to avoid stagnation.

A fast computation of the autocorrelation function of the k th guess with an element, located at coordinates (a, b) , and perturbed by an amount h , may be done as follows:

Let $g^{(k)}(\xi, \eta)$ be the guess function prior to the perturbation of this element, then the resulting guess function $g^{(k+1)}(\xi, \eta)$ after this element has been perturbed will be $g^{(k)}(\xi, \eta) + h \delta(\xi - a, \eta - b)$; therefore, the autocorrelation function of $g^{(k+1)}(\xi, \eta)$ will be

$$\begin{aligned} Q^{(k+1)}(x, y) \\ = \iint g^{(k+1)}(\xi, \eta) g^{(k+1)}(\xi + x, \eta - y) d\xi d\eta \end{aligned}$$

Fig. 1. Block diagram of the simulated annealing algorithm.

$$\begin{aligned}
&= \iint [g^{(k)}(\xi, \eta) + h\delta(\xi - a, \eta - b)] \\
&\quad \times [g^{(k)}(\xi + x, \eta + y) \\
&\quad + h\delta(\xi + x - a, \eta + y - b)] d\xi d\eta. \quad (7)
\end{aligned}$$

Performing the operations in the integrand of (7) we obtain

$$\begin{aligned}
Q^{(k+1)}(x, y) &= Q^{(k)}(x, y) + hg^{(k)}(a - x, b - y) \\
&\quad + hg^{(k)}(a + x, b + y) + h^2\delta(x, y). \quad (8)
\end{aligned}$$

In eq. (8) $Q^{(k+1)}(x, y)$ is the autocorrelation resulting from adding an amount h , $-\alpha \leq h \leq \alpha$, to the sample in (a, b) of the k th iteration $g^{(k)}(x, y)$. $\delta(x, y)$ is the δ -function representing the perturbed sampling point.

A block diagram of the algorithm, performed by the subroutine ANNE in the program, is shown in fig. 1. With the aid of an image monitor the samples of the successive iterations $g^{(k)}(x, y)$ can

be seen as pixels of a picture of the estimated object function $g(x, y)$.

The following example has been solved on a VAX 9210. All computations were done in single precision arithmetic and T vary as described above:

α , semi-interquartile (RINT in the program)

h , perturbation (RINC in the program)

The probability density of h is then the Cauchy distribution [17]:

$$P(h) = \frac{\alpha}{\alpha^2 + h^2}, \quad (9)$$

$$T_{n+1} = T_n - \delta T_n, \quad (10)$$

where [18]:

$$\delta T_n = \frac{\epsilon T_n}{\tau(T_n)}, \quad (11)$$

$$\tau(T_n) = \exp\left(\frac{CT_{\text{initial}}}{T_n}\right), \quad (12)$$

Fig. 2. Contour lines of the 32×32 test array.

Fig. 3. Contour lines of the solution.

T_{initial} being the starting temperature. In the program ϵ is the parameter RK.

The system then has a time τ to respond to a decrease of temperature given by ϵ .

The CPU time was about 9 minutes for estimating the 32×32 array on the VAX. It should be remarked, however, that all simulated annealing algorithms, although efficient, are computa-

Fig. 4. Evolution of the cost function.

Fig. 5. Evolution of the temperature.

tionally costly. The computing time may be minimized by saving as many redundant scan perturbations as possible near "equilibrium". The initial value of T was $T = 1$. The final value $T = 9 \times 10^{-4}$ was obtained by lowering T in 1000 steps.

3. Test run

Figure 2 shows the contour lines of a 32×32 array corresponding to the sampling values of the test function to be estimated from its autocorrelation. The initial guess is a constant array: GUESS $(i, j) = 8$. The result and performance of the algorithm done by subroutine ANNE, given by the cost functions of both the initial guess and result, are shown in fig. 3. The estimation is almost perfect apart from a slight shift and the 180° rotation. The evolution of the cost function in this test is shown in fig. 4.

The evolution of the temperature is shown in fig. 5.

Finally, it should be remarked that once a "temperature" $T = 0$ has been reached, one can obtain further improvements of the result as required, by repeating several times the algorithm, with very low values of T at the start.

Acknowledgements

Work supported by the CICYT under grant PB 0278.

References

- [1] J.R. Fienup, Phase retrieval algorithms: a comparison, *Appl. Opt.* 21 (1982) 2758.
- [2] J.R. Fienup and C.C. Wackerman, Phase retrieval stagnation problems and solutions, *J. Opt. Soc. Am. A* 3 (1986) 1897.
- [3] M. Nieto-Vesperinas, A study on the performance of non-linear least-square optimization methods in the problem of phase retrieval, *Opt. Acta* 33 (1986) 713.
- [4] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Non-Linear Equations* (Prentice Hall, New York, 1983).
- [5] J.J. Moré, B.S. Garbow and K.E. Hillstom, User Guide for MINPACK-1, Argonne National Laboratory Report ANL-80-74 (1980).
- [6] M. Nieto-Vesperinas and J.A. Méndez, 'Phase retrieval by Monte Carlo methods. *Opt. Commun.* 59 (1986) 249.
- [7] M. Nieto-Vesperinas, R. Navarro and F.J. Fuentes, Performance of a simulated annealing algorithm for phase retrieval. *J. Opt. Soc. Am. A* 5 (1988) 30.
- [8] R. Navarro, F.J. Fuentes and M. Nieto-Vesperinas, 'Simulated annealing image reconstruction in photonlimited stellar speckle interferometry', *Astron. Astrophys.*, in press.
- [9] Y.M. Bruck and L.G. Sodin, On the ambiguity of the image reconstruction problem. *Opt. Commun.* 30 (1979) 304.
- [10] M.H. Hayes and J.H. McClellan, Reducible polynomials in two or more variables, *Proc. IEEE* 70 (1982) 197.
- [11] M.H. Hayes, The unique reconstruction of multidimensional sequences from Fourier transform magnitude or phase, in: *Image Recovery: Theory and Application*, H. Stark, ed. (Academic New York, 1987) ch. 6, pp. 195-230. J.L.C. Sanz and T.S. Huang, Unique reconstruction of a bandlimited multidimensional signal from its phase or magnitude, *J. Opt. Soc. Am.* 73 (1983) 1446.
- [12] S. Kirkpatrick, C.D. Gelatt and M. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671.
- [13] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, Equation of state calculations by fast computing machines *J. Chem. Phys.* 21 (1953) 1087.
- [14] M. Pincus, A closed form solution of certain programming problems, *Oper. Res.* 16 (1968) 690; A Monte Carlo method for the approximate solution of certain types of constrained optimization problems, *oper. Res.* 18 (1970) 1225.
- [15] J.M. Hammersley and D.C. Handscomb, *Monte Carlo Methods* (Methuen, London, 1967) p. 113.
- [16] R.Y. Rubinstein, *Simulation and the Monte Carlo Method* (Wiley, New York, 1981) p. 260.
- [17] H. Szu and R. Hartley, Fast simulated annealing, *Phys. Lett. A* 122 (1987) 157.
- [18] S. Rees and R.C. Ball, Criteria for optimum simulated annealing schedule for problems of the travelling salesman type. *J. Phys. A* 20 (1987) 1239.