**New Folder Name** Sensitivity of Detectors

# SENSITIVITY OF A PAIR OF LASER INTERFEROMETRIC GRAVITATIONAL WAVE DETECTORS OPERATING IN COINCIDENCE AS A FUNCTION OF THEIR ORIENTATIONS

**WRITTEN BY: Yekta Gürsel and Massimo Tinto August 27, 1990**

## I. INTRODUCTION

The likely sources to be detected by the initial LIGO interferometers are the burst sources. In this memorandum, we will consider burst signals detected in coincidence by a network of two detectors which are separated by a large distance. The responses of two interferometers widely separated on the earth to a given burst signal are different due to the dependence of the detector antenna patterns on the relative orientations of the detectors. This prevents one from directly cross-correlating the two detectors outputs with each other for arbitrarily polarized bursts. The reason for this is that the source location is a free parameter constrained to a circular band in the sky by the time delay between the two detectors. If the burst is linearly or perfectly circularly polarized, such a cross correlation becomes feasible as an indicator of the presence of the signal in the data. A similar calculation may be performed in the search for the stochastic gravitational wave background, but the wide separation of the detectors reduces the sensitivity in this case. The complete determination of waves' parameters requires at least three widely separated detectors.[1]

The antenna patterns of laser interferometric gravitational wave detectors are more sensitive to the locations and the orientations of the detectors for linearly polarized bursts. For this reason, we will develop a measure of coincidence sensitivity for a pair of widely separated detectors in the case of linearly polarized bursts.

In section II, we give the general theory for the responses of the laser interferometric gravitational wave detectors as a function of their locations and orientations. In section III, we introduce the concept of coincidence probability. In section IV, we develop the detection efficiency of two detectors, and we define a coincidence sensitivity. In section V, we numerically calculate the detection efficiency for several pairs of detector locations. In section VI, we compute the coincidence sensitivity and we present a user-friendly program for general use.

## II. THE THEORY

The response of a gravitational wave detector to a plane wave, the wavelength of which is much larger than the size of the detector, is given in the literature.[2,3,4,5] Here we briefly summarize the derivation of that response using the notation and the language of Dhurandhar and Tinto,[3] and we refer the reader to them for a more complete discussion. In what follows we will use geometrical units in which the speed of light $c$ and the gravitational constant $G$ are set to 1: $c = G = 1$.

Consider a plane gravitational wave with amplitudes $h_+(t)$ and $h_\times(t)$ associated with the two independent polarizations and with a direction of propagation $\vec{n}$ incident on a detector. Let the wave coordinate system be $(X, Y, Z)$ with the wave traveling in the $Z$ direction and let the axes with respect to which $h_+(t)$ and $h_\times(t)$ are referred be the $(X, Y)$ axes. The detector coordinate system $(x', y', z')$ is obtained from them by a rotation described by the Euler angles $(\theta', \phi', \psi')$ [Fig. 1]. In the transverse-traceless (TT) gauge and in the wave coordinate system, the tensor $h_{ij}$ has the non-vanishing components:

$$h_{XX} = -h_{YY} = h_+(t) \quad , \quad h_{XY} = h_{YX} = h_\times(t) . \tag{2.1}$$

In the detector coordinate system $h_{ij}$ are complicated functions of the Euler angles.

Let us now consider the null vector $\vec{m}$ defined by :

$$\vec{m} = \frac{1}{\sqrt{2}}(\vec{e}_X + i\,\vec{e}_Y), \tag{2.2}$$

where $\vec{e}_X$ and $\vec{e}_Y$ are the unit vectors in the $(X, Y)$ directions respectively. The tensor $h_{ij}$ is then just $2\,h_+(t)\,\mathrm{Re}(m_i m_j) + 2\,h_\times(t)\,\mathrm{Im}(m_i m_j)$. We define the Symmetric-Trace-Free (STF) wave tensor $W_{ij}(t)$ to be half of the tensor $h_{ij}$ :

$$W_{ij}(t) = h_+(t)\,\mathrm{Re}(m_i m_j) + h_\times(t)\,\mathrm{Im}(m_i m_j) . \tag{2.3}$$

The factor one half is inherited from the geodesic deviation equation. In the detector coordinate system the components of $\overleftrightarrow{W}(t)$ can be obtained by finding the components of $\vec{m}$ in this system. The vector $\vec{m}$ can be written as:

$$\vec{m} = \frac{1}{\sqrt{2}}[(\cos\phi' - i\cos\theta' \sin\phi')\,\vec{e}_{x'} + (\sin\phi' + i\cos\theta' \cos\phi')\,\vec{e}_{y'} + (i\sin\theta')\,\vec{e}_{z'}]\,e^{-i\psi'}, \tag{2.4}$$

and we note that the vector $\vec{m}$ is the same as the one that appears in the Newman-Penrose formalism.[6] The vector $\vec{n}$ in the detector coordinate system assumes the following form:

$$\vec{n} = (\sin\theta' \sin\phi')\,\vec{e}_{x'} + (-\sin\theta' \cos\phi')\,\vec{e}_{y'} + (\cos\theta')\,\vec{e}_{z'} . \tag{2.5}$$

The detector can also be represented by an STF tensor $D_{ij}$. For an interferometer with its arms in the direction of the unit vectors $\vec{l}_1$ and $\vec{l}_2$, the detector tensor $D_{ij}$ is equal to :

$$D_{ij} = l_{1i}\,l_{1j} - l_{2i}\,l_{2j} . \tag{2.6}$$

The response amplitude $\eta(t)$ of a detector is then simply determined by the scalar product between the STF tensor of the wave and the STF tensor of the detector:

$$\eta(t) = D_{ij}\,W^{ij}(t) . \tag{2.7}$$

If there is more than one detector then it is convenient to introduce a common orthonormal coordinate system $(x, y, z)$ and refer to it the detectors and the wave tensors. In what follows, we will use a coordinate system referred to the center of the Earth. The Earth coordinate system is defined by choosing the $x$-axis to lie in the direction of the line passing through the center of the Earth and the intersection of the great circle which passes through Greenwich, England and the equator. The $z$-axis is chosen to lie in the direction of the line passing through the center of the Earth and the North Pole. The $y$-axis is chosen to form a right-handed Cartesian coordinate system with the $x$- and $z$-axis.

In this coordinate system, the response of a laser interferometric gravitational wave detector takes the form:

$$\eta(t) = F_+(\theta,\phi,\psi,\alpha,\beta,\gamma)\, h_+(t) + F_\times(\theta,\phi,\psi,\alpha,\beta,\gamma)\, h_\times(t), \qquad (2.8)$$

where $\beta$, $\gamma$ are the latitude and longitude of the detector; $\alpha$ is measured in the plane tangent to the Earth at the location $(\beta,\gamma)$ and it is the angle between the bisector of the arms of the detectors and the local East-West direction; $(\theta,\phi,\psi)$ are the Euler angles relating the wave coordinate system $(X, Y, Z)$ to the Earth coordinate system $(x, y, z)$ and $h_+(t)$ , $h_\times(t)$ are the two independent wave amplitudes. The "beam-pattern functions" $F_+$ and $F_\times$ depend only on the location and orientation of the detectors $(\alpha,\beta,\gamma)$ and the Euler angles $(\theta,\phi,\psi)$; they are less than one in absolute value, except when the source direction and the polarization are precisely optimal and the angle between the two arms is precisely 90 degrees, in which case one of $F_+$ or $F_\times$ is $\pm 1$ and the other is zero. In this memorandum, we assume that the angle between the two arms of a detector is precisely 90 degrees. In general, the response is given by:

$$\eta(t) = \sin(2\Omega)\,[\,F_+(\theta,\phi,\psi,\alpha,\beta,\gamma)\, h_+(t) + F_\times(\theta,\phi,\psi,\alpha,\beta,\gamma)\, h_\times(t)\,], \qquad (2.9)$$

where $2\Omega$ is the angle between the arms of a detector.[2]

In what follows, we assume that the gravitational wave impinging on the detectors is linearly polarized, i.e. either $h_+(t)$ or $h_\times(t)$ is zero. We choose $h_\times(t)=0$ without loss of generality. The response of a detector then becomes:

$$\eta(t) = F_+(\theta,\phi,\psi,\alpha,\beta,\gamma)\, h_+(t). \qquad (2.10)$$

## III. COINCIDENCE PROBABILITY

The response of a detector given by Eq. (2.10) does not take into account the noise present in realistic detectors. In general, the response depends on the level of the noise and its statistical distribution as well as the angles $(\theta,\phi,\psi,\alpha,\beta,\gamma)$ . In practice, a certain, positive threshold $T$ will be set to distinguish the signal from the noise during a given observation period. Any signal producing a response with absolute value larger than the threshold $T$ is detected, otherwise it is considered lost.

In this memorandum, we will consider the signals detected in coincidence by a network of two detectors which may be separated from each other by a large distance. We assume that the detectors are identical in construction and performance. This implies that the threshold levels of the two detectors are equal. We will call this common threshold $T$.

The probability of coincident observations of a given event depends on the geometrical factors determined by the locations and the orientations of the two detectors relative to the source and on their common threshold $T$. We assume that the locations of the sources in the sky is uniformly distributed in $(\cos(\theta), \phi)$. We also assume the polarization angle $\psi$ is uniformly distributed.

Let $h = | h_+(t) |$ be the maximum response of a given detector. We define the relative threshold $T_{rel}$ with respect to the maximum response $h$ by:

$$T_{rel} = T / h \,. \tag{3.1}$$

This is the minimum detectable value of the relative response $\eta_{rel} = | \eta / h |$. We define the single antenna detection probability $S(T_{rel})$ by:

$$S(T_{rel}) = \frac{1}{8\pi^2} \int_{\eta_{rel} \geq T_{rel}} \sin(\theta)\, d\theta\, d\phi\, d\psi \,. \tag{3.2}$$

Note that $S(T_{rel})$ is the volume element normalized by the entire volume of the $(\theta, \phi, \psi)$ space integrated over a region which gives a relative response $\eta_{rel}$ greater than or equal to the relative threshold value $T_{rel}$. We point out that the maximum value of $S(T_{rel})$ is 1.

The coincidence probability $C(T_{rel})$ for a network consisting of two identical detectors is given by:

$$C(T_{rel}) = \frac{1}{8\pi^2} \int_{\substack{\eta_{1rel} \geq T_{rel} \\ \eta_{2rel} \geq T_{rel}}} \sin(\theta)\, d\theta\, d\phi\, d\psi \,. \tag{3.3}$$

Note that the coincidence probability $C(T_{rel})$ depends on the orientations $\alpha_1, \alpha_2$, the latitudes $\beta_1, \beta_2$ and the longitudes $\gamma_1, \gamma_2$ of the two detectors. For brevity, we did not explicitly show these dependencies in Eq. (3.3). For a single detector, the detection probability $S(T_{rel})$ depends only on the relative threshold level $T_{rel}$.

The gravitational wave amplitude $h$ has the following general form:

$$h = \frac{f(M, \omega, \text{other source parameters})}{r} \,, \tag{3.4}$$

where $M$ is the mass of the source, $\omega$ is the frequency of the radiation and $r$ is the distance to the source. For a given type of source, the function $f$ is constant. This implies that the relative threshold $T_{rel} = T / h$ is equal to $r / R$ where $r$ is the distance to a source of given type and $R$ is the distance to a similar kind of source which produces the maximum signal amplitude $T$ at the detector. Using this, Eq. (3.3) becomes:

$$C(T_{rel}) = \frac{1}{8\pi^2} \int_{\substack{\eta_{1rel} \geq \frac{r}{R} \\ \eta_{2rel} \geq \frac{r}{R}}} \sin(\theta)\, d\theta\, d\phi\, d\psi \,. \tag{3.5}$$

## IV. DETECTION EFFICIENCY

We assume that the sources are distributed isotropically and homogeneously over the sky. Then the probability density $P(r/R)$ of having a source at a distance r is given by:

$$P(r/R) = 3\frac{r^2}{R^3}. \tag{4.1}$$

We define the detection efficiency $N(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2)$ as the average of the coincidence probability given by Eq. (3.5) over the observable volume:[7]

$$N(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2) = \int_0^1 C(r/R) \left[3\left[\frac{r}{R}\right]^2\right] d\left[\frac{r}{R}\right], \tag{4.2}$$

where $\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2$ are the orientation, the latitude and the longitude of the detectors 1 and 2 respectively.

The detection efficiency $N(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2)$ as defined by Eq. (4.2) is the fraction of the total number of events occurring within the observable volume of radius $R$ which will be detected in coincidence by the network. This definition of detection efficiency does not give a value of 1 even in the case of two identical detectors located at the same place and precisely aligned with each other. The reason for this is that the function $C(T_{rel})$ is 1 only when the relative threshold $T_{rel}$ is equal to zero. It is less than 1 for any other value of the relative threshold.

After reading an earlier version of this manuscript, Thorne[8] observed that the detection efficiency $N(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2)$ as defined by Eq. (4.2) can be expressed as:

$$N(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2) = \frac{1}{8\pi^2}\int_0^{\pi}\int_0^{2\pi}\int_0^{2\pi} \text{minimum}(|F_{1+}|^3, |F_{2+}|^3)\sin(\theta)\, d\theta\, d\phi\, d\psi. \tag{4.3}$$

Thorne[8] defines the coincidence sensitivity for bursts $H$ as the inverse cube root of our detection efficiency:

$$H = N^{-1/3} \tag{4.4}$$

This $H$ is directly proportional to the average amplitude noise of the two detector network. As $H$ decreases, the sensitivity of the system to the gravitational wave bursts increases.

## V. THE COMPUTATION OF DETECTION EFFICIENCY

We computed the detection efficiency given by Eq. (4.2) in two steps. First, the function $C(T_{rel})$ was computed by a Monte Carlo technique. We used 100000 uniformly generated sets of source positions and polarizations for a given relative threshold. We computed the ratio of the number of points which give a relative response above the relative threshold to the total number of sets generated as the value of the function $C(T_{rel})$ at that particular threshold level. The range of relative thresholds was covered with 100 equally spaced points which gave a relative threshold spacing of 0.01. Second, the integration over the observable volume in Eq. (4.2) was performed using the trapezoidal rule.

In Fig. 2 we plot the detection efficiency of two identical detectors located at the same place as a function of their relative orientation. In this case, the detection efficiency attains a maximum when the detectors are precisely aligned with each other as expected. The efficiency is minimum when the detectors' orientations differ by 45 degrees. Note that the derivative of the efficiency curve seems to be discontinuous at the point of precise alignment. In order to illustrate this effect more clearly, we made a coordinate transformation which located the point of precise alignment at 45 degrees in the new coordinate system. Fig. 3 shows the region around the point of precise alignment at high resolution. The total number of random sets generated was 1000000 in this case in order to improve the angular resolution.

In Fig. 4 we plot the detection efficiency of two detectors located in California and Maine. The detector in California was held fixed while the detector in Maine was rotated by 90 degrees in the plane tangent to the Earth at that location. The detector in California was located at the latitude of 36.0 degrees and the longitude of -115.0 degrees. Its orientation angle was 12.0 degrees. The detector in Maine was located at the latitude of 45.0 degrees and the longitude of -67.5 degrees.

In Fig. 5, we plot the detection efficiency of two detectors located in Germany and Maine. The detector in Germany was held fixed while the detector in Maine was rotated by 90 degrees in the plane tangent to the Earth at that location. The detector in Germany was located at the latitude of 48.0 degrees and the longitude of 11.5 degrees. Its orientation angle was 0.0 degrees. The detector in Maine was located at the latitude of 45.0 degrees and the longitude of -67.5 degrees.

In Fig. 6, we plot the detection efficiency of two detectors located in California and Germany. The detector in California was held fixed while the detector in Germany was rotated by 90 degrees in the plane tangent to the Earth at that location. The detector in California was located at the latitude of 36.0 degrees and the longitude of -115.0 degrees. Its orientation angle was 12.0 degrees. The detector in Germany was located at the latitude of 48.0 degrees and the longitude of 11.5 degrees.

In Fig. 7, we plot contours of the detection efficiency as a function of the orientations of the detectors located in California and Maine. The detector in California was located at the latitude of 36.0 degrees and the longitude of -115.0 degrees. The detector in Maine was located at the latitude of 45.0 degrees and the longitude of -67.5 degrees. Note that the contours of constant efficiency are approximately lines with slope of 1. This means that if one detector is rotated from the optimal orientation then the other one has to be rotated by the same amount in order to retain the optimality. We point out that the optimal orientations determined using the contours of the detection efficiency agrees very closely with the values given by Schutz and Tinto.[2] In Fig. 8 we show the 3-dimensional representation of the contour plot shown in Fig. 7.

A similar calculation has also been performed by Robert Spero.[9] He plots $T_{rel}$ at constant $C(T_{rel})=0.20$ as a function of the orientation of one of the detectors in Fig. 7 of his document. We reproduce his plots using our data as shown in Fig. 9 of this document. The results are in very good agreement. Note that this only means that our raw data were in agreement, the definitions of sensitivity are quite different. Spero[9] has compared our definition to his. He concludes that they agree within 5 percent.

Weiss[10] has performed a calculation based on the cross-correlated responses of two widely separated detectors to a stochastic background of gravitational waves. His results do not agree with ours to a good degree.

## VI. THE COMPUTATION OF THE COINCIDENCE SENSITIVITY

We also wrote a program to compute the coincidence sensitivity $H$ defined by Eq. (4.4). We perform the average over the angles $\theta, \phi, \psi$ on a uniform grid consisting of $47 \times 47 \times 47 = 103283$ points. The antenna patterns $F_{1+}$ and $F_{2+}$ are computed according to the formalism given in section II. The accuracy of the program is about 5 percent with the given grid size.

The program requires the latitudes and the longitudes of the detectors as well as the orientation of one of the detectors as its inputs. The other detector's orientation is then stepped through a range with a given step size. The range and the step size are determined by the user. The program computes the detection efficiency as a function of the orientation of the second detector and outputs the results in the form of print-outs and plots.

The program is located in the directory "~yekta/fast-orient" and is called "fast-orient". It is self-explanatory. The output file consists of two columns of numbers. The first column lists the orientations of the rotating detector in degrees and the second column gives the corresponding coincidence sensitivities of the system.

The computation of coincidence sensitivity for a given pair of orientations of two detectors takes about 6.5 seconds on a Sun SPARCstation 1+. The total time it takes to step through a full 90 degree range of orientations with a step size of 1 degree is about 10 minutes.

In Fig. 10 we give an example plot of the coincidence sensitivity for the detector pair located in California and Maine. The detector in California is taken as the reference detector, and the Maine detector is rotated by 90 degrees in steps of 1 degree.

As an appendix, we also include the FORTRAN 77 listing of our program "fast-orient". The code is not very transparent to the untrained eye as it is written for maximum speed.

## VI. REFERENCES

[1] Y. Gürsel and M. Tinto, Phys. Rev. D, 40, 3884 (1989).

[2] B. F. Schutz and M. Tinto, Mon. Not. R. Astr. Soc., 224, 131 (1987).

[3] S. V. Dhurandhar and M. Tinto, Mon. Not. R. Astr. Soc., 234, 663 (1988).

[4]F. B. Estabrook, Gen. Rel. Grav., 17, 719 (1985).

[5]R. L. Forward, Phys. Rev. D, 17, 379 (1978).

[6]E. T. Newman and R. Penrose, J. Math. Phys., 3, 566 (1962)

[7]M. Tinto, in *Proceedings of the NATO Workshop on Gravitational Wave Data Analysis*, edited by B. F. Schutz, (D.Reidel, Dordrecht, 1988).

[8]K. S. Thorne, private communication.

[9]R. Spero, Draft Report *Sensitivity of Two Way Coincidence Measurements, as Related to Site Selection*, 6th Draft, August 1, 1990.

[10]R. Weiss, Draft Report *Simple Formulation of Coincidence Sensitivity of a Pair of Interferometers*, July 9, 1990.

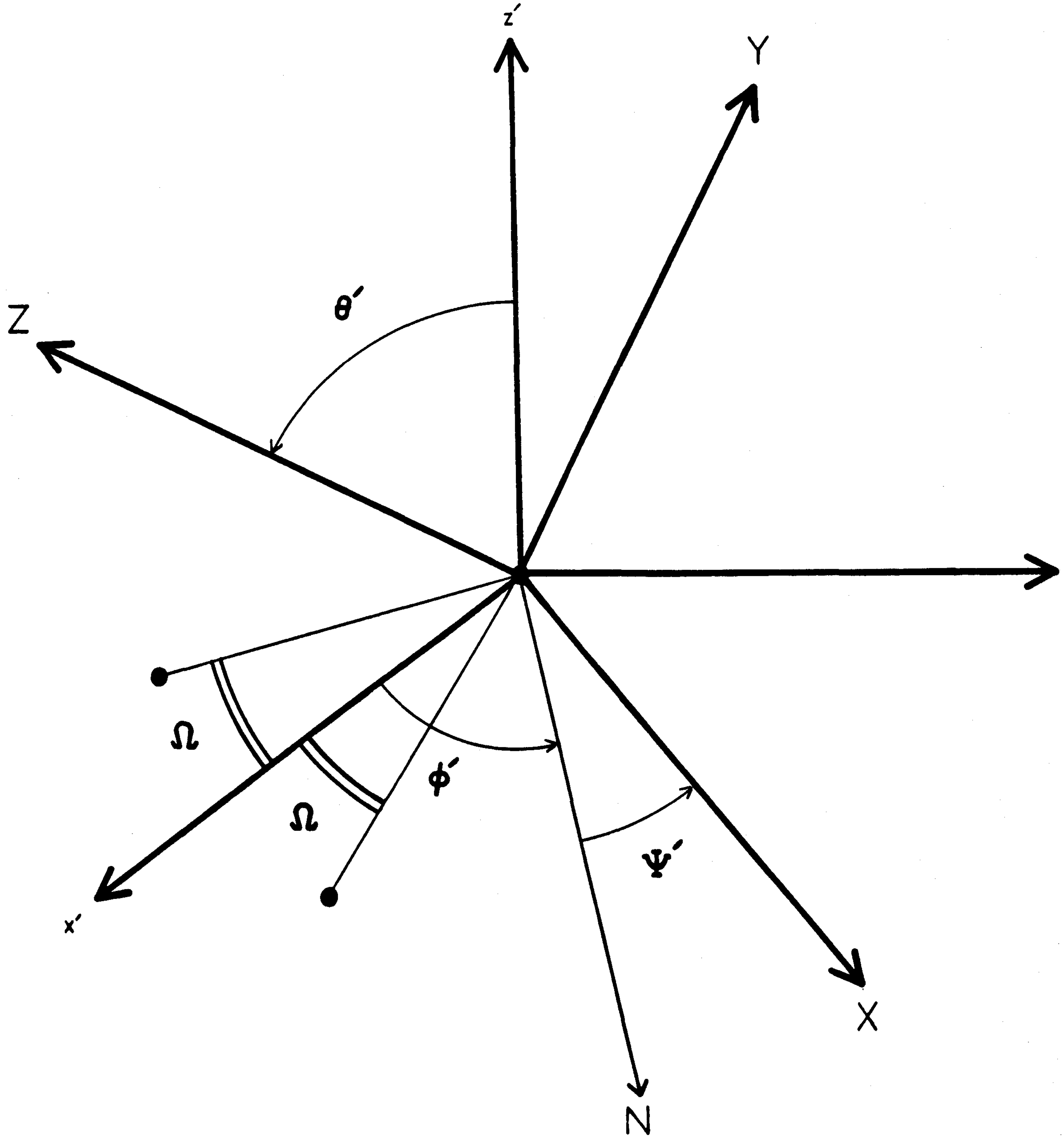



Fig. 1

TWO IDENTICAL DETECTORS AT THE SAME LOCATION

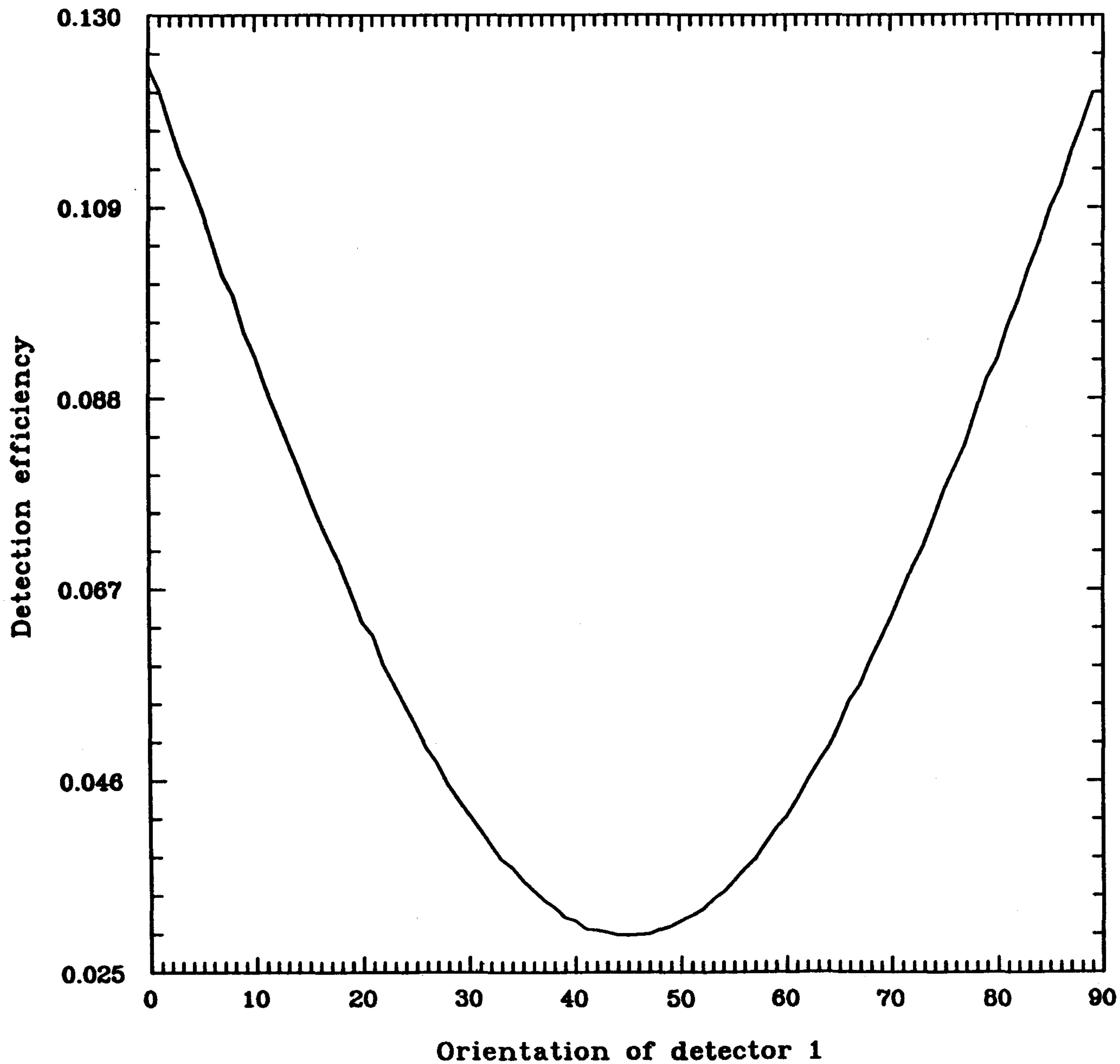



Fig. 2

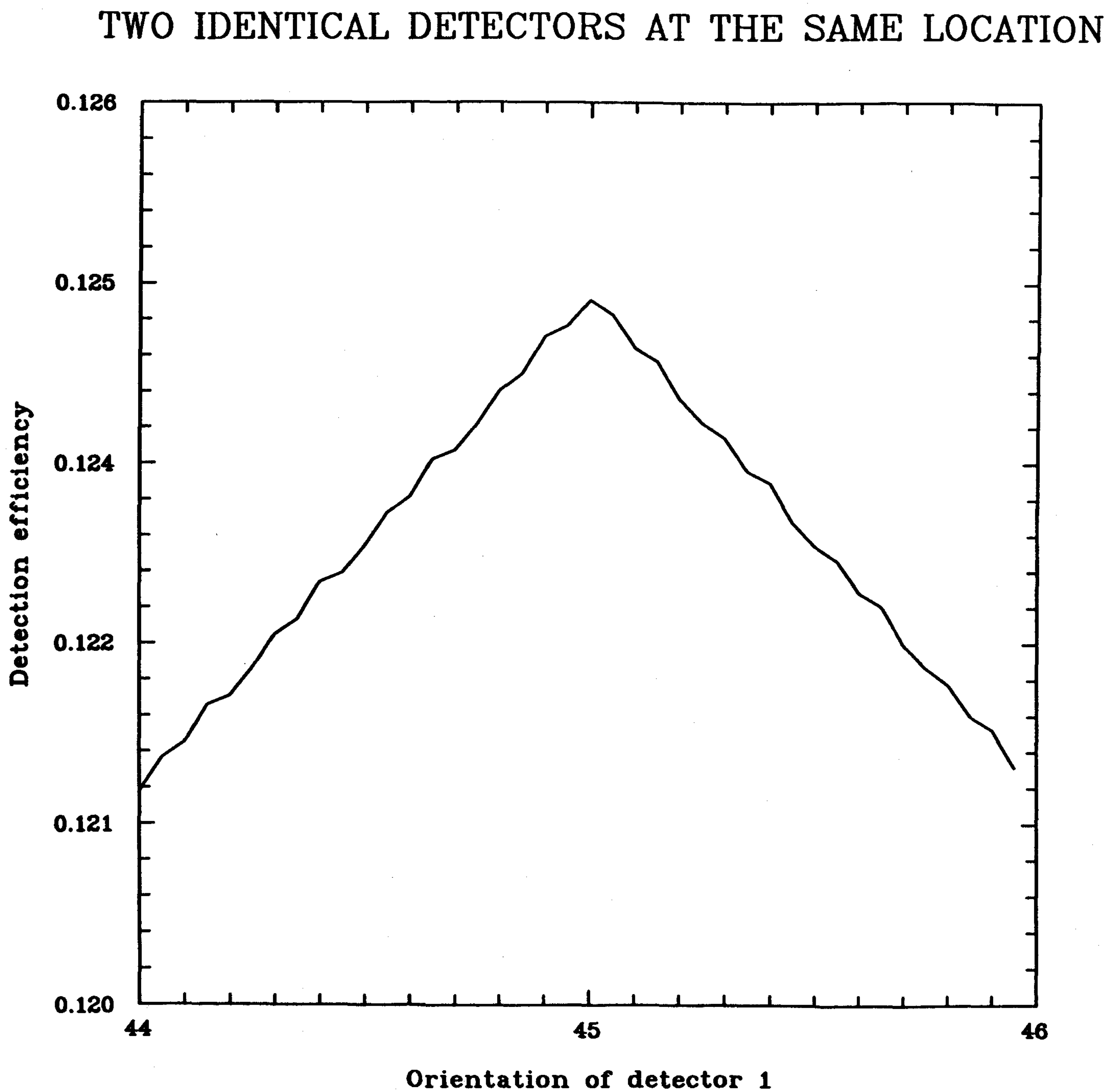
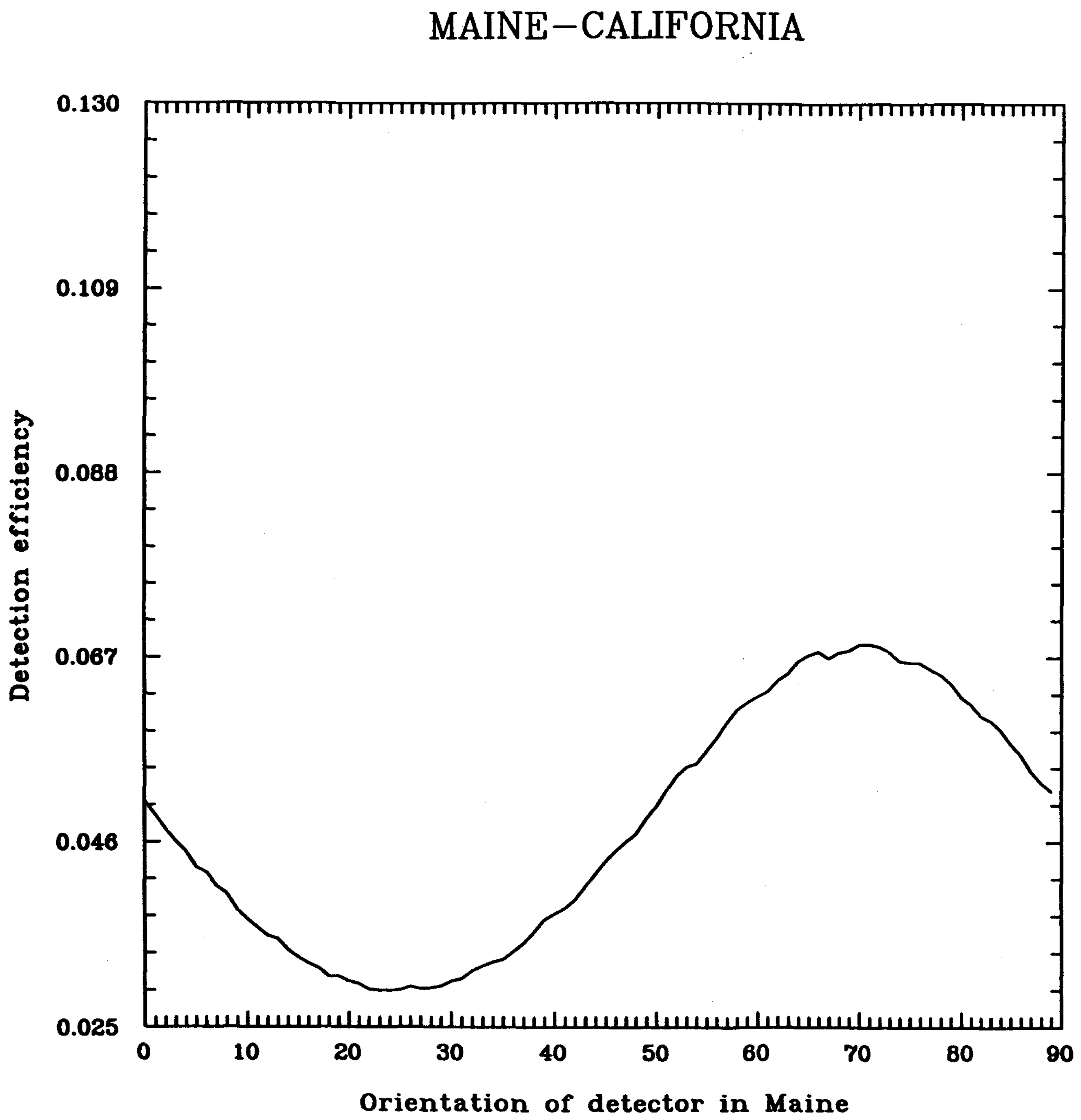

TWO IDENTICAL DETECTORS AT THE SAME LOCATION
0.126
0.125
0.124
0.122
0.121
0.120
Detection efficiency
44
45
46
Orientation of detector 1

Fig. 3

(3)



MAINE–CALIFORNIA
0.130
0.109
0.088
0.067
0.046
0.025
Detection efficiency
0
10
20
30
40
50
60
70
80
90
Orientation of detector in Maine


Fig. 4

(4)

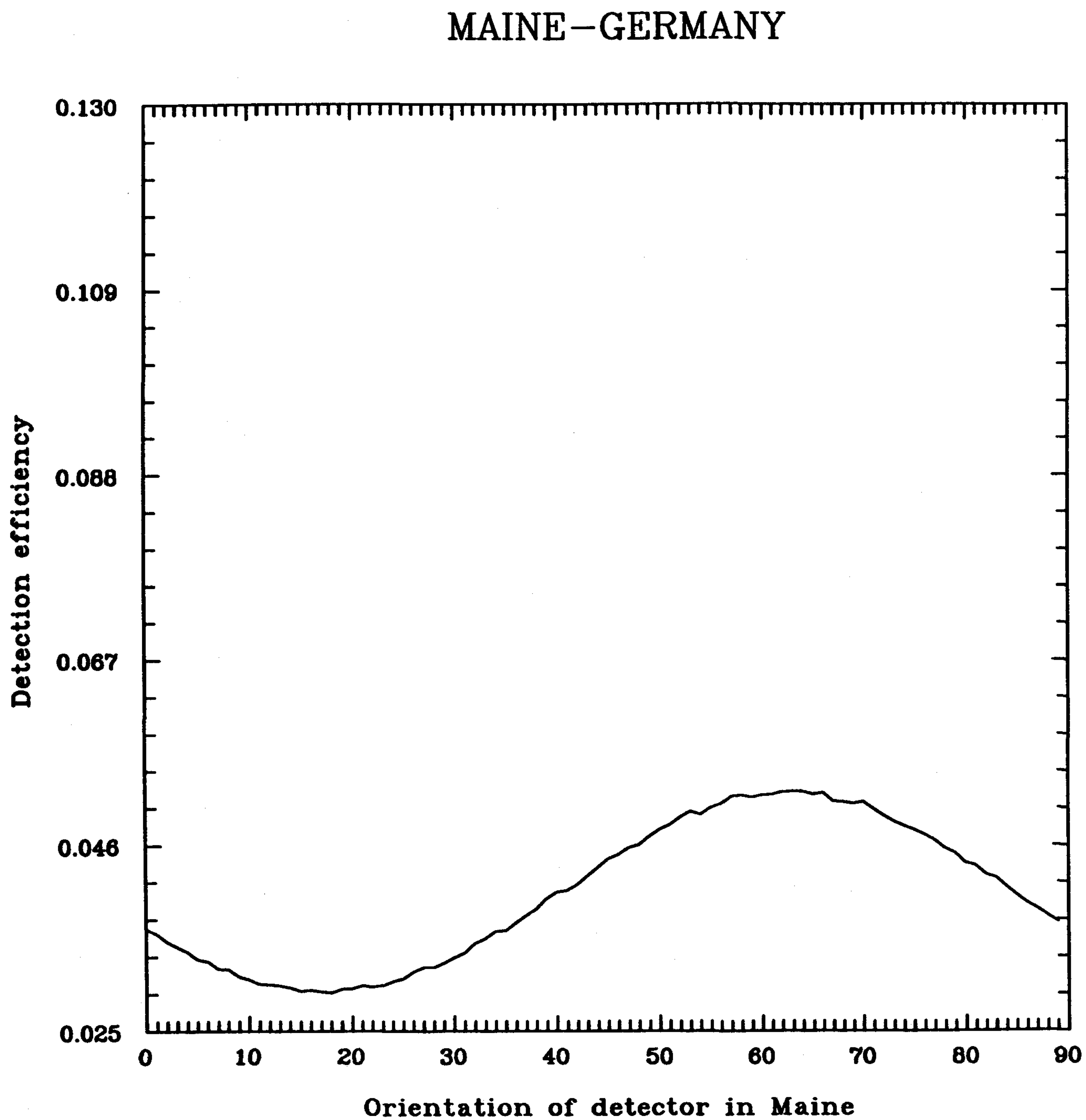

MAINE−GERMANY
0.130
0.109
0.088
0.067
0.046
0.025
Detection efficiency
0
10
20
30
40
50
60
70
80
90
Orientation of detector in Maine

Fig. 5

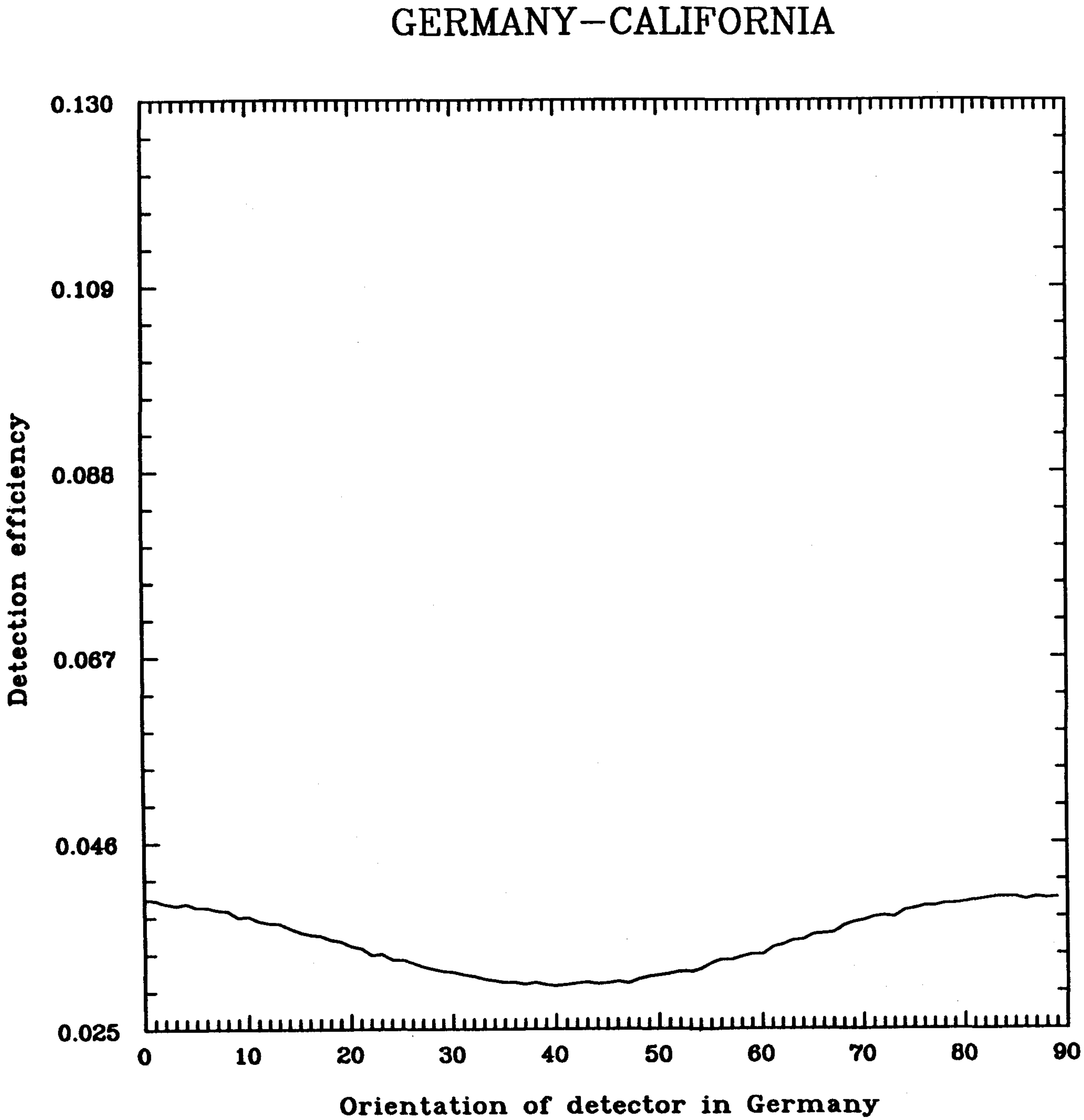

GERMANY–CALIFORNIA
0.130
0.109
0.088
0.067
0.046
0.025
Detection efficiency
0
10
20
30
40
50
60
70
80
90
Orientation of detector in Germany

Fig. 6

DETECTION EFFICIENCY ( $\times 10^4$)

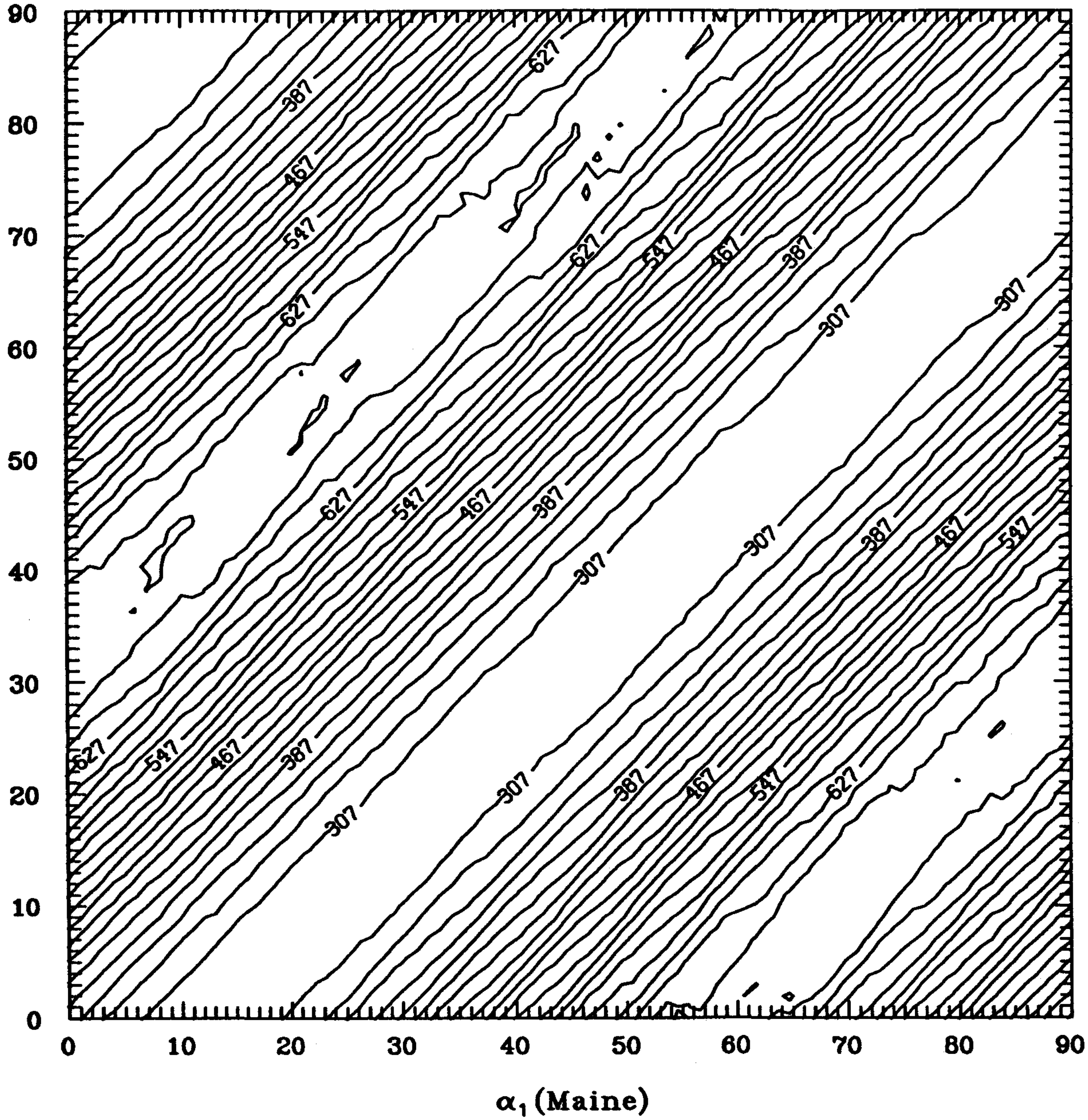



Fig. 7.

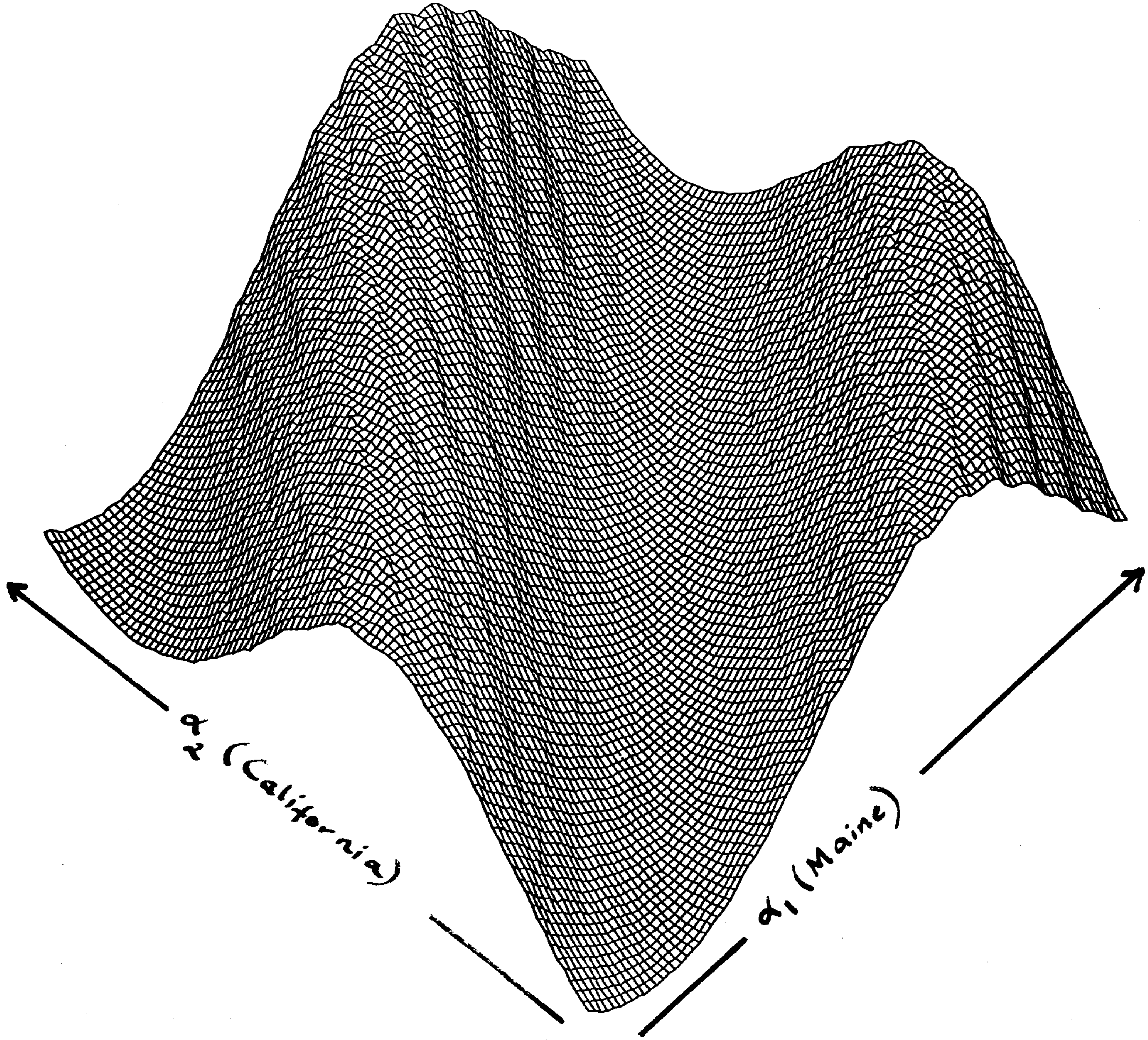



Fig. 8

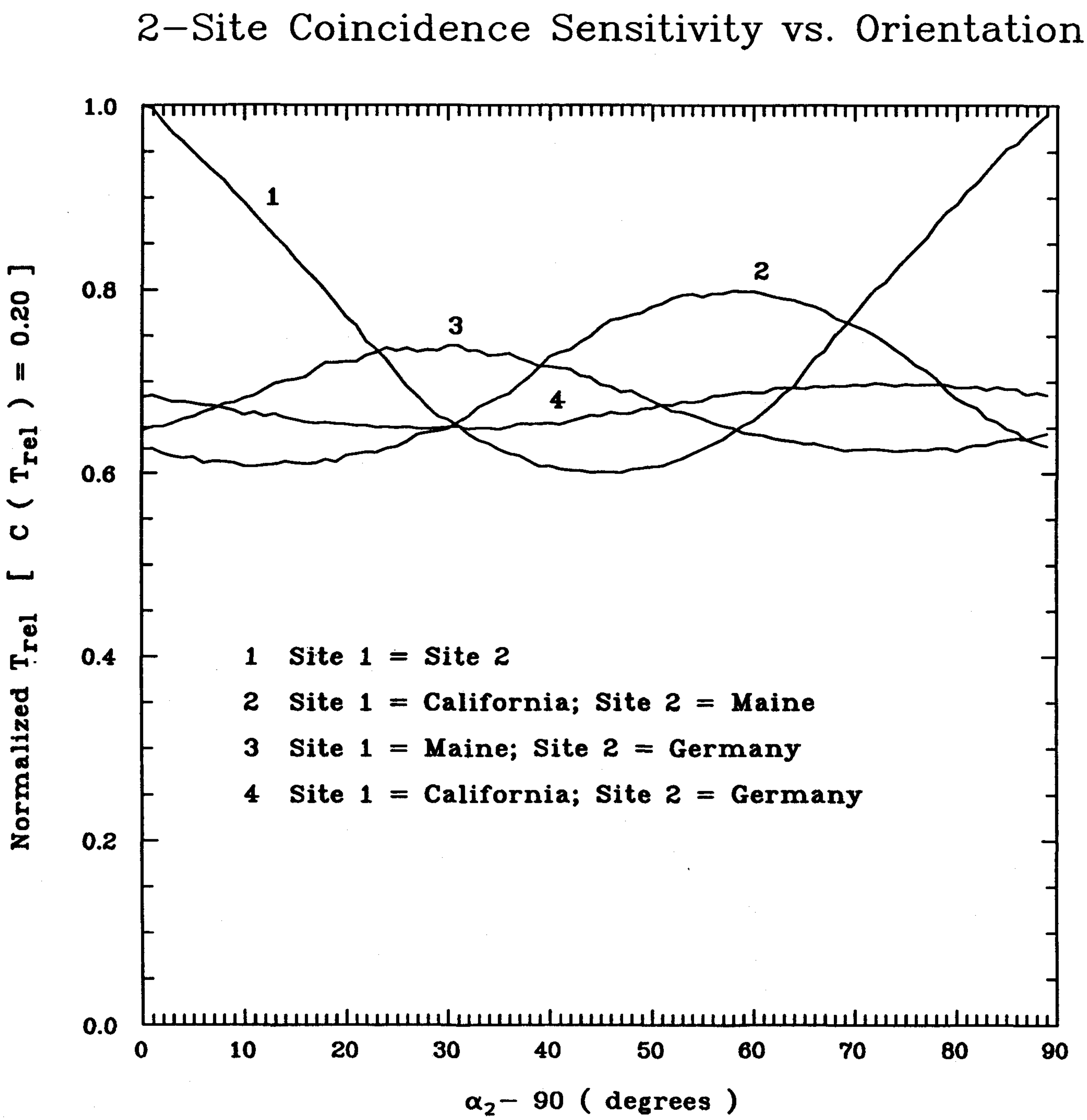
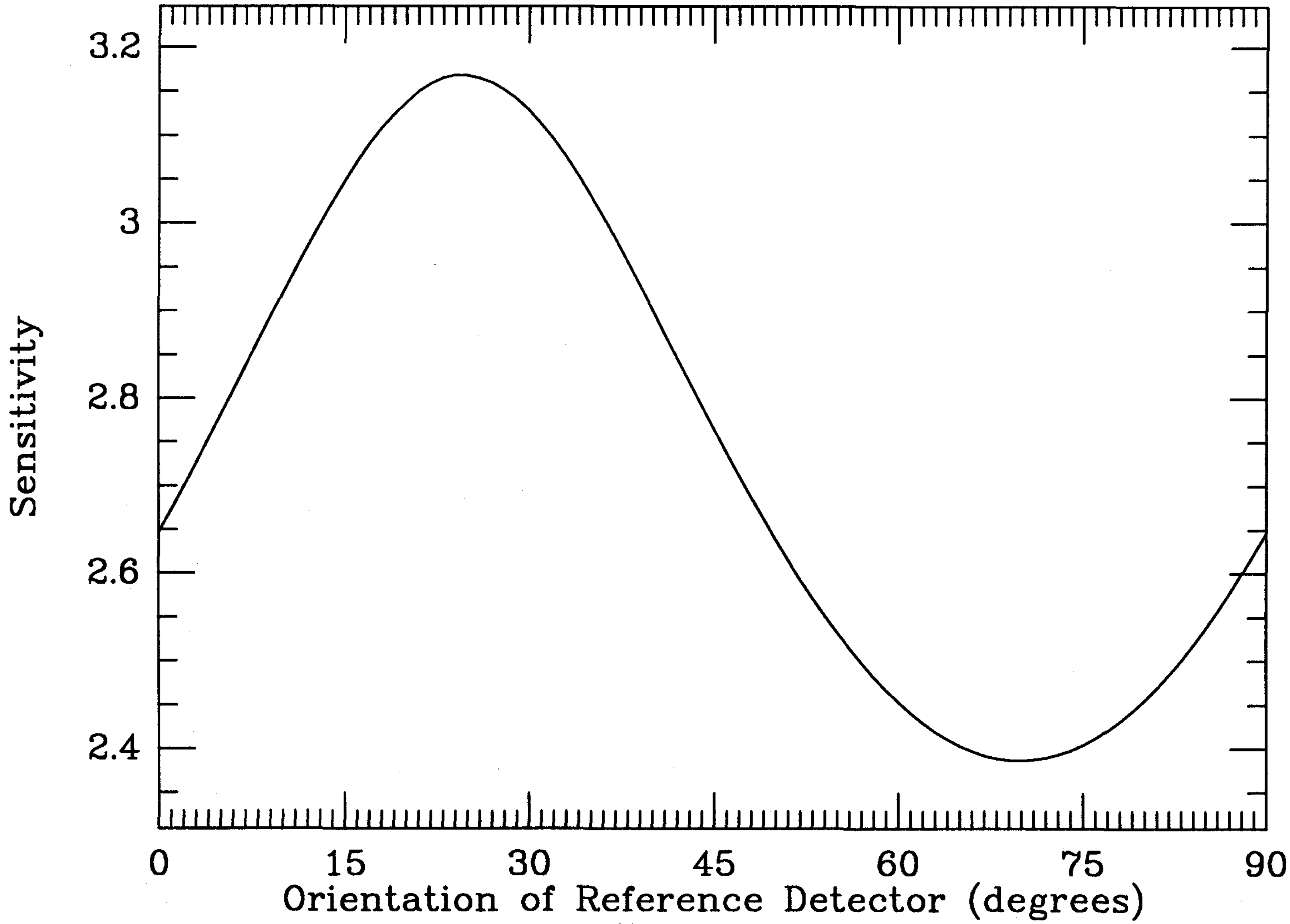

2–Site Coincidence Sensitivity vs. Orientation
1.0
0.8
0.6
0.4
0.2
0.0
Normalized $T_{rel}$ [ C ( $T_{rel}$ ) = 0.20 ]
1
2
3
4
1 Site 1 = Site 2
2 Site 1 = California; Site 2 = Maine
3 Site 1 = Maine; Site 2 = Germany
4 Site 1 = California; Site 2 = Germany
0
10
20
30
40
50
60
70
80
90
$\alpha_2 - 90$ ( degrees )

Fig. 9.

# Maine–California




Reference Detector: Latitude = 36.0 Longitude = −115.0 Orientation = 12.0

Rotating Detector: Latitude = 45.0 Longitude = −67.5

Fig 10.

```
      PROGRAM FASTORIENT

C     ARRAYS:

      COMPLEX MV(3)

      REAL SD1(3),CD1(3),SD2(3),CD2(3),D1(3,3),D2(3,3),W(3,3),
     $     CTHE(200),STHE(200),SPHI(200),CPHI(200),SPSI(200),CPSI(200)

      REAL X(200), Y(200)

C     OTHER REALS:

      REAL DEGRAD,PI1,COST,PHI,PSI,DELTA_CTHE,DELTA_PHI,DELTA_PSI,
     $     NORMFAC,FP1,FP2,FPU1,FPU2,FUNC,ABSF1P,ABSF2P,RFUNC,LATR,
     $     LONGR,ORIR,LATM,LONGM,SORIM,EORIM,DELTA_CORIT,CORIT,YMIN,
     $     YMAX,MARGIN,LXMIN,LXMAX,LYMIN,LYMAX,GLABELX,GLABELY,
     $     XMINOR,XMAJOR,LINEX,LINE1Y,LINE2Y,EXPONENT,TWOOVERLAP

C     INTEGERS:

      INTEGER STEPNUM,SCRTY,PLOTLY,NVEC,STRLENGTH,GRIDNUM,
     $        XLABELL,YLABELL,GLABELL,LINE1L,LINE2L

C     "DO LOOP" INDICES:

      INTEGER I,J,K,L,M,N

C     LOGICAL VARIABLES:

      LOGICAL DELTR,ITISNOTTHERE,GETYESNO,PLTOK,CHPR,INVERSE,NORMALIZE

C     STRING VARIABLES:

      CHARACTER*256 OUTFILENAME,XLABEL,YLABEL,GLABEL,LINE1,LINE2

      CHARACTER*20, ANSWER,NUM1,NUM2,NUM3

C     MONGO VARIABLES:

      REAL USERVAR(10)

      REAL X1,X2,Y1,Y2,GX1,GX2,GY1,GY2,GX,GY,CX,CY,EXPAND,
     $     ANGLE,CHEIGHT,CWIDTH,CXDEF,CYDEF,PXDEF,PYDEF,PI

      INTEGER LX1,LX2,LY1,LY2,LWEIGHT,LTYPE,COFF,NUMDEV

      LOGICAL TERMOUT,XYSWAPPED,AUTODOT

C     COMMON BLOCKS (INTERFACE TO MONGO):

      COMMON /MONGOPAR/
     $     X1,X2,Y1,Y2,GX1,GX2,GY1,GY2,LX1,LX2,LY1,LY2,
     $     GX,GY,CX,CY,
     $     EXPAND,ANGLE,LTYPE,LWEIGHT,
     $     CHEIGHT,CWIDTH,CXDEF,CYDEF,PXDEF,PYDEF,COFF,
     $     TERMOUT,XYSWAPPED,NUMDEV,
     $     PI,USERVAR,AUTODOT

C     THE CONSTANTS:

      PI1=4.*ATAN(1.)

      DEGRAD=PI1/180.
```

```
      TWOOVERLAP=0.132051075

      PRINT*, "            Welcome to the fast-orient program."
      PRINT*, "    Yekta Gursel and  Massimo Tinto  August 24, 1990"
      PRINT*, " "
      PRINT*, "  This program computes and plots the detection efficienc
     $y"
      PRINT*, "of two laser interferometer gravitational wave detectors"
      PRINT*, "running in coincidence."
      PRINT*, " "
      PRINT*, "  The orientation of one of the detectors is fixed.  This
     $"
      PRINT*, "detector is called the reference detector.  The orientati
     $on"
      PRINT*, "of the other detector is stepped through a range of value
     $s"
      PRINT*, "and the detection efficiency is computed as a function of
     $"
      PRINT*, "the orientation of the rotating detector.  The orientatio
     $n"
      PRINT*, "is defined to be the angle between the bisector of the ar
     $ms"
      PRINT*, "and the local East-West direction."
      PRINT*, "  "
      PRINT*, "  The computation of the detection efficiency for a given
     $"
      PRINT*, "pair of orientations takes about 6.5 seconds on a Sun"
      PRINT*, "SPARCstation 1+.  The total time it takes to step through
     $"
      PRINT*,"a 90 degree range of orientations for the moving detector"
      PRINT*, "with a 1 degree step size is about 10 minutes.  The accur
     $acy"
      PRINT*, "of the computation is within 5 percent with the default
     $grid."
      PRINT*, " "

 180  CONTINUE

      PRINT*, "Ready?"

      IF(GETYESNO()) THEN

         GOTO 190

      ELSE

         STOP

      ENDIF

 190  CONTINUE

      PRINT*, "Do you want to change the integration grid size?"

      IF (GETYESNO()) THEN

 150     CONTINUE

         PRINT*, "Enter the new grid size:"
         PRINT*, "(It should be greater than 46 and less than 201."
         PRINT*, " Remember that the integration time goes as the"
         PRINT*, " cube of the grid size.)"
```

```
         CALL GETINT(GRIDNUM)

         IF ((GRIDNUM .LT. 47) .OR.
     $        (GRIDNUM .GT. 200)) THEN

            PRINT*, "Grid size is out of range."

            GOTO 150

         ENDIF

      ELSE

         GRIDNUM=47

      ENDIF

      DELTA_CTHE=2.0/(REAL(GRIDNUM-1)+0.001)
      DELTA_PHI=2.0*PI1/REAL(GRIDNUM-1)
      DELTA_PSI=2.0*PI1/REAL(GRIDNUM-1)

      COST=-1.0-DELTA_CTHE
      PHI=-DELTA_PHI
      PSI=-DELTA_PSI

      DO 200 I=1,GRIDNUM

         COST=COST+DELTA_CTHE

         CTHE(I)=COST
         STHE(I)=SQRT(1.0-COST*COST)

         PHI=PHI+DELTA_PHI

         SPHI(I)=SIN(PHI)
         CPHI(I)=COS(PHI)

         PSI=PSI+DELTA_PSI

         SPSI(I)=SIN(PSI)
         CPSI(I)=COS(PSI)

 200  CONTINUE

      PRINT*, "Enter the latitude of the reference detector in degrees:"

      CALL GETREAL(LATR)

      PRINT*, "Enter the longitude of the reference detector in degrees:
     $"

      CALL GETREAL(LONGR)

      PRINT*, "Enter the orientation of the reference detector in degree
     $s:"

      CALL GETREAL(ORIR)

      PRINT*, "Enter the latitude of the rotating detector in degrees:"

      CALL GETREAL(LATM)

      PRINT*, "Enter the longitude of the rotating detector in degrees:"
```

```
      CALL GETREAL(LONGM)

      PRINT*, "Enter the starting angle for the orientation of this dete
     $ctor:"
      PRINT*, "(in degrees)"

      CALL GETREAL(SORIM)

      PRINT*, "Enter the final angle for the orientation of this detecto
     $r:"
      PRINT*, "(in degrees)"

      CALL GETREAL(EORIM)

 170  CONTINUE

      PRINT*, "Enter the number of steps for the rotating detector:"

      CALL GETINT(STEPNUM)

      IF ((STEPNUM .LT. 1) .OR.
     $     (STEPNUM .GT. 199)) THEN

         PRINT*, "This number should be larger than 0 and less than 199.
     $"

         GOTO 170

      ENDIF

 210  CONTINUE

      PRINT*, "Do you want to compute inverse of the sensitivity?"
      PRINT*, "(H=N^(1/3) as opposed to H=N^(-1/3), H is the sensitivity
     $,"
      PRINT*, "N is the detection efficiency as defined in the memo.)"

      INVERSE=GETYESNO()

      IF (INVERSE) THEN

         EXPONENT=1./3.

      ELSE

         EXPONENT=-1./3.

      ENDIF

      PRINT*, "Do you want the results normalized to the case of two"
      PRINT*, "overlapping detectors?"

      NORMALIZE=GETYESNO()

      IF (NORMALIZE) THEN

         NORMFAC=DELTA_CTHE*DELTA_PHI*DELTA_PSI/
     $           (8.0*PI1*PI1*TWOOVERLAP)

      ELSE

         NORMFAC=DELTA_CTHE*DELTA_PHI*DELTA_PSI/(8.0*PI1*PI1)

      ENDIF
```

```
      PRINT*, "Enter the file name for the output:"

      READ (*,'(A80)') OUTFILENAME

      ITISNOTTHERE=DELTR(OUTFILENAME)

      IF(.NOT. ITISNOTTHERE) THEN

         GOTO 210

      ENDIF

      PRINT*, "Do you want the output plotted?"

      PLTOK=GETYESNO()

      IF (PLTOK) THEN

         XLABEL='Orientation of Reference Detector (degrees)'
         XLABELL=STRLENGTH(XLABEL)

         IF (INVERSE) THEN

            IF (NORMALIZE) THEN

               YLABEL='Inverse Normalized Sensitivity'

            ELSE

               YLABEL='Inverse Sensitivity'

            ENDIF

         ELSE

            IF (NORMALIZE) THEN

               YLABEL='Normalized Sensitivity'

            ELSE

               YLABEL='Sensitivity'

            ENDIF

         ENDIF

         YLABELL=STRLENGTH(YLABEL)

         PRINT*, "Enter the graph title:"

         READ (*,'(A80)') GLABEL

         GLABELL=STRLENGTH(GLABEL)

         PRINT*, "Do you want to send the plot only to the printer?"

         CHPR=GETYESNO()

         IF(.NOT. CHPR) THEN

240         CONTINUE
```

```
         PRINT*, "Enter the display type:"
         PRINT*, "(Type tek for Tektronix 4010, sun for Sun Windows."
         PRINT*, " After the plot is displayed: Pressing carriage"
         PRINT*, " return exits the program; pressing p and carriage"
         PRINT*, " return prints the displayed plot in portrait forma
     $t;"
         PRINT*, " pressing l and carriage return prints the displaye
     $d"
         PRINT*, " plot in landscape format.)"

         READ (*, '(A10)') ANSWER

         CALL UPPERCASE(ANSWER)

         IF (ANSWER .EQ. 'TEK') THEN

            SCRTY=3

         ELSEIF (ANSWER .EQ. 'SUN') THEN

            SCRTY=6

         ELSE

            PRINT*, "That display is not available."
            GOTO 240

         ENDIF

      ELSE

250      CONTINUE

         PRINT*, "Enter the plot layout:"
         PRINT*, "(Type either portrait or landscape)"

         READ (*, '(A10)') ANSWER

         CALL UPPERCASE(ANSWER)

         IF (ANSWER .EQ. 'PORTRAIT') THEN

            PLOTLY=-5

         ELSEIF (ANSWER .EQ. 'LANDSCAPE') THEN

            PLOTLY=-6

         ELSE

            PRINT*, "That layout is not available."
            GOTO 250

         ENDIF

      ENDIF

   ENDIF

   DELTA_CORIT=DEGRAD*(EORIM-SORIM)/REAL(STEPNUM)

   SD1(2)=SIN(DEGRAD*LATM)
   SD1(3)=SIN(DEGRAD*LONGM)
```

```
      CD1(2)=COS(DEGRAD*LATM)
      CD1(3)=COS(DEGRAD*LONGM)

      SD2(1)=SIN(DEGRAD*ORIR)
      SD2(2)=SIN(DEGRAD*LATR)
      SD2(3)=SIN(DEGRAD*LONGR)

      CD2(1)=COS(DEGRAD*ORIR)
      CD2(2)=COS(DEGRAD*LATR)
      CD2(3)=COS(DEGRAD*LONGR)

C     THE MAIN PROGRAM STARTS:

      OPEN(20, FILE=OUTFILENAME, STATUS='NEW')

      CALL DETTENS(SD2,CD2,D2)

      CORIT=SORIM*DEGRAD

      SD1(1)=SIN(CORIT)
      CD1(1)=COS(CORIT)

      DO 20 I=1,STEPNUM+1

         CALL DETTENS(SD1,CD1,D1)

         RFUNC=0.0

         DO 100 J=1,GRIDNUM

            DO 110 K=1,GRIDNUM

               DO 120 L=1,GRIDNUM

C     THE COMPONENTS OF THE NULL VECTOR M:

                  MV(1)=CMPLX(CPHI(J),
     $                  -CTHE(L)*SPHI(J))*CMPLX(CPSI(K),-SPSI(K))
                  MV(2)=CMPLX(SPHI(J),
     $                  CTHE(L)*CPHI(J))*CMPLX(CPSI(K),-SPSI(K))
                  MV(3)=CMPLX(STHE(L)*SPSI(K),STHE(L)*CPSI(K))

C     THE REAL PART OF THE TENSOR M x M:

                  DO 40 M=1,3
                     DO 50 N=M,3

                        W(M,N)=REAL(MV(M))*REAL(MV(N))
     $                         -IMAG(MV(M))*IMAG(MV(N))
                        W(N,M)=W(M,N)

 50                  CONTINUE
 40               CONTINUE

C     COMPUTE F+:

                  FP1=0.0
                  FP2=0.0

                  DO 60 M=1,2
                     DO 70 N=M+1,3

                        FP1=FP1+D1(M,N)*W(M,N)
                        FP2=FP2+D2(M,N)*W(M,N)
```

```
70                      CONTINUE
60                   CONTINUE

                     FPU1=D1(1,1)*W(1,1)+D1(2,2)*W(2,2)+D1(3,3)*W(3,3)
                     FPU2=D2(1,1)*W(1,1)+D2(2,2)*W(2,2)+D2(3,3)*W(3,3)

                     FP1=FP1+0.5*FPU1
                     FP2=FP2+0.5*FPU2

                     ABSF1P=ABS(FP1)
                     ABSF2P=ABS(FP2)

                     FUNC=MIN(ABSF1P,ABSF2P)

                     RFUNC=RFUNC+FUNC*FUNC*FUNC

120               CONTINUE

110            CONTINUE

100         CONTINUE

            IF(PLTOK) THEN

               X(I)=CORIT/DEGRAD

               Y(I)=(NORMFAC*RFUNC)**EXPONENT

            ENDIF

            WRITE(20,'(1X,2E15.5)') CORIT/DEGRAD,(NORMFAC*RFUNC)**EXPONENT

            CORIT=CORIT+DELTA_CORIT

            SD1(1)=SIN(CORIT)
            CD1(1)=COS(CORIT)

20       CONTINUE

         CLOSE(20)

160      CONTINUE

         IF (PLTOK) THEN

            YMAX=0.0
            YMIN=1.0E33

            DO 300 I=1,STEPNUM

               IF(Y(I) .GT. YMAX) THEN

                  YMAX=Y(I)

               ENDIF

               IF(Y(I) .LT. YMIN) THEN

                  YMIN=Y(I)

               ENDIF

300         CONTINUE
```

```
      IF (YMAX .EQ. YMIN) THEN

         PRINT*, "The maximum of the curve is equal to the minimum of
     $ the curve."
         PRINT*, "Plot can not be produced."

         STOP

      ENDIF

      MARGIN=0.1*(YMAX-YMIN)

      YMIN=YMIN-MARGIN
      YMAX=YMAX+MARGIN

      X(STEPNUM+1)=EORIM

      CALL MGOINIT

      IF(CHPR) THEN

         CALL MGOSETUP(PLOTLY)

         CALL MGOSETLWEIGHT(2)

      ELSE

         CALL MGOSETUP(SCRTY)

      ENDIF

      CALL MGOERASE

      LXMIN=REAL(LX1)+0.2*REAL(LX2-LX1)
      LYMIN=REAL(LY1)+0.2*REAL(LY2-LY1)
      LXMAX=REAL(LX2)-0.2*REAL(LX2-LX1)
      LYMAX=REAL(LY2)-0.2*REAL(LY2-LY1)

      CALL MGOSETLOC(LXMIN,LYMIN,LXMAX,LYMAX)

      CALL MGOSETLIM(SORIM,YMIN,EORIM,YMAX)

      XMAJOR=(EORIM-SORIM)/6.0
      XMINOR=XMAJOR/15.0

      CALL MGOTICKSIZE(XMINOR,XMAJOR,0.0,0.0)

      CALL MGOBOX(1,2)

      CALL MGOCONNECT(X,Y,STEPNUM+1)

      CALL MGOXLABEL(XLABELL,XLABEL)

      CALL MGOYLABEL(YLABELL,YLABEL)

      LINEX=(SORIM+EORIM)/2.0

      IF (CHPR) THEN

         LINE1Y=YMIN-2.0*MARGIN
         LINE2Y=YMIN-3.0*MARGIN

      ELSE
```

```
         IF (SCRTY .EQ. 3) THEN

            LINE1Y=YMIN-2.0*MARGIN
            LINE2Y=YMIN-3.0*MARGIN

         ELSEIF (SCRTY .EQ. 6) THEN

            LINE1Y=YMIN-1.5*MARGIN
            LINE2Y=YMIN-2.0*MARGIN

         ENDIF

      ENDIF

      WRITE(NUM1,'(F6.1)') LATR

      WRITE(NUM2,'(F6.1)') LONGR

      WRITE(NUM3,'(F6.1)') ORIR

      IF (CHPR) THEN

         IF (PLOTLY .EQ. -5) THEN

            LINE1='Ref. Detector:  Lat. = ' //
     $           NUM1(1:STRLENGTH(NUM1)) //
     $           '  Long. = ' //
     $           NUM2(1:STRLENGTH(NUM2)) //
     $           '  Orient. = ' //
     $           NUM3(1:STRLENGTH(NUM3))

         ELSEIF (PLOTLY .EQ. -6) THEN

            LINE1='Reference Detector:  Latitude = ' //
     $           NUM1(1:STRLENGTH(NUM1)) //
     $           '  Longitude = ' //
     $           NUM2(1:STRLENGTH(NUM2)) //
     $           '  Orientation = ' //
     $           NUM3(1:STRLENGTH(NUM3))

         ENDIF

      ELSE

         IF (SCRTY .EQ. 3) THEN

            LINE1='Ref. Detector:  Lat. = ' //
     $           NUM1(1:STRLENGTH(NUM1)) //
     $           '  Long. = ' //
     $           NUM2(1:STRLENGTH(NUM2)) //
     $           '  Orient. = ' //
     $           NUM3(1:STRLENGTH(NUM3))

         ELSEIF (SCRTY .EQ. 6) THEN

            LINE1='Reference Detector:  Latitude = ' //
     $           NUM1(1:STRLENGTH(NUM1)) //
     $           '  Longitude = ' //
     $           NUM2(1:STRLENGTH(NUM2)) //
     $           '  Orientation = ' //
     $           NUM3(1:STRLENGTH(NUM3))

         ENDIF
```

```
      ENDIF

      WRITE(NUM1,'(F6.1)') LATM

      WRITE(NUM2,'(F6.1)') LONGM

      LINE2='Rotating Detector:  Latitude = ' //
     $        NUM1(1:STRLENGTH(NUM1)) //
     $        '  Longitude = ' //
     $        NUM2(1:STRLENGTH(NUM2))

      LINE1L=STRLENGTH(LINE1)
      LINE2L=STRLENGTH(LINE2)

      CALL MGORELOCATE(LINEX,LINE1Y)

      CALL MGOPUTLABEL(LINE1L,LINE1,8)

      CALL MGORELOCATE(LINEX,LINE2Y)

      CALL MGOPUTLABEL(LINE2L,LINE2,8)

      GLABELX=(SORIM+EORIM)/2.0

      GLABELY=YMAX+0.5*MARGIN

      CALL MGORELOCATE(GLABELX,GLABELY)

      CALL MGOSETANGLE(0.0)

      CALL MGOSETEXPAND(2.0)

      GLABELL=STRLENGTH(GLABEL)

      CALL MGOPUTLABEL(GLABELL,GLABEL,8)

      IF (CHPR) THEN

         CALL MGOPRNTPLOT(NVEC)

      ELSE

         CALL MGOTIDLE

         READ (*, '(A10)') ANSWER

         CALL UPPERCASE(ANSWER)

         IF (ANSWER .EQ. 'P') THEN

            CHPR=.TRUE.
            PLOTLY=-5

            GOTO 160

         ELSEIF (ANSWER .EQ. 'L') THEN

            CHPR=.TRUE.
            PLOTLY=-6

            GOTO 160

         ENDIF
```

```
            ENDIF

      ENDIF

      STOP
      END

      SUBROUTINE DETTENS(SC,CC,D)

      COMPLEX N(3)

      REAL SC(3),CC(3),D(3,3)

      REAL F1,F2,F3,F4

      INTEGER I,J

C     CONSTRUCT THE DETECTOR TENSOR D:

      F1=-SC(3)*CC(1)-SC(2)*CC(3)*SC(1)
      F2=SC(3)*SC(1)-SC(2)*CC(3)*CC(1)
      F3=CC(3)*CC(1)-SC(2)*SC(3)*SC(1)
      F4=-SC(2)*SC(3)*CC(1)-CC(3)*SC(1)
      N(1)=CMPLX(F1,F2)
      N(2)=CMPLX(F3,F4)
      N(3)=CMPLX(SC(1)*CC(2),CC(2)*CC(1))

      DO 10 I=1,3
         DO 20 J=I,3

               D(I,J)=REAL(N(I))*IMAG(N(J))+IMAG(N(I))*REAL(N(J))
               D(J,I)=D(I,J)

 20      CONTINUE
 10   CONTINUE

      RETURN

      END

      LOGICAL FUNCTION DELTR(FILENAME)

      INTEGER SYSRET,UNLINK,STRLENGTH

      CHARACTER*(*) FILENAME

      CHARACTER*80 MESSAGE1,MESSAGE2

      LOGICAL ITISTHERE,GETYESNO

      INQUIRE(FILE=FILENAME,EXIST=ITISTHERE)

      MESSAGE1="An old " // FILENAME(1:STRLENGTH(FILENAME))
     $                    // " file exists. It should be"
      MESSAGE2="removed.  Delete the old " //
     $          FILENAME(1:STRLENGTH(FILENAME)) // " file?"

      IF (ITISTHERE) THEN

         WRITE(6,*) MESSAGE1
         WRITE(6,*) MESSAGE2

         IF (GETYESNO()) THEN
```

```
            SYSRET=UNLINK(FILENAME)

            DELTR=.TRUE.

         ELSE

            DELTR=.FALSE.

         ENDIF

      ELSE

         DELTR=.TRUE.

      ENDIF

      RETURN

      END

      INTEGER FUNCTION STRLENGTH(ASTRING)

      CHARACTER*(*) ASTRING

      INTEGER STRLENGTH,N

      INTEGER I

      N=LEN(ASTRING)

      DO 10 I=N,1,-1

         IF(ASTRING(I:I) .NE. ' ') THEN
            GOTO 20
         ENDIF

 10   CONTINUE

      I=0

      RETURN

 20   CONTINUE

      STRLENGTH=I

      RETURN
      END


C     CONVERTS A FORTRAN STRING TO UPPERCASE

      SUBROUTINE UPPERCASE(ASTRING)

      CHARACTER*26 UPPERCASELETTERS

      CHARACTER*(*) ASTRING

      INTEGER I,J

      UPPERCASELETTERS='ABCDEFGHIJKLMNOPQRSTUVWXYZ'

      DO 10 I=1, LEN(ASTRING)
```

```
      J=INDEX('abcdefghijklmnopqrstuvwxyz',ASTRING(I:I))

      IF (J .NE. 0) THEN

         ASTRING(I:I)=UPPERCASELETTERS(J:J)

      ENDIF

10    CONTINUE

      RETURN
      END


      LOGICAL FUNCTION GETYESNO()

      CHARACTER*10 ANSWER

10    CONTINUE

      READ (*, '(A10)') ANSWER

      CALL UPPERCASE(ANSWER)

      IF (ANSWER .EQ. 'YES') THEN

         GETYESNO=.TRUE.

      ELSEIF (ANSWER .EQ. 'NO') THEN

         GETYESNO=.FALSE.

      ELSE

         PRINT*, "Please answer yes or no."

         GOTO 10

      ENDIF

      RETURN
      END


      SUBROUTINE GETREAL(NUMBER)

      REAL NUMBER

      CHARACTER*80 ANSWER

10    CONTINUE

      READ (*, '(A80)') ANSWER

      READ (ANSWER, *, ERR=20) NUMBER

      RETURN

20    PRINT*, "That does not look like a number to me. Please re-enter:"

      GOTO 10

      END
```

```
      SUBROUTINE GETINT(NUMBER)
      INTEGER NUMBER
      CHARACTER*80 ANSWER
 10   CONTINUE
      READ (*, '(A80)') ANSWER
      READ (ANSWER, *, ERR=20) NUMBER
      RETURN
 20   PRINT*, "That does not look like an integer to me. Please re-enter
     $:"
      GOTO 10
      END
```