US /GR BK/

The MODEX Series

Model-fitting Programs

for Interferometer Observations

G. Purcell

Index

The MODEX Series

Model-fitting Programs
for Interferometer Observations

Introduction

The MODEX series comprises four programs -- MODEX1, MODEX2, MODEX3, and MODEX4 -- written in FORTRAN for the IBM 360 computer. The programs are used to recover the brightness distributions of celestial radio sources from interferometric observations, and they differ among themselves in the kind of observations upon which they operate as well as in the computational methods they employ.

MODEX1 and MODEX2 were the first of the programs to be written. Ordinarily they are not practical methods for dealing with an actual problem, and I have included them in this discussion solely for their didactic value -- as examples of alternate ways in which the basic problem may be attacked. MODEX3 and MODEX4 are identical except in procedural details; I shall usually refer to them together as MODEX3/4.

All the programs model the true brightness distribution of a source as a group of point sources distributed over the region occupied by the actual source at fixed points specified in advance by the programmer. MODEX1 could be used when both the relative amplitude and the absolute phase of the source's complex visibility function are known at each point of observation in the brightness transform plane (the u-v plane). MODEX2 operates on amplitude information alone, by linearizing the equations in the unknown amplitudes of the point sources. MODEX3/4 proceed in a similar but distinct manner, using the "method of steepest descent."

Using the Programs: An Operational Summary

Input Requirements:

The organization of the input data has been made as nearly identical for all the programs as possible. It consists of the following parts:

1) A single card containing general information about the source and the model, along with certain parameters.
2) A set of data cards each containing a single observation point.
3) A set of cards assigning the positions of the model points.
4) Optionally, in MODEX2-4, a set of cards specifying initial values of the amplitudes of the model points.

Following is a detailed description of these input cards:

1a) The General Information Card (all programs):

The source name (columns 5-12) -- any eight characters identifying the source.

The source declination (col. 15-28) -- expressed in degrees, minutes, and seconds to the nearest tenth, with a single space separating each of the three subfields.

An identification number for the run (col. 29-33) -- a five-digit integer used to distinguish, for example, different sets of computations for the same source.

The number of observation points (col. 37-40) -- an integer indicating the number of cards in the second part of the input data (which is the same as the number of observation points since, as we shall see, each card will contain one point). In MODEX1/2 this number may be no larger than 500, but in MODEX3/4 the maximum value is 1100.

The number of model points (col. 45-47) -- an integer specifying the number of point sources in the proposed model, or, equivalently, the number of cards in the third part of the input (except in MODEX4). In MODEX1/2 this number may not exceed 140, but in MODEX3/4 it may be as large as 700.

1b) The General Information Card (additional parameters required by MODEX2-4 only)

Flux (col. 52-63) -- six significant figures giving the total flux responsible for the measured visibility amplitudes (and expressed in the same units, of course). This datum is used only when the fourth part of the input is absent.

A tolerance parameter (col. 64-69) -- three significant figures. Since MODEX2-4 are iterative procedures, there must be a criterion according to which the program may determine when to stop iterating. In MODEX2, iteration stops whenever the fractional change in each of the model-point amplitudes is less than the tolerance parameter on a particular iteration. In MODEX3/4 iteration halts when the fractional change in the mean square deviation of the observed visibilities from those which would result from the current model becomes less than the tolerance parameter.

The maximum number of iterations (col. 70-72) -- a three-digit integer declaring the maximum allowable number of iterations. After this number of iterations processing will stop even though the condition based on the tolerance parameter is not yet satisfied.

An option selector (col. 79-80) -- an integer between 1 and 8, inclusive, specifying whether or not the program is to use each of the three following options:

1) The program may use both amplitude and phase information as input, or amplitudes only.
2) The program may or may not read in an initial set of model amplitudes.
3) If a model amplitude happens in the course of iteration to become negative, it may or may not be reset immediately to zero.

The eight possibilities are listed in the following table:

Parameter	Amps. and Phases?	Initial Model?	Reset?
1	No	No	No
2	No	No	Yes
3	No	Yes	No
4	No	Yes	Yes
5	Yes	No	No
6	Yes	No	Yes
7	Yes	Yes	No
8	Yes	Yes	Yes

2) The Set of Observation Cards

In MODEX1 each "observation point" is the amplitude and phase of the complex visibility function (not necessarily normalized), at a given hour angle of the source, on a particular baseline. In MODEX2 the amplitude alone is used, and in MODEX3/4 either amplitudes only or amplitudes and phases together may be used. Each card is organized as follows:

The baseline parameter (col. 11-12) -- an integer from 1 to 10, inclusive, designating the baseline on which the observation was made. At present the baseline parameters 1,2,...,6 refer to the baselines named the same way (alternately called baseline 12,15,18,21,24, and 27, respectively). The four other possible parameters are not now being used.

The hour angle of the source (col. 18-31) -- expressed in hours, minutes, and seconds to the nearest tenth. A single blank space must separate each of the three subfields.

The relative amplitude of the complex visibility (col. 38-47) -- five significant figures, in any convenient units.

The corresponding phase of the complex visibility (col. 48-59) -- four significant figures, expressed in degrees. This datum is necessary only in MODEX1 and is optional in MODEX3/4. The phases must be absolute.

3a) The Model Specification Cards (except for MODEX2C and MODEX4)

These cards give the coordinates ~~of the points~~ of the points at which sources are to be located. In MODEX1 the origin of coordinates should be put as near to the supposed centroid of the source as possible, but in the other programs the location of the origin is not so important. Each card specifies one model point, consisting of two coordinates:

The x-coordinate (col. 11-20) -- three significant figures giving the distance east of the origin in seconds of arc.

The y-coordinate (col. 21-30) -- three significant figures giving the distance north of the origin in seconds of arc.

3b) The Model Specification Cards (for MODEX2C and MODEX4)

In these two programs the model points are specified not individually but in rectangular blocks consisting of points on a square grid. A block is determined by its center point, its half-widths along the x- and y-coordinate axes, and its grid spacing. As many as ten blocks may be declared, one to a card, as follows:

The x-coordinate of the center of the block (col. 1-10) -- three significant figures giving the distance east of the origin, in seconds of arc.

The y-coordinate of the center of the block (col. 11-20) -- three significant figures giving the distance north of the origin, in seconds of arc.

The half-width of the block in the x-direction (col. 21-30) -- three significant figures; seconds of arc.

The half-width of the block in the y-direction (col. 31-40) -- three significant figures; seconds of arc.

The grid spacing of the block (col. 41-50) -- three significant figures; seconds of arc. (Notice that the grids of different blocks may be of different sizes. Model points will be placed at all lattice points of the grid within or on the boundary of the region determined by the two half-widths.

4) The Initial Model (MODEX2-4 only)

If it is desired to specify an initial source model, from which iteration is to begin, the set of data cards containing the model amplitudes must be placed at the end of the input data deck. Each card (except, perhaps, for the last) will contain six amplitudes, each of five significant figures, in columns 5-16, 17-28, 29-40, 41-52, 53-64, and 65-76. If the source model is specified point by point, then each amplitude in the initial model will be associated with its ordinal counterpart among the source points. When the source model is specified in blocks, all the points in the first block come first, starting with the most northwesterly point and proceeding east along the northernmost row to the end, then taking the next row south in the same way, and so on, to the southernmost row and finally the most southeasterly point. Then comes the next block, arranged in the same order, and so on.

Output:

The output of each of the programs is largely self-explanatory. Here is a summary of what each program will produce:

MODEX1

1. The first line identifies the source and the run number.
2. There follows a list of the model points. For each point there will be given an identification number, its x- and y-coordinates, and its calculated amplitude.
3. There will then be a numbered list of the observation points, giving for each point the observed [illegible]

visibility amplitude, the amplitude due to the model distribution, the observed phase of the visibility, and the phase derived from the model.

4. Finally the machine will print the rms deviations of the model phases and amplitudes from the observed values.

<u>MODEX2</u>

1. The first line identifies the source and the run number.
2. For each iteration, the iteration number is printed, followed by a list of the model points giving the coordinates of each along with its amplitude at the end of that iteration.
3. After the last iteration a message is printed explaining why iteration was halted.
4. Then there comes a numbered list of the observation points, giving the observed visibility amplitude of each along with the amplitude and phase calculated for that point from the model.
5. The last line of printed output gives the rms deviation of the model amplitudes from the observed amplitudes.
6. After all the printed output has been produced, the machine punches on cards the final source model. This model is in the same format as the

input model (part 4 of the input), so that the deck may be used on a subsequent run to continue iteration from the same point.

MODEX3/4

The output from MODEX3/4 is practically identical to that of MODEX2, except for point 5. In both MODEX3 and MODEX4 the rms deviation of the model is printed after each iteration except the last. In MODEX3 the deviation printed is the deviation of the model amplitudes; but in MODEX4 the number given is the rms value of the distance in the complex visibility plane between the observed visibilities and the model visibilities.

Detailed Description of the Computational Methods

MODEX1 Mathematical Background

The true brightness distribution and the complex visibility function comprise a Fourier transform pair:

$$\mathcal{V}(u,v) = V e^{i\sigma} = \iint_{-\infty}^{\infty} T(x,y)\, e^{i2\pi(ux+vy)}\, dx\, dy \qquad (1)$$

$$T(x,y) = \iint_{-\infty}^{\infty} \mathcal{V}(u,v)\, e^{-i2\pi(ux+vy)}\, du\, dv \qquad (2)$$

Since $\mathcal{V}$ is not completely known, it is impossible to restore the brightness exactly -- that is, to calculate T(x,y) unequivocally. But by assuming that the brightness distribution can be modeled as an array of point sources disposed at pre-assigned points in the x-y plane, one can lift part of the "degeneracy" in the brightness and at the same time simplify the mathematics to a manageable level.

Suppose, then, that

$$T(x,y) = \sum_{k=1}^{K} b_k\, \delta(x - x_k, y - y_k) \qquad (3)$$

so that,

$$\mathcal{V} = \sum_{k=1}^{K} b_k\, e^{i2\pi(ux_k + vy_k)} \qquad (4)$$

Now, in MODEX1, V and σ are both known at every point (u_q, v_q) in the transform plane at which an observation has been made. Let the phase of the kth term in $\mathcal{V}$ (eq.(4)) at point "q" be called $\theta_{kq} = 2\pi(u_q x_k + v_q y_k)$. Then, if σ_q is the observed phase of the visibility at the same point "q", the observed visibility amplitude at that point is just the sum of the projections of its component vectors, namely $b_k e^{i2\pi(u_q x_k + v_q y_k)} = b_k e^{i\theta_{kq}}$, on the line $\phi = \sigma_q$ in the u-v plane. That is,

$$V_q = \sum_{k=1}^{K} b_k \cos(\sigma_q - \theta_{kq}) \quad ; \quad q = 1, 2, \ldots, Q \qquad (5)$$

where the total number of observation points has been called Q.

These equations are linear in the b_k, and in order to establish a solution we must have $Q \geq K$.

Now, imposing the least-squares condition on eq.(5),

$$\sum_q \left[\sum_k b_k \cos(\sigma_q - \theta_{kq}) - V_q \right]^2 = \text{minimum} \Rightarrow \qquad (6)$$

$$\frac{\partial}{\partial b_\ell} \sum_q \left\{ \left[\sum_k b_k \cos(\sigma_q - \theta_{kq}) \right]^2 - 2 \sum_k b_k \cos(\sigma_q - \theta_{kq}) V_q \right\} = 0 ; \qquad (7)$$

$$\ell = 1, 2, \ldots, K$$

or,

$$\sum_q \left\{ \cos(\sigma_q - \theta_{\ell q}) \left[\sum_k b_k \cos(\sigma_q - \theta_{kq}) - V_q \right] \right\} = 0 \qquad (8)$$

or, rearranging,

$$\sum_k b_k \sum_q \cos(\sigma_q - \theta_{\ell q}) \cos(\sigma_q - \theta_{kq}) = \sum_q V_q \cos(\sigma_q - \theta_{\ell q}) ; \qquad (9)$$

$$\ell = 1, 2, \ldots, K$$

If $Q \geq K$, one can hope to have an independent set of equations and to solve for the b's.

MODEX1

Program Summary

ISN	
5-9	Read in input, convert phases from degrees to radians.
10-15	Compute (u,v) for each observation point; an extra factor 3.041674 is inserted so that when u and v are multiplied by x and y, expressed in seconds of arc, the result will be converted to radians x 2π
16-24	Clear each element in the upper-right half of the matrix of coefficients of the b's; form $\sum_q \cos(\sigma_q - \theta_{\ell q})\cos(\sigma_q - \theta_{kq})$ and $\sum_q V_q \cos(\sigma_q - \theta_{\ell q})$ of eq.(9) in situ in the appropriate matrix elements.
25-27	Fill in the lower-left half of the matrix by symmetry.
28-32	Solve the equations (9) for the amplitudes of the point sources; write out the source name and run number; if the equations are inconsistent, redundant, or poorly conditioned, stop.
33-42	Compute amplitude and phase according to the model at each observation point; convert phases from radians to degrees.
43-53	Compute rms deviations of model amplitudes and phases, and finish writing the output.

MODEX2 Mathematical Background

In MODEX2 only the amplitude of the complex visibility is known, not the phase. Suppose that we have an approximate solution for the b_k; call it β_i ; i= 1,2,...,K. Then at each observation point, "q," one can compute

$$\alpha_q \triangleq \left\| \sum_k \beta_k e^{i2\pi(u_q x_k + v_q y_k)} \right\| - V_q \tag{10}$$

α_q is obviously the "error" in the visibility amplitude computed from the model at point q. Let us try to find some incremental changes, Δ_i , that can be made in the β_i to make the errors smaller. If the equations were linear in the β's, and the number of parameters, K, to be varied were exactly equal to the number of indepent errors, Q, to be removed, then we could simply write the equations

$$\sum_{k=1}^{K} \frac{\partial \alpha_q}{\partial \beta_k} \Delta_k = -\alpha_q \; ; \quad q = 1, 2, \ldots, Q \tag{11}$$

Then, knowing $\frac{\partial \alpha_q}{\partial \beta_k}$ and α_q for all k and q, we could solve for the Δ_k. In that case the change in α_q due to the change Δ_k in β_k is $\frac{\partial \alpha_q}{\partial \beta_k} \Delta_k$, and since the sum of all the changes is $-\alpha_q$, the application of all the increments makes all the $\alpha_q = 0$. In other words, the model reproduces the observed visibility amplitudes exactly.

But actually the equations (10) are not linear in the β's. Still, if the initial model is a fair approximation to the true solution values, the equations may be so nearly linear in the neighborhood of the model and the solution that the correct values of the β's will be approached after a few iterations of the same procedure. This method is simply a generalization of Newton's Method to a function of several variables.

Now suppose that in addition there are random errors in the V_q and that there are more data points than amplitudes to be determined.-- that is, $Q > K$. It is possible to incorporate these features into the previous scheme by solving the set of Q linearized equations for values of the Δ_k satisfying the least-squares criterion.

(Intuitively, the justification for this procedure is not at all clear: inasmuch as it is impossible to find β_i such that all $\alpha_i = 0$ (The equations are over-constrained.), it is not at all apparent that the least-squares values of Δ_i determined on succeeding iterations will converge to zero or even become less than a fixed absolute value. For example, one might reasonably imagine that after a few iterations the Δ_i would begin to oscillate at a more or less constant amplitude, negative increments following positive ones, and <u>vice versa</u>, indefinitely, with the values of the β_i being neither much improved nor much degraded in the process.

The trials which have been made with this particular problem show, however, that the procedure does work, even when the initial model is not a very good one.)

Let us develop the equations. Thinking again of the visibility vector in the complex plane as being the sum of vectors due to each of the point brightness sources, you can easily see that

$$\frac{\partial \alpha_q}{\partial \beta_k} = \cos(\sigma_q - \theta_{kq}) \tag{12}$$

and applying the least-squares condition along with eq. (12) to eqs. (11) gives us K equations in the b_k:

$$\sum_{k=1}^{K} \Delta_k \sum_{q=1}^{Q} \cos(\sigma_q - \theta_{kq}) \cos(\sigma_q - \theta_{\ell q}) = -\sum_{q=1}^{Q} \alpha_q \cos(\sigma_q - \theta_{kq}) \tag{13}$$

$$\ell = 1, 2, \ldots, K$$

Having solved for the Δ_k , we simply increment the β_k accordingly, recompute the α's from the (hopefully improved) β_k , and continue in the same pattern until the agreement between the computed and observed visibilities is sufficiently good.

There are three versions of MODEX2. All of them use the algorithm outlined here and produce the same results, but they differ in the following respects: In MODEX2A certain coefficients involving trigonometric functions which are used on every iteration are stored in scratch areas on the disks, whereas in MODEX2B/C these coefficients are recomputed at every iteration. MODEX2C differs from MODEX2B in that the source model is read in in blocks rather than as separate points in MODEX2C. This method of organizing the data later permits a great saving of time in the computation of trigonometric functions, as will be seen in the following analysis of all three programs.

MODEX2A/B/C

Program Summary

ISN MODEX2A	ISN MODEX2B	ISN MODEX2C	
--5-6	--	--	Rewind the "scratch tapes".
7-9	5-7	6-7	Read in the first part of the input data.
---	---	8-26	Read in the blocks of model points and compute coordinates for all the points.
10-14	8-12	27-31	Set the value of a constant to be used later; preset the model points to a uniform value, then read in the initial model if one is given.
15-20	13-18	32-37	Compute (u,v) for each observation point; an extra factor of 3.046174×10^{-5} is inserted so that when u and v are multiplied by x and y, in seconds of arc, the seconds will be converted to radians x 2π .
21-27	--	--	Using u and v, compute the phase associated with each model point for each observation point; store the phases, their sines and cosines.
28-32	19-23	38-42	Write out the source name and run number, set the iteration number to zero; increment it; if the maximum allowable number of iterations has been made, stop.

33-34	--	--	Rewind the tapes.
35-38	24-27	43-46	Clear the augmented matrix of coefficients of the Δ_i .
--	--	47-72	At each observation point compute the sine and cosine of the phase angle of the contribution to the visibility from each of the point sources.
39-58	28-45	73-87	For each observation point compute V_q and σ_q ; compute the factors in eq.(13) and form $\sum_{q=1}^{Q} \cos(\sigma_q - \theta_{kq}) \cos(\sigma_q - \theta_{\ell q})$ and $-\sum_{q=1}^{Q} \alpha_q \cos(\sigma_q - \theta_{kq})$ <u>in situ</u> in the appropriate matrix elements, using the symmetry of the matrix to fill in the elements below the principal diagonal.
59-62	46-49	88-91	Solve the equations for the Δ's ; if no solution is possible, stop.
63-76	50-63	92-105	Increment the β's according to the values of the Δ's . Under ordinary circumstances the increment will be equal to the corresponding Δ . But during the first few iterations one may expect erratic behavior, so in this program the first three iterations are given special attention. During these iterations a value

			of β is not permitted to decrease below zero or to increase by an amount larger than twice its present value or half the average flux of a point source, whichever is larger.
77-83	64-70	106-112	Write the results of the iteration just completed; if the changes in the source model have been considerable, begin another iteration; otherwise proceed to the last part of the program.
84-87, 91-106	71-72, 76-90	113-114, 118-132	Compute the visibility amplitude and phase at each observation point according to the final model; compute the rms deviations of the model amplitudes from the observed amplitudes; write out these data, along with the final model.

MODEX3/4   Mathematical Background

MODEX3/4 applies the method of steepest descent to the restoration. This technique is similar in conception to Newton's Method (in that it involves a linearization of the equations to be solved), but it is rather different in execution.

Define a function W as follows:

$$W(\vec{b}) = \sum_{q=1}^{Q} \left[V_q'(\vec{b}) - V_q \right]^2 \qquad (14)$$

where $\vec{b} = [b_1, b_2, \ldots, b_k] = \sum_{k=1}^{K} \hat{b}_k b_k$, a vector whose components are the amplitudes of the point sources in the source model, arranged in some arbitrary fixed order; V_q is as usual the observed visibility amplitude at point q in the u-v plane, and V_q' is the amplitude calculated at the same point q from the source model $\vec{b}$. Our objective is to minimize W; that is, we want to find an argument, $\vec{b}$, of W such that the root mean square deviation of the model amplitudes from the observed amplitudes is a minimum.

In principle the approach is easy. At an initial point $\vec{b}_I$ in b-space, W increases most rapidly along the vector $\vec{\nabla}_b W(\vec{b}_I)$. To decrease W then, we should set out in the direction $-\vec{\nabla}_b W(\vec{b}_I)$. Of course, since W is not a linear function of the b's, its value will not continue indefinitely to decrease as we move in the chosen direction. Nor is it likely that a local minimum of the function will occur on the line. But what we can do is to move along the line until we find the point on the line at which the function reaches its minimum value. Calling this point $\vec{b}_{II}$, we can then move from $\vec{b}_{II}$ as origin along the line $-\vec{\nabla} W(\vec{b}_{II})$ in the same fashion to a point $\vec{b}_{III}$, and so on, until we approach an actual local minimum of the function $W(\vec{b})$.

In practice it may be difficult to compute the exact gradient, or to find the precise point along the line at which the function being examined is a minimum. Ordinarily, though, a first approximation is good enough in both cases. In MODEX3/4 the gradient is computed exactly, and the minimum point is approximated. Let me be specific:

From eq. (14),

$$\vec{\nabla}_b W(\vec{b}_I) = \sum_{k=1}^{K} \hat{b}_k \left[2 \sum_{q=1}^{Q} \left(V_q'(\vec{b}_I) - V_q \right) \frac{\partial V_q'(\vec{b}_I)}{\partial b_k} \right] \tag{15}$$

We know that $\partial V_q'(\vec{b}_I)/\partial b_k = \cos \theta_{kq}$, so

$$\vec{\nabla}_b W(\vec{b}_I) = 2 \sum_{k=1}^{K} \hat{b}_k \left[\sum_{q=1}^{Q} \left(V_q'(\vec{b}_I) - V_q \right) \cos \theta_{kq} \right] \tag{16}$$

Now we need to find the positive constant, c, such that $W(\vec{b}_I - c\vec{\nabla}_b W(\vec{b}_I))$ is a minimum. To do this we choose two trial values for c in the way discussed in the succeeding paragraphs. Call them c_1 and c_2. Then we compute the function $W(\vec{b}_I - c\vec{\nabla}_b W(\vec{b}_I))$ for c_1 and c_2. Considering W as a function of c, we can now use the three known values of W -- namely W(0), $W(c_1)$, and $W(c_2)$ -- to find a quadratic function, W'(c), such that W'(0) = W(0), $W'(c_1) = W(c_1)$, and $W'(c_2) = W(c_2)$. From the function W' we can easily find the argument c_0 such that $W'(c_0)$ is a minimum. (Look at Figure 1.) If we assume that W'(c) is nearly equal to W(c) over the

Figure 1

Using W(0), W(c_1), and W(c_2) to find c_0

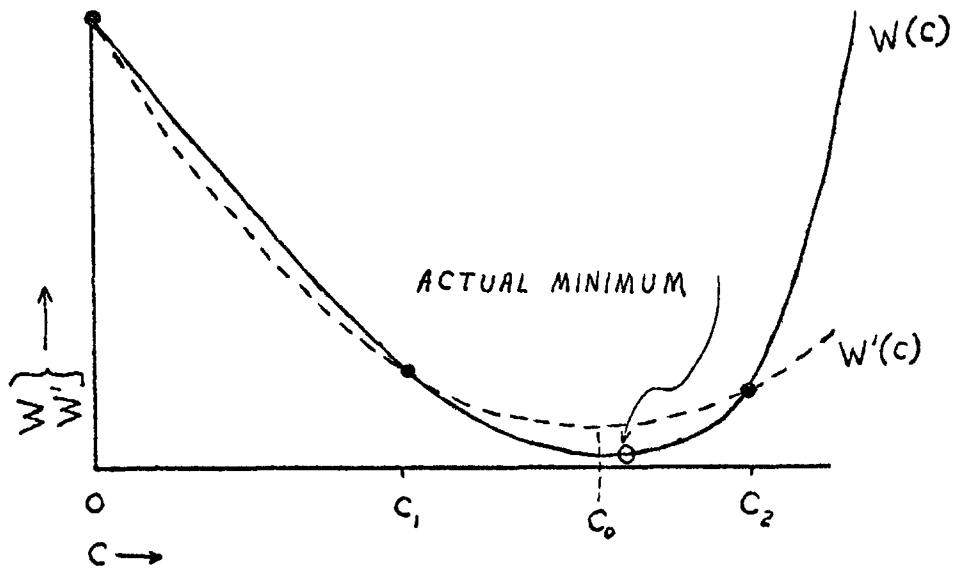
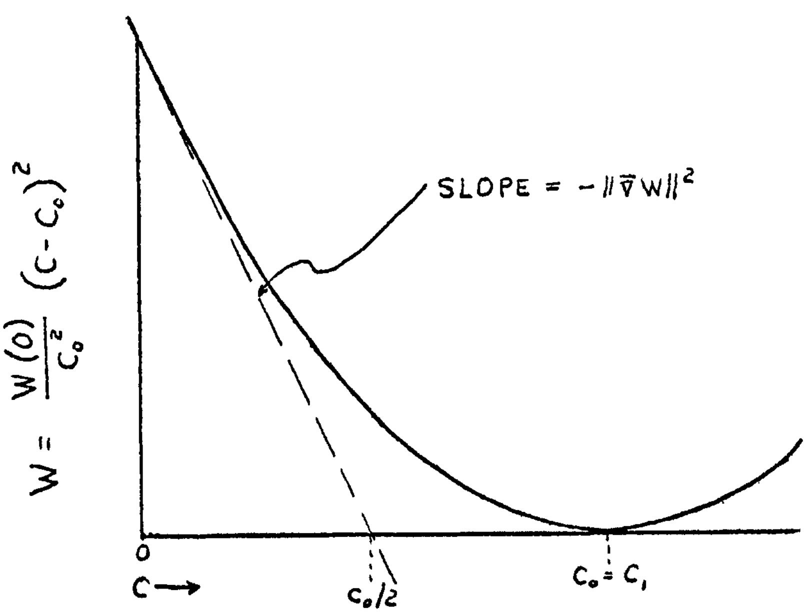


Figure 2

Computing c_1



entire range of arguments under consideration, then $W(c_0)$ will be nearly the minimum value of the function along the gradient line. Therefore we should use $\vec{b}_{II} = \vec{b}_I - c_0 \vec{\nabla}_b W(\vec{b}_I)$ as the next source model and treat it in the same way as $\vec{b}_I$ to obtain an even better model, $\vec{b}_{III}$, etc.

The problem of choosing c_1 and c_2 remains to be dealt with. Suppose that $c_1 < c_2$. Then it seems intuitively that the arguments 0, c_1, and c_2 should be approximately equally spaced, and that c_0 should be not much larger (and preferably smaller) than c_2, as in Figure 1. Now suppose that $W(c)$ is a quadratic function having the minimum value $W(c_0) = 0$, as in Figure 2. We are at $c = 0$, and we know $W(c = 0)$ and $\vec{\nabla} W(c = 0)$. How can we choose c_1 so that $W(c_1) = 0$ (that is, so that the minimum point is exactly in the center of the interval $(0, c_2)$?

We know from elementary vector calculus that (in general!),

$$\frac{dW(\vec{b})}{ds} = \frac{d\vec{s}}{ds} \cdot \vec{\nabla}_b W(\vec{b}) \tag{17}$$

where $d\vec{s} = \hat{b}_1\, db_1 + \hat{b}_2\, db_2 + \cdots + \hat{b}_K\, db_K = d\vec{b}$,and (18a)

$$ds = \| d\vec{s} \| = \left(db_1^{\,2} + db_2^{\,2} + \cdots + db_K^{\,2} \right)^{1/2} \tag{18b}$$

We have decided that $d\vec{b} = -dc\, \vec{\nabla}_b W(\vec{b})$, or

$$ds = dc\, \| \vec{\nabla} W \| \tag{19}$$

Substituting for $d\vec{s}$ and ds in eq.(17) gives,

$$\frac{dW}{dc} = -\|\vec{\nabla} W\|^2 \tag{20}$$

Knowing $W(0)$, $\left(\frac{dW}{dc}\right)_{c=0}$ and $W(c_1)=0$ for the parabola in Figure 2, we can now get c_1. The algebra is simple, and it turns out that

$$c_1 = \frac{2W(0)}{\|\vec{\nabla} W(0)\|^2} \tag{21}$$

A considerable amount of practice with this method of calculating c_1 has shown that the procedure is a good one, despite the fact that $W(c)$ is not even approximately quadratic over any wide range of c, and that $W(c_0)$ usually turns out to be at least half as big as $W(0)$.

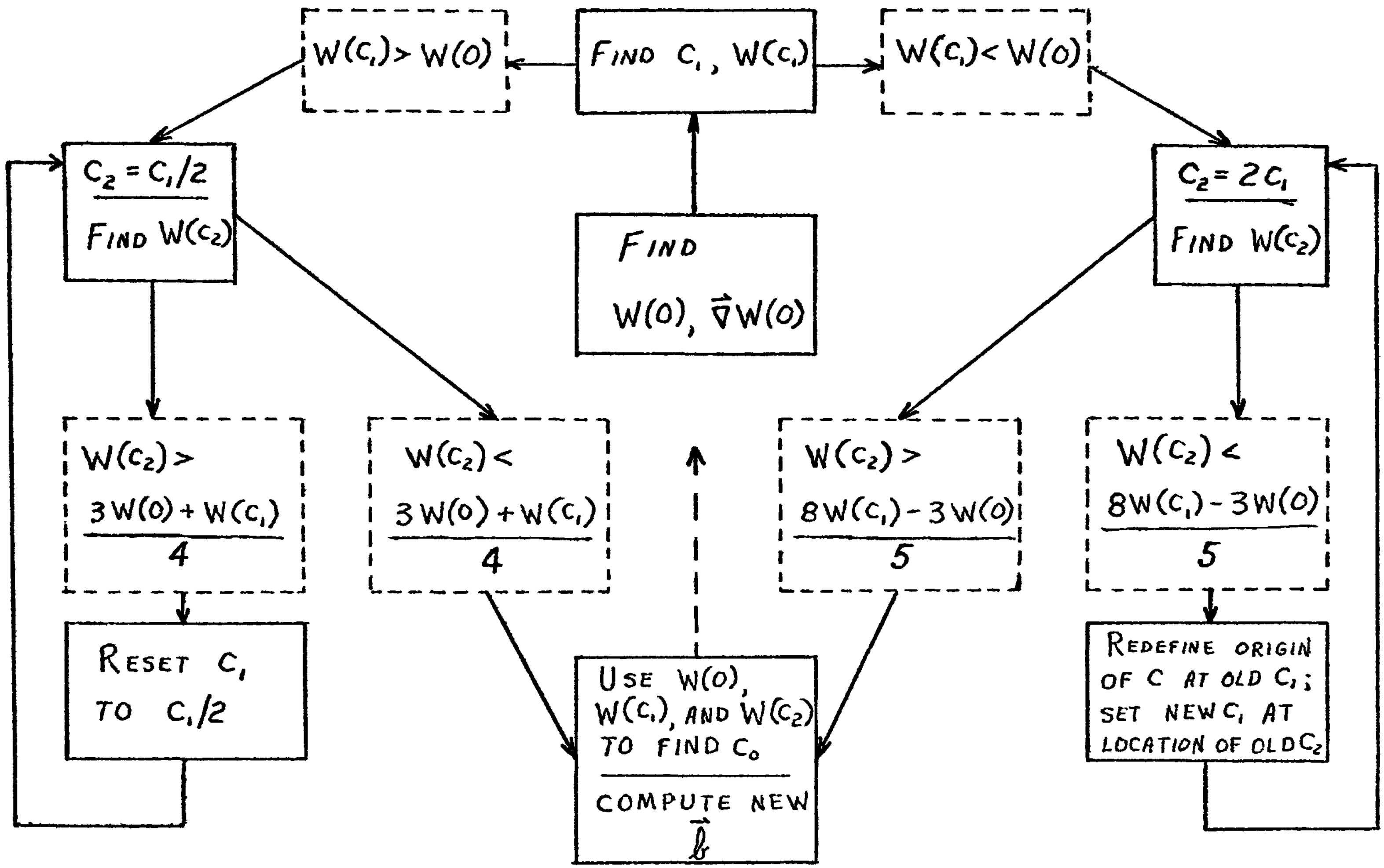
Once $W(c_1)$ has been computed, c_2 can be chosen. Sometimes, when $W(c_2)$ is calculated a peculiarity in the dependence of W on c becomes apparent which makes it necessary to discard either c_1 or c_2 and substitute another value in order to approximate c_0 accurately. The decision structure of this series of alternatives is block-diagrammed in Figure 3.

Notice that at every step the argument values are such that either $c_1 = 2c_2$ or $c_2 = 2c_1$. Having selected appropriate values of c_1 and c_2, and assuming that c_1 is the smaller, we can finally compute c_0 from the formula,

$$c_0 = c_1 + \frac{[W(0) - W(c_2)]\, c_1}{2[W(0) + W(c_2) - 2W(c_1)]} = c_1 \frac{3W(0) - 4W(c_1) + W(c_2)}{2[W(0) + W(c_2) - 2W(c_1)]} \tag{22}$$

Figure 3

Procedure for Finding c_1, c_2, and c_0



The method described in the foregoing paragraphs is designed to handle the problem of matching amplitude observations; however, a slight modification enables the same procedure to cope with amplitude-phase observations. In this case we simply consider the real and imaginary parts of the observed visibility as "amplitudes" to be matched separately. We do this by trying to minimize, instead of W, the function

$$U(\vec{b}) = \sum_{q=1}^{Q} \left\{ \left(Re[\mathcal{V}_q'(\vec{b})] - Re[\mathcal{V}_q] \right)^2 + \left(Im[\mathcal{V}_q'(\vec{b})] - Im[\mathcal{V}_q] \right)^2 \right\} \tag{23}$$

The gradient of U is easily calculated, since

$$\frac{\partial U}{\partial b_k} = 2 \sum_{q=1}^{Q} \left(Re[\mathcal{V}_q'(\vec{b})] - Re[\mathcal{V}_q] \right) \cos \theta_{kq} + 2 \sum_{q=1}^{Q} \left(Im[\mathcal{V}_q'(\vec{b})] - Im[\mathcal{V}_q] \right) \sin \theta_{kq} \tag{24}$$

The rest of the program is unchanged.

The only difference between MODEX3 and MODEX4 is that in MODEX3 the model points are read in individually, whereas in MODEX4 they are read in in rectangular blocks. (For more information see the section on input requirements.) The following summary outlines both programs.

MODEX3/4

Program Summary

ISN MODEX3	ISN MODEX4	
8	8	Define the function used to estimate the position of the minimum of U or W along the gradient line.
9-19	9-18	Read in the first part of the input according to the specified format; convert degrees to radians; compute the real and imaginary parts of the amplitudes if necessary.
--	19-37	Read in the blocks of model points, assigning coordinates to each point.
20-24	38-42	Set the model points to a uniform initial value, then read in the initial model if there is one.
25-30	43-48	Find u and v for each observation point.
31-37	49-55	Write the source name and run number, set the iteration number to zero, and set negative amplitudes to zero if required.
38-41	56-59	Write the results of the previous iteration; increment the iteration number, and stop if the maximum allowable number of iterations has been made.

42-50	60-68	Call the subroutine; compute the rms deviation of the visibilities computed from the model and print it; if the specified tolerance has been achieved, stop.
51-89	69-107	Follow the procedure diagrammed in Figure 3 for finding the position of the minimum of W or U along the gradient line.
90-102	108-120	Reconvert the phases to degrees and write out the rest of the output.

Subroutine

The function of the subroutine in both programs is to compute the function W or U, depending on whether amplitudes only or both amplitudes and phases are being fitted, and the appropriate gradient. In MODEX4 the procedure is short-cutted by means of sum and difference formulas for the sines and cosines of all the phases.

Miscellaneous Comments and Advice

1. Choosing a good set of model points is ninety percent of the job, whichever program you use. If the points selected don't cover the actual source, or at least the strongest part of it, the result can only be bad. MODEX1 is practically useless for practical work, partly because the problem of choosing the model points is even more ticklish for it than for the other programs (and partly because the results are too sensitive to errors in the input data). Therefore it should be used only when you have a very good idea of where the source is and very good data.

2. MODEX2 seems to be slightly more efficient than MODEX3/4 in the sense that fewer iterations are required in MODEX2 than in MODEX3/4 to effect the same amount of improvement in the model. Unfortunately, though, MODEX2 is quite inefficient in its use of time. In a problem using a hundred model points it takes about 23 minutes to solve the 100 equations for the hundred increments in the amplitudes. Using MODEX3/4 with phase information, on the other hand, the machine can complete a single iteration on a problem with 550 model points and 650 observations points in something less than three minutes. Without phases the iteration takes a little longer, but even then MODEX3/4 is vastly superior to MODEX2.

3. It is almost always desirable to use MODEX4 (with model points read in in blocks) than MODEX3, because this method of handling the model points makes possible a tremendous saving of time in the computation of trigonometric functions. Even if many model points have to be added to fill out the blocks the change is worthwhile. The best configuration is a single large rectangle.

4. The number of model points is limited in MODEX1/2 by the size of arrays that can be accommodated conveniently in core, but this problem does not arise in MODEX3/4. If it should be necessary to use more than 1100 observation points or 700 model points in MODEX3/4, you need only change the dimension and common statements at the beginning of the main program and the subroutine.

5. It is possible to run the program with fewer observation points than model points, but you had better not trust the results. I have had good luck using as few as 1.2 times as many observation points as model points, but I would recommend using twice as many.

6. I have observed that MODEX3/4 works a little too hard to match the phases, at the expense of the amplitudes, which no doubt are more reliable. The effect is particularly pronounced in regions where the amplitude is relatively large, as you can see with a little thought. If you have the same problem it might be worthwhile after the phases are satisfactory to make a few iterations with amplitudes alone.

7. If you run a source deck you can save a lot of time by directing the compiler to optimize the object program.

Program Listings

```
      EQUIVALENCE (IBASE(1),A(1)), (HA(1),A(501)), (A(1001), VV(1)),
     1(A(1501),PH(1))
      DATA B1, B2, BH /4067.67, 5073., 6088.33, 7104.25, 8107.7, 9117.7,
     14*0., 9992.05, 12493.2, 14993.6, 17491.01, 19993., 22495.7, 4*0.,
     2-1.87669, -1.87763, -1.87797, -1.87757, -1.87822, -1.87857, 4*0./
     3,B /140*0./, RMSA, RMSP /2*0./
      READ(5,90) (SRCNM(I), I=1,2), DEC, IDNO, NOP, NMP
      DO 1  I=1,NOP
      READ(5,91) IBASE(I), HA(I), AMP(I), PHASE(I)
    1 PHASE(I) = PHASE(I)/57.29578
      READ(5,92) (X(I),Y(I),I=1,NMP)
      SIND = SIN(DEC)
      COSD = COS(DEC)
      DO 3  I=1,NOP
      IB = IBASE(I)
      U(I) = 3.046174E-5*B2(IB)*SIN(HA(I)-BH(IB))
    3 V(I) = 3.046174E-5*(B1(IB)*COSD - B2(IB)*SIND*COS(HA(I)-BH(IB)))
      DO 6  L=1,NMP
      DO 6 K=L,NMP
      A(L,K) = 0.
      DO 6  M=1,NOP
      ANG1 = PHASE(M) - U(M)*X(L)-V(M)*Y(L)
      ANG2 = PHASE(M) - U(M)*X(K) - V(M)*Y(K)
      A(L,K) = A(L,K) + COS(ANG1)*COS(ANG2)
    6 IF(K.EQ.NMP) B(L) = B(L) + AMP(M)*COS(ANG1)
      DO 5  L=1,NMP
      DO 5  K=1,L
    5  A(L,K) = A(K,L)
      CALL ARRAY(2,NMP,NMP,140,140,A,A)
      CALL SIMQ(A,B,NMP,INOPE)
      WRITE (6,93) (SRCNM(I),I=1,2), IDNO
      IF (INOPE.EQ.1) GO TO 7
      DO 9   M=1, NOP
      VC = 0.
      VS = 0.
      DO 8 K=1,NMP
      Q = U(M)*X(K) + V(M)*Y(K)
      VC = VC + B(K)*COS(Q)
    8 VS = VS + B(K)*SIN(Q)
      VV(M) = SQRT(VC*VC + VS*VS)
      PHASE(M) = 57.29578*PHASE(M)
    9 PH(M) = 57.29578*ATAN2(VS,VC)
      WRITE (6,94) (I,X(I),Y(I),B(I), I=1,NMP)
      WRITE(6,97) (I,AMP(I),VV(I),PHASE(I),PH(I), I=1,NOP)
      DO 10  I=1,NOP
      RMSA = RMSA + (AMP(I) - VV(I))**2
      RMSPP = PHASE(I) - PH(I)
      IF(ABS(RMSPP).GT.3.141593) RMSPP = 6.283185 - ABS(RMSPP)
   10 RMSP = RMSP + RMSPP*RMSPP
      RMSA = SQRT(RMSA/NOP)
      RMSP = SQRT(RMSP/NOP)
      WRITE (6,95) RMSA, RMSP




      GO TO 11
    7 WRITE (6,96)
   11 RETURN
   90 FORMAT(4X,2A4,2X,S14.1,I5,4X,I3,4X,I3)
   91 FORMAT(10X,I2,5X,R14.1,6X,G10.5,G12.4)
```

FOLD OUT

Box GBI42 of ________

Item # Moder 2A

Between Pages Moder 1 & Moder 2B

FOLD OUT

Box GBI 42 of ________

Item # Moder 2B

Between Pages Moder 2A & 2C

FOLD OUT

Box G13I 42 of ________

Item # 77

Between Pages Model 2B & - 3

FOLD OUT

Box GBI4J **of** ________

Item # 79

Between Pages Modexo2C **&** -4|

FOLD OUT

Box GBI42 **of** ________

Item # 81

Between Pages Modex 3 **&** End