ESD-TR 91-079

AD-A241 026

(2)

Technical Report
917

Application of Adaptive Nulling to Electromagnetic Hyperthermia for Improved Thermal Dose Distribution in Cancer Therapy

A.J. Fenn

3 July 1991

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Air Force
under Contract F19628-90-C-0002.

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The work was sponsored by the Department of the Air Force under Contract F19628-90-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The ESD Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Hugh L. Southall

Hugh L. Southall, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

Non-Lincoln Recipients

PLEASE DO NOT RETURN

Permission is given to destroy this document when it is no longer needed.

5 September 1991

ERRATA

Document: Technical Report 917
Application of Adaptive Nulling to Electromagnetic Hyperthermia
for Improved Thermal Dose Distribution in Cancer Therapy
3 July 1991

Errors in the production process require the following corrections:

The figure on page 59 is Figure A-6, and belongs with the caption on page 61.

The figure on page 60 is Figure A-7, and belongs with the caption on page 62.

The figure on page 61 is Figure A-4, and belongs with the caption on page 59.

The figure on page 62 is Figure A-5, and belongs with the caption on page 60.

Publications Office
MIT Lincoln Laboratory
PO Box 73
Lexington, MA 02173-9108

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

APPLICATION OF ADAPTIVE NULLING TO ELECTROMAGNETIC HYPERTHERMIA FOR IMPROVED THERMAL DOSE DISTRIBUTION IN CANCER THERAPY

A.J. FENN
Group 61

TECHNICAL REPORT 917

3 JULY 1991

LEXINGTON MASSACHUSETTS

ABSTRACT

Adaptive nulling is applied to the problem of generating a therapeutic thermal dose distribution in electromagnetic hyperthermia treatment of cancer. A system design concept for implementing adaptive hyperthermia is introduced. With the proposed design concept, it may be possible to maximize the applied electric field at a tumor position in the target body and simultaneously minimize or reduce the electric field at target positions where undesired high temperature regions (hot spots) occur. In a clinical situation, either a gradient search algorithm or sample matrix inversion algorithm would be used to rapidly form the adaptive null (or nulls) prior to any significant tissue heating. Auxiliary short-dipole field probes are used in effecting the desired electric field nulls. An adaptive null formed at the surface of the target has a finite width and extends into the target. This finite null width can allow for noninvasively positioned auxiliary probes in an adaptive hyperthermia system. The allowed spacing or resolution between a deep null and focus is fundamentally equal to the hyperthermia antenna half-power beamwidth in the tissue. A closer spacing between the desired null position and focus is achieved by reducing the null depth via the signal-to-noise ratio at the auxiliary probe.

Analysis of an annular phased-array antenna embedded in an infinite homogeneous medium shows the potential merit of combining adaptive nulling with conventional near-field focusing used in hyperthermia. The analysis is based on a well-known moment-method theory for conducting thin-wire antennas in a homogeneous conducting medium. The theory and software used to compute the moment-method received voltage at a short-dipole probe due to a transmitting dipole array are documented. Computer simulation results are presented for a fully adaptive eight-element annular array operating at 120 MHz and embedded in homogeneous muscle tissue. Multiple simultaneous nulls are used in adaptively generating a desired radiation pattern in the muscle tissue. The power received at a movable short-dipole E-field sensor is then used as the power source in a second computer simulation that performs a transient thermal analysis of an elliptical target body surrounded by a constant-temperature water bolus. The computer simulations show that adaptive nulling can prevent undesired high-temperature regions from occurring while simultaneously heating a deep-seated tumor site.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

LIST OF ILLUSTRATIONS
(Continued)

ACKNOWLEDGMENTS

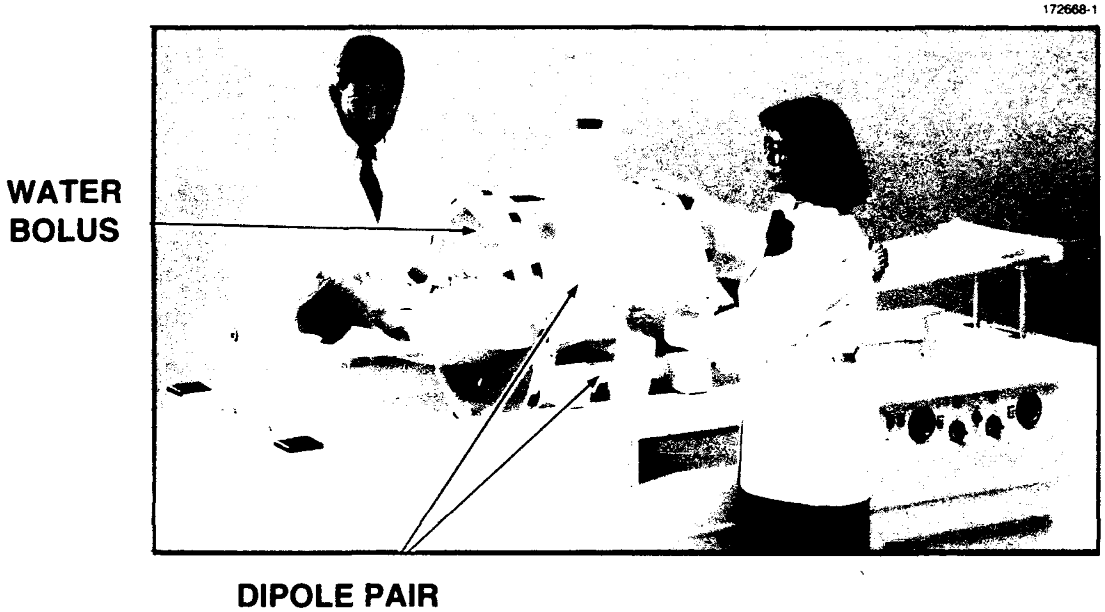
The author wishes to express his gratitude to K.P. Lawton for software support and to D.H. Temme, R.P. Rafuse, D.M. Nathanson, and J.R. Johnson for technical discussions. The encouragement of G.N. Tsandoulas is sincerely appreciated. Technical discussions with numerous individuals outside Lincoln Laboratory are gratefully acknowledged.

1. INTRODUCTION

The successful treatment of deep-seated malignant tumors within a patient is often a difficult task. The objective of the treatment is to reduce in size or completely remove the tumor mass by one or more modalities available at the treatment facility. Common modalities are surgery, chemotherapy, and x-ray therapy [1]. A modality used alone or in conjunction with one of the above modalities is "tissue heating," or hyperthermia [1,2,3,4]. Hyperthermia can be considered as a form of high fever within the body; a controlled thermal dose distribution is required for hyperthermia to have a therapeutic value. Typical localized-hyperthermia temperatures required for therapeutic treatment of cancer are in the 43–45°C range. Normal tissue should be kept at temperatures below 43°C during the treatment. The most difficult aspect of implementing hyperthermia, with either radio-frequency (RF) waves or acoustic (ultrasound) waves, is producing sufficient heating at depth. Multiple-applicator RF hyperthermia arrays are commonly used to provide a focused near-field main beam at the tumor position. A focal region should be concentrated at the tumor with minimal energy delivered to surrounding normal tissue. As the hyperthermia antenna beamwidth is proportional to the wavelength, a small focal region suggests that the RF wavelength be as small as possible. However, due to propagation losses in tissue, the RF depth of penetration decreases with increasing transmit frequency. One of the major problems in heating a deep-seated tumor with a hyperthermia antenna is the formation of undesired "hot spots" in surrounding tissue. This additional undesired heating often produces pain, burns, and blistering in the patient, which requires terminating the treatment immediately. The patient does not receive anesthetics during the hyperthermia treatment in order to provide direct verbal feedback of any pain. Thus, techniques for reducing hot spots are necessary in hyperthermia treatment.

Several journals have published special issues on acoustic and electromagnetic hyperthermia treatment of cancer [5,6]. Electromagnetic transmitting arrays in the frequency band 60–2000 MHz are used to localize heating of malignant tumors within a target body. Many studies have been conducted to produce improved therapeutic field distributions with hyperthermia phased arrays [7–20]. Phase control can be used to synthesize therapeutic RF radiation patterns without adaptive control of the transmit array weights [7–12]. Array transmit weights can be adaptively controlled to maximize the tumor temperature (or RF power delivered to the tumor) while minimizing the surrounding tissue temperature (or RF power delivered to the surrounding tissue) [13–20]. All other studies require invasive techniques to optimize the radiation pattern [13–20]. This report addresses the potential benefit of using adaptive nulling with noninvasive auxiliary sensors to reduce the field intensity at selected positions in the target body while maintaining a desired focus at a tumor.

It is useful to describe certain features of a clinical hyperthermia system used to treat deep-seated tumors. A hyperthermia annular phased-array antenna system (Model BSD-2000, SIGMA-60 applicator [21], BSD Medical Corporation, Salt Lake City, Utah) is shown in Figure 1. By fully surrounding the patient with an annular phased array, it is possible to obtain constructive interference (or signal enhancement) deep within the target volume. This hyperthermia system uses a 60-cm array diameter with eight uniformly spaced dipole elements operating over the frequency



BSD MEDICAL CORP. MODEL BSD-2000	
OPERATING FREQ.	60–120 MHz
NO. CHANNELS	4
POWER	500 W/CHANNEL
ANTENNA	8-DIPOLE RING

Figure 1. A clinical hyperthermia phased-array antenna system.

band 60–120 MHz [22]. The eight dipoles are fed as four active pairs of elements. There are four high-power amplifiers which drive the dipole pairs with up to 500 W average power per channel. Each of the four active channels has an electronically controlled variable-phase shifter for focusing the array. Temperature and electric-field probe sensors (both invasive and noninvasive) are used to monitor the treatment. A cool-water (5–40°C) bolus between the patient and the phased array is used to prevent excess heating of the skin surface. The bolus is filled with circulating distilled water, which has a very low propagation loss. Computer simulation models for this hyperthermia antenna system with inhomogeneous targets have been developed [7,9]. To the author's knowledge, the hyperthermia system in Figure 1 has not been used to form adaptive nulls, although the system may have sufficient temperature and electric-field sensing capability to implement adaptive nulling with an appropriate transmit-weight control algorithm. A candidate adaptive nulling algorithm would be a gradient search based on minimizing the signal received by electric-field sensors at the desired null positions while maximizing the signal at the focal point. The reason for considering this algorithm is that the BSD-2000 system is currently capable of measuring the applied electric-field amplitude but not the phase. Note: the E-field detectors used in the BSD-2000 system are semiconductor diode detectors. A potentially better approach to performing adaptive hyperthermia is to use a channel-covariance-matrix-based algorithm to control the weights. If the electric-field phase and amplitude are measured (by modification of the E-field probe hardware to form in-phase and quadrature signals) the sample matrix inversion (SMI) algorithm [23] could be used to form the nulls. The concept of adaptive nulling as it applies to hyperthermia will now be described.

Recently, the author has been investigating a technique called focused near-field adaptive nulling [24–30], for which the general concept is shown in Figure 2. A calibration probe antenna is used to focus the main beam at approximately one diameter D of the phased array. Additional (auxiliary) probe antennas are located in the sidelobe region of the phased-array quiescent radiation pattern. Adaptive nulls are formed in the direction of these auxiliary probes. Computer simulations show that a radar system can be tested by using the focused near-field nulling technique [29]. Experimental measurements of focused near-field adaptive nulling are available [30]. During the radar system testing study, similarities between the radar application and electromagnetic hyperthermia were noticed by the author. These similarities are that 1) phased-array near-field focusing is used to form the main beam and 2) uncontrolled or high sidelobes are potentially deleterious to system performance. For the radar system, interference signals entering through uncontrolled sidelobes can reduce the signal-to-noise ratio (SNR); adaptive nulling [31] is commonly used to counteract this possible degradation. In the hyperthermia system, high-transmit antenna sidelobes can give rise to hot spots in the target tissue. These high-temperature regions could possibly be alleviated with adaptive nulling of the electric-field sidelobes in the target. Much of the software developed for the radar system testing analysis may be readily modified to handle adaptive hyperthermia analysis.

There is continued interest at Lincoln Laboratory in applying techniques used in Department of Defense (DoD)-sponsored research to other disciplines, such as medical applications. This study applies radar system testing technology (currently in the form of software simulation) to the medical application of hyperthermia. Figure 3 depicts the similarities between radar system near-field

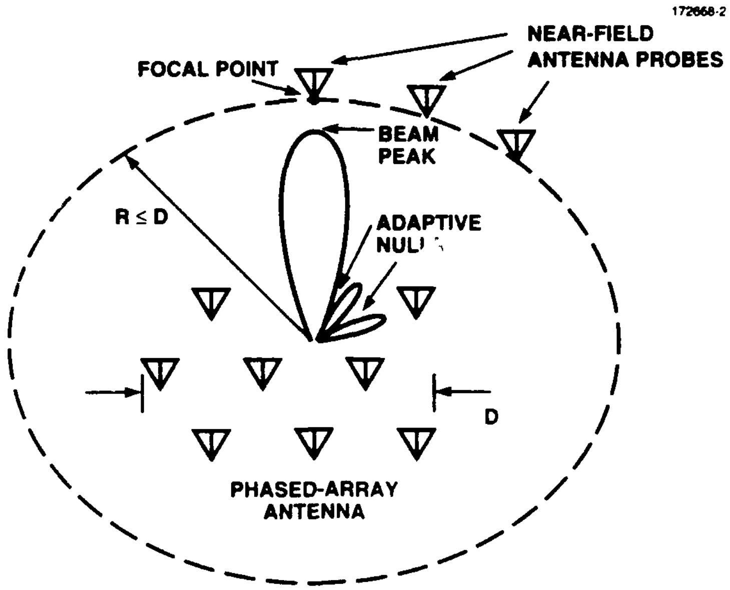


Figure 2. General concept for focused near-field adaptive nulling.

testing and electromagnetic hyperthermia treatment of cancer. In Figure 3(a), a radar phased-array antenna system is positioned in an anechoic test chamber with radiating sources that simulate typical radar signals. Notice that the test antenna is focused in the near field at one aperture diameter. Computer simulations have shown that the near-field signals (jammer, radar clutter, and target) received and processed by an adaptive radar system are equivalent to fielded radar conditions [29]. In Figure 3(b), a hyperthermia (transmit) phased array is used to therapeutically illuminate a target body by focusing its electromagnetic energy on a malignant tumor deep within the body. Nulls are formed adaptively to reduce the electromagnetic energy delivered to potential hot spots. The nulls are shown invasive to the target; however, noninvasive null positions on the surface of the target are also possible as shown in this report.

An example of a deep-seated tumor is cancer of the prostate [1]. The tumor volume often has a decreased blood flow which aids in heating the tumor, compared to normal tissue for which heat is carried away by normal blood flow. Note: a blood-flow thermal analysis [32,33] is beyond the scope of this report; we would like to show initially that with a computer simulation of RF-illuminated phantom muscle tissue [34] adaptive nulling can improve the thermal dose distribution. The thermal analysis used in this report is similar to the method presented in Zhang, et al. [35], where blood flow effects are ignored. In practice, undesired high-temperature regions away from the focus can occur inside the target volume. For example, scar tissue, which has a decreased blood flow rate, will tend to heat up more rapidly than normal tissue. In the proposed adaptive hyperthermia concept, electric-field nulls are used to reduce the power delivered to potential hot spots. Computer simulations show that noninvasive field probes can be used to eliminate hot spots interior to the phantom target tissue.

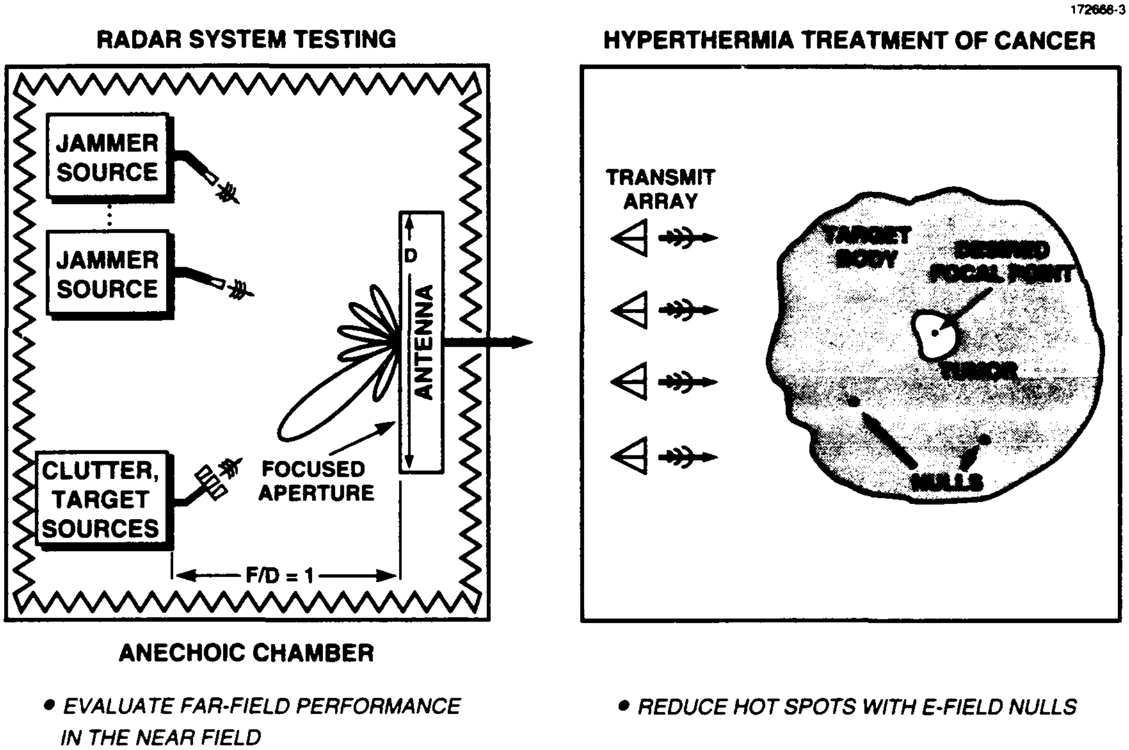


Figure 3. Two applications of focused near-field adaptive nulling: (a) Radar system testing and (b) Hyperthermia treatment of cancer.

An antenna analysis code (called WIRES) capable of analyzing a wide variety of antenna or radar cross section problems has been developed by J.H. Richmond [36–38]. The software can analyze complex geometries involving multiple-connected and/or isolated wires and is well-suited to analyzing low-frequency phased arrays radiating into an infinite homogeneous conducting medium. WIRES is a moment-method code that uses the electric field integral equation (EFIE) to enforce the boundary condition of the tangential electric field being zero at the surface of the antenna of interest. The moment-method basis and testing functions used in this code are piecewise sinusoidal. For a complete description of the theory and capabilities of the WIRES code, the reader is referred to the user's manual [37]. The WIRES software has been modified by the author to include near-field focusing and adaptive nulling capabilities. This modified code has been used to analyze the near-field and far-field adaptive nulling performance of thin-wire phased arrays in free space [27,29]. A new version of the thin-wire code that can analyze adaptive hyperthermia arrays in an infinite homogeneous conducting medium has recently been written and is documented in this report.

This report is organized in the following manner. Section 2 discusses the concept for a noninvasive adaptive hyperthermia system and describes in detail the theory used in performing the focused near-field adaptive nulling simulation. Pertinent theory for wave propagation in a conducting medium is reviewed. A brief description of the transient thermal analysis approach used to compute the temperature distribution in a target is given. Section 3 presents simulation results for a fully adaptive eight-element hyperthermia ring array operating at 120 MHz. The calculated received RF-power distributions, before and after adaptive nulling, at a short-dipole field probe with both the field probe and the hyperthermia array embedded in homogeneous muscle tissue, are presented. The RF power distribution is then fed into transient thermal analysis software which computes the two-dimensional temperature distribution in an elliptical muscle-tissue target surrounded with a constant-temperature water bolus. The temperature distribution obtained by using multiple independent adaptive nulls clearly indicates the elimination of undesired hot spots. Conclusions are made in Section 4. Performance degradation due to insufficient number of auxiliary null positions is described in Appendix A. Appendix B contains data files and source code for the focused near-field adaptive nulling computer simulations.

2. THEORY

2.1 NONINVASIVE ADAPTIVE HYPERTHERMIA SYSTEM CONCEPT

The concept of a noninvasive adaptive-nulling hyperthermia system [39,40] is shown in Figure 4. Theoretically, to generate the desired field distribution in a clinical adaptive hyperthermia system, receiving sensors are positioned as close as possible to the focus (tumor site) and to where high temperatures are to be avoided (such as near the spinal cord and scar tissue). As simulations in this report will show, for an annular array configuration the receiving sensors can be located noninvasively on the surface (skin) of the target. Initially, the hyperthermia array is focused to produce the required field intensity at the tumor. An invasive probe is required to achieve the optimum focus at depth. To avoid undesired hot spots, it is necessary to minimize the power received at the desired null positions and to constrain the array weights to deliver a required amount of transmitted or focal region power. The adaptive array weights (with gain g and phase ϕ) are controlled by either the SMI algorithm or a gradient search algorithm to rapidly (within seconds) form the nulls before a significant amount of target heating takes place. With this adaptive technique, it should be possible to avoid unintentional hot spots and maintain a therapeutic thermal dose distribution at the tumor.

Figure 5(a) shows a specific example of an eight-element hyperthermia ring phased array with a target cross section at the prostate level [9]. Figure 5(b) shows an elliptical phantom target which is used to model the prostate-level cross section. A noninvasive adaptive nulling system is achieved by placing auxiliary sensors $1, 2, \cdots, N_{aux}$ on the target skin as shown. The null zones centered at each auxiliary probe naturally extend into the elliptical target region to eliminate undesired hot spots. (An example of a two-dimensional near-field null zone is shown in Figure 10(b) of Fenn [28]). The width of each null zone is directly related to the depth of each null. The depth of each null (sometimes referred to as the amount of cancellation) is directly related to the SNR at the sensor position. A low SNR produces a small amount of nulling, and a high SNR produces a large amount of nulling. The resolution or minimum spacing between the focus and null position is normally equal to the half-power beamwidth of the antenna. The resolution is enhanced somewhat by using weak nulls whenever the separation between the null and focus is closer than the half-power beamwidth.

Figure 6 summarizes the thermal simulation for a hyperthermia ring array transmitting into a homogeneous elliptical target region. The tumor site is assumed to be at the center of the ellipse. The thermal distribution in Figure 6(a) contains two undesired hot spots to the left and right of the focus. In Figure 6(b), adaptive nulling is applied and the hot spots are eliminated. The details of the computer simulations are contained in Section 3.

In the near-field nulling technique described here, it is assumed that the hyperthermia phased-array antenna is focused (as it normally is) in the near field and that a main beam and possibly sidelobes are formed in the target. In this report, it is assumed that phase focusing is used to produce the desired quiescent main beam. Figure 7 shows a primary (calibration) probe antenna located at a desired focal point of an array. The array can maximize the signal received by the

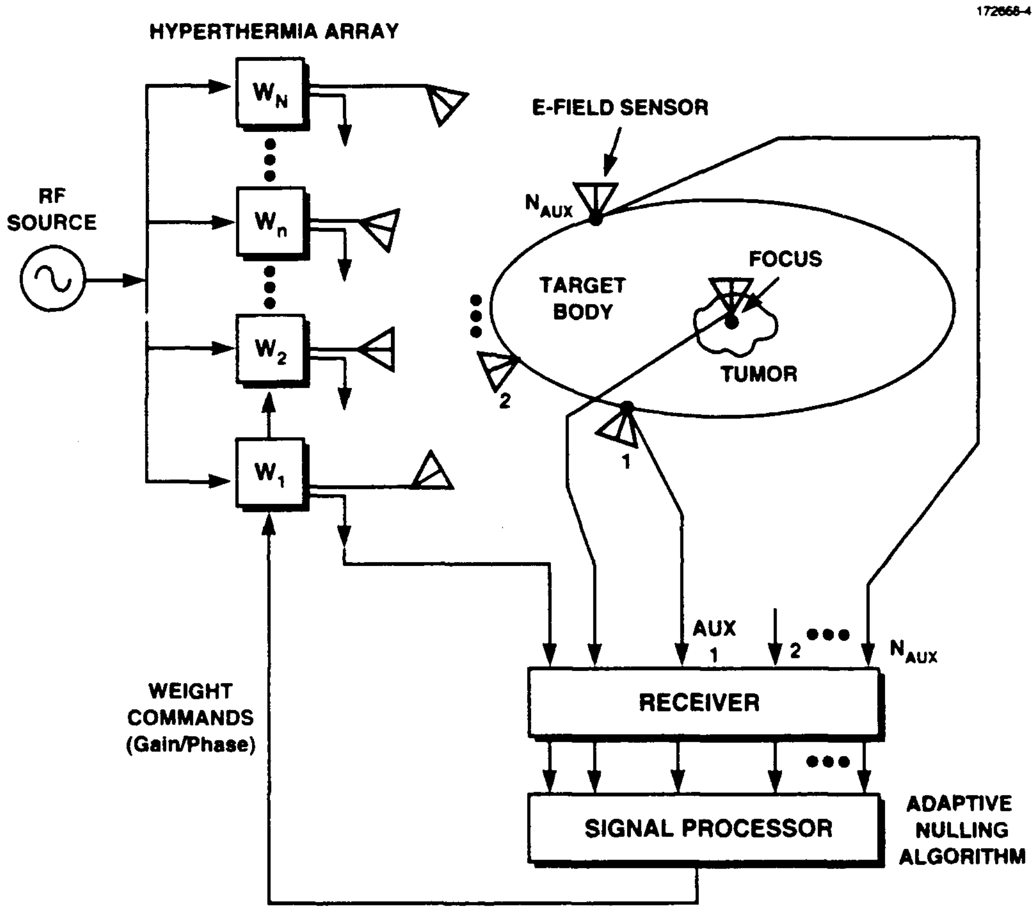


Figure 4. Noninvasive adaptive hyperthermia system concept.

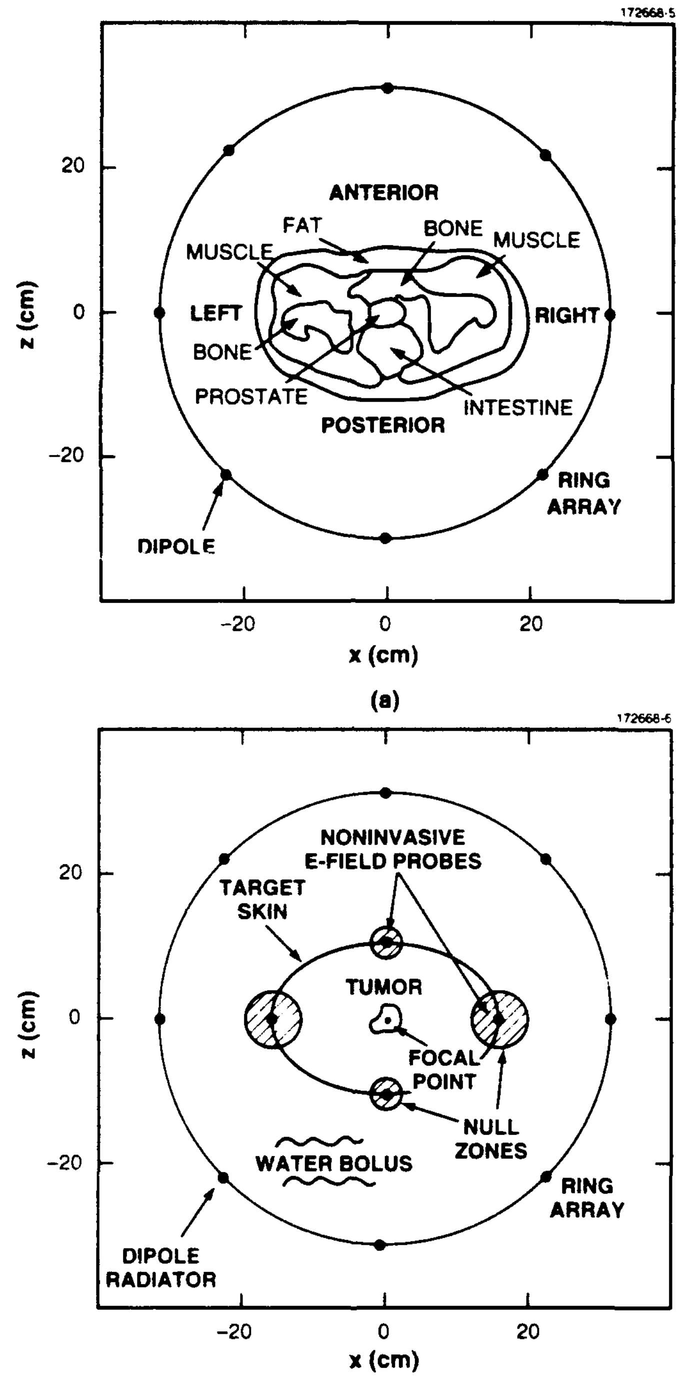


Figure 5. (a) Transverse cross section at the prostate level. An eight-element hyperthermia ring array of dipole elements is used to irradiate a target body with sufficient power to elevate the temperature of the target body to a therapeutic level. (b) Elliptical phantom target used to model the prostate (target)-level cross section. Shown are four noninvasive auxiliary E-field probes that are used in forming null zones with widths controlled by the individual null depths. The null zones extend into the body to reduce the internal electric field at specified locations.

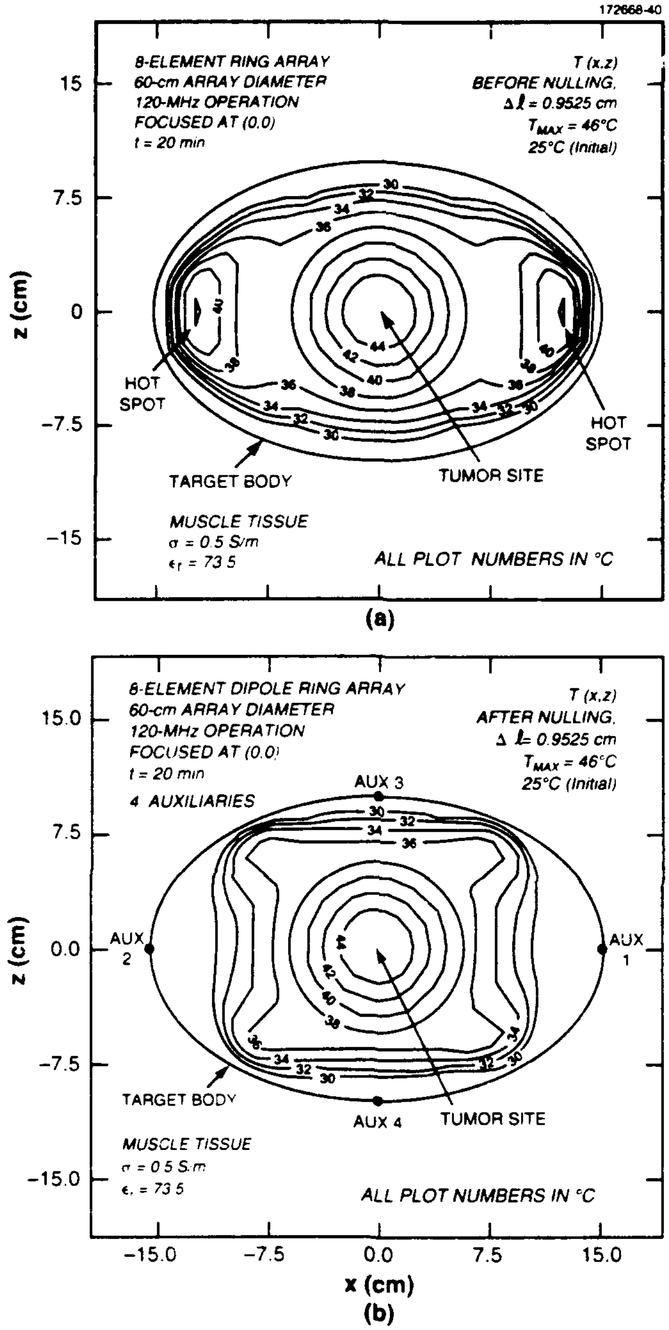


Figure 6. Simulated two-dimensional thermal pattern at time $t = 20$ minutes in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The incident RF power distribution is at 120 MHz and the initial temperature of the phantom is 25° C. Temperature contour levels are given in 2° C steps. (a) Before nulling; hot spots on the left and right sides of the target are present. (b) After nulling; hot spots are eliminated.

calibration probe by adjusting its phase shifters so that the observed element-to-element phase variation is removed. One way to do this in a numerical simulation is to choose a reference path length as the distance from the focal point to the phase center of the array. This distance is denoted r_F and the distance from the focal point to the nth array element is denoted r_n^F. The voltage received at a probe (located at the focal point) due to the nth array element is computed here using the method of moments as described in Section 2.3. To maximize the received voltage at the focal-point probe output, it is necessary to apply the phase conjugate of the observed signals, due to the array elements, at the transmit array. The resulting near-field radiation pattern will have a main beam and sidelobes. The main beam will be pointed at the array focal point, and sidelobes will exist at angles away from the main beam. Auxiliary probe antennas can then be placed at the desired null positions in the quiescent sidelobe region. These sidelobes are where tissue hot spots are likely to occur; they are nulled by the adaptive nulling algorithms described below.

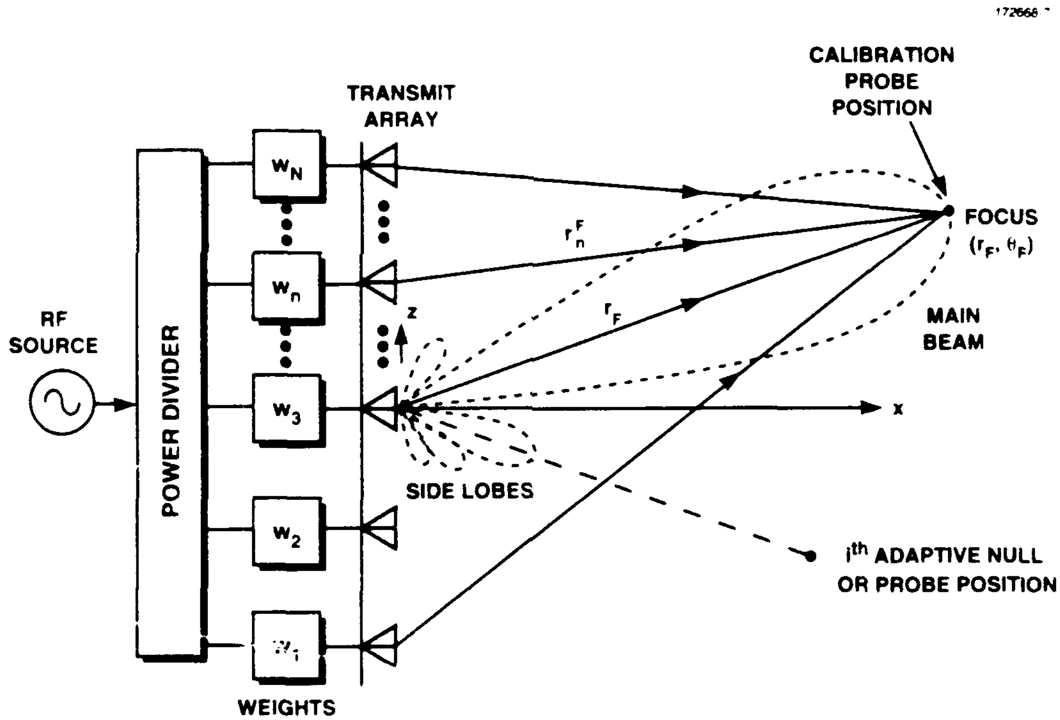


Figure 7. Adaptive transmit phased-array antenna near-field focusing concept.

2.2 ADAPTIVE TRANSMIT ARRAY FORMULATION

Consider the hyperthermia array and probe geometry shown in Figure 7. Typically, the hyperthermia array contains N identical antenna elements. The input signal to each of the N array elements is obtained from the weighted signal distributed by the power divider. The number of adaptive channels is denoted M, and for a fully adaptive array $M = N$. Note: the report will concentrate on fully adaptive arrays (arrays in which all transmit array elements have adaptive weight control); however, the computer simulation program is developed for both fully adaptive and sidelobe canceller arrays. In this report, ideal transmit weights (a complex voltage gain vector) are assumed in the computer simulation, with $\boldsymbol{w} = (w_1, w_2, \cdots, w_N)^T$ denoting the adaptive channel weight vector as shown in Figure 4.[1] (Superscript T means transpose). To generate adaptive nulls, the transmit weights (phase and gain) are controlled by either the SMI algorithm or a gradient search algorithm. The SMI algorithm has the flexibility to operate in either open- or closed-loop feedback modes [41]; the gradient search algorithm operates only in a feedback mode.

2.2.1 Sample Matrix Inversion Algorithm

With the SMI algorithm [23], the fundamental quantities required to fully characterize the incident field for adaptive nulling purposes are the adaptive channel cross correlations. To implement this algorithm it is necessary to know the complex received voltage at the auxiliary probes. For example, the moment-method formulation (described in Section 2.3) allows computation of the complex-received voltage at the auxiliary probes.

The block diagram in Figure 8 shows how the SMI algorithm is used in the hyperthermia analysis. The diagram indicates four performance measures used to quantify the computer simulations: electric-field distribution, covariance matrix eigenvalues, adaptive transmit weights, and interference cancellation. The calculation of these performance measures is described in detail below.

Referencing Figure 7, let a spherical wavefront be incident at the ith probe antenna, due to each array element (radiating one at a time with a unity-amplitude reference signal), which results in a set of probe-received complex voltages denoted $v_1^i, v_2^i, \cdots, v_N^i$. The cross correlation R_{mn}^i of the received voltages due to the mth and nth adaptive transmit channels at the ith probe is given by

$$R_{mn}^i = \boldsymbol{E}(v_m v_n^*) \quad . \tag{1}$$

[1]The effects of transmit-weight quantization and transmit-weight random errors are addressed in the software.

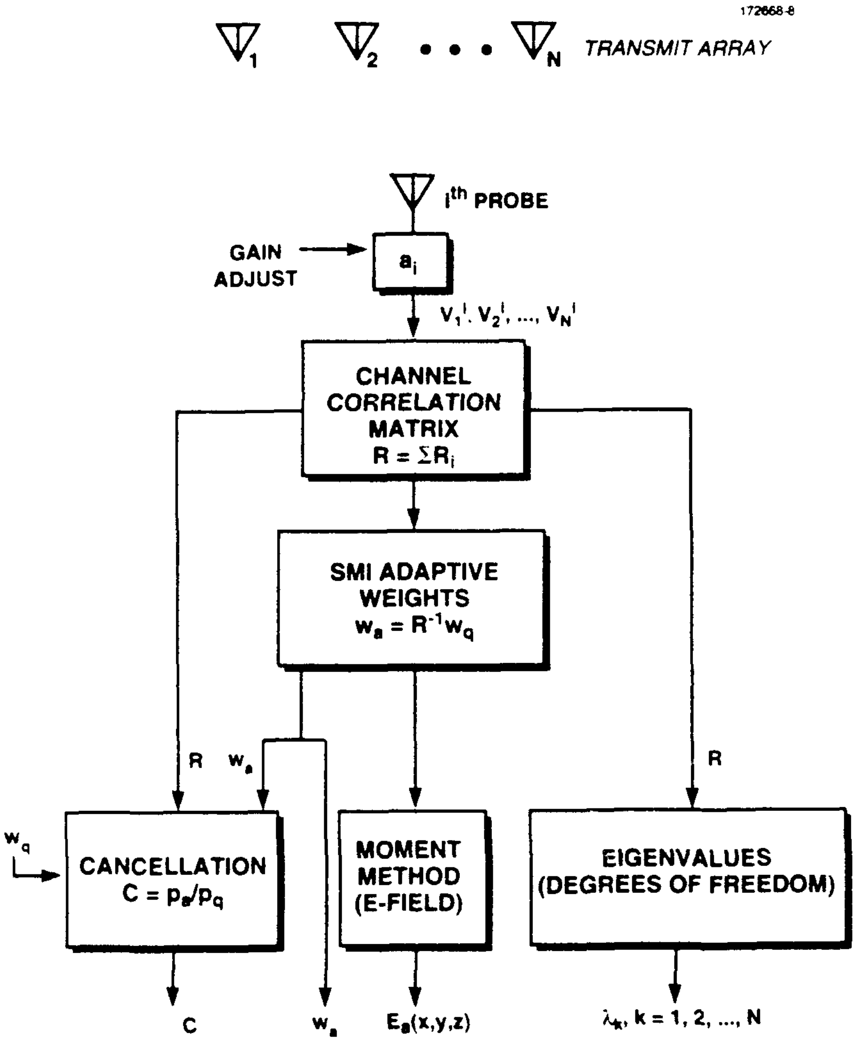


Figure 8. Implementation of the SMI algorithm for an adaptive hyperthermia system. Four performance measures are used in evaluating the adaptive response of the system: electric-field distribution, covariance matrix eigenvalues, adaptive transmit weights, and interference cancellation.

where * means complex conjugate and $\boldsymbol{E}(\cdot)$ means mathematical expectation. Because v_m and v_n represent voltages of the same waveform but at different times, R^i_{mn} is also referred to as an autocorrelation function. Note: for convenience, in Equation (1) the superscript i in v_m and in v_n has been omitted.

In the frequency domain, assuming the transmit waveform has a band-limited white noise power spectral density (as commonly assumed in jamming of a radar system), Equation (1) can be expressed as the frequency average

$$R^i_{mn} = \frac{1}{B}\int_{f_1}^{f_2} v_m(f)v_n^*(f)\,df \quad , \tag{2}$$

where $B = f_2 - f_1$ is the nulling bandwidth and f is the frequency. It should be noted that $v_m(f)$ takes into account the transmit wavefront shape, which is spherical for the hyperthermia application. For the special case of a continuous wave (CW) transmit waveform, as normally used in hyperthermia, the cross correlation reduces to

$$R^i_{mn} = v_m(f_o)v_n^*(f_o) \quad , \tag{3}$$

where f_o is the transmit frequency of the hyperthermia array.

Let the channel or interference covariance matrix be denoted $\boldsymbol{R}$. (Note: in hyperthermia, interference is used to refer to the signals received at the auxiliary probes. The undesired hot spots can be thought of as interfering with the therapy.) If there are J_{aux} independent desired null positions or auxiliary probes, the J_{aux}-probe covariance matrix is the sum of the covariance matrices observed at the individual probes; that is,

$$\boldsymbol{R} = \sum_{i=1}^{J_{aux}} \boldsymbol{R}_i + \boldsymbol{I} \quad , \tag{4}$$

where $\boldsymbol{R}_i$ is the sample covariance matrix observed at the ith probe and $\boldsymbol{I}$ is the identity matrix used to represent the thermal noise level of the receiver. (It should be noted that both N_{aux} and J_{aux} can be used interchangeably to describe the number of auxiliary nulling sensors or received-interference signal positions. The symbol J_{aux} is perhaps more descriptive and can equivalently be thought of as the number of positions where jamming signals are nulled).

Prior to generating an adaptive null, the adaptive channel weight vector, $\boldsymbol{w}$, is chosen to synthesize a desired quiescent radiation pattern. When nulling is desired, the optimum set of transmit weights to form an adaptive null (or nulls), denoted $\boldsymbol{w}_a$, is computed [23] by

$$\boldsymbol{w}_a = \boldsymbol{R}^{-1}\boldsymbol{w}_q \quad . \tag{5}$$

where $^{-1}$ means inverse and $\boldsymbol{w}_q$ is the quiescent weight vector. During array calibration, the normalized quiescent transmit weight vector, with element 1 radiating, is chosen to be $\boldsymbol{w}_q = (1, 0, 0, \cdots, 0)^T$; that is, the transmit channel weight of element 1 is unity and the remaining transmit channel weights are zero. Similar weight settings are used to calibrate the remaining transmit elements. For a fully adaptive annular array focused at the origin in homogeneous tissue, the normalized quiescent weight vector is simply $\boldsymbol{w}_q = (1, 1, 1, \cdots, 1)^T$. Commonly, the weight vector is constrained to deliver a required amount of power to the hyperthermia array or to the tumor. For simplicity in the software used to analyze the hyperthermia array, the weights are constrained such that

$$\sum_{m=1}^{M} |w_n|^2 = 1 \quad , \tag{6}$$

where w_n is the transmit weight for the nth element. Note: in the computer simulations, the electric field due to the normalized weight vector is scaled appropriately to deliver the required amount of power to the tissue so that a desired focal-region temperature level is achieved after t minutes.

The summation of power received at the probes is given by

$$p = \boldsymbol{w}^\dagger \boldsymbol{R} \boldsymbol{w} \quad , \tag{7}$$

where † means complex conjugate transpose. The interference-plus-noise-to-noise ratio, denoted INR, is computed as the ratio of the auxiliary probe array output power [defined in Equation (7)] with the transmit signal present to the probe array output power with only receiver noise present; that is,

$$INR = \frac{\boldsymbol{w}^\dagger \boldsymbol{R} \boldsymbol{w}}{\boldsymbol{w}^\dagger \boldsymbol{w}} \quad . \tag{8}$$

The adaptive array cancellation ratio, denoted C, is defined here as the ratio of the summation of probe-received power after adaptation to the summation of probe-received power before adaptation; that is,

$$C = \frac{p_a}{p_q} \quad . \tag{9}$$

Substituting Equation (7) in Equation (9) yields

$$C = \frac{{\boldsymbol{w}_a}^\dagger \boldsymbol{R} \boldsymbol{w}_a}{{\boldsymbol{w}_q}^\dagger \boldsymbol{R} \boldsymbol{w}_q} \quad . \tag{10}$$

Next, the covariance matrix defined by the elements in Equations (2) or (3) is Hermitian (that is, $\boldsymbol{R} = \boldsymbol{R}^\dagger$), which, by the spectral theorem, can be decomposed in eigenspace [42] as

$$\boldsymbol{R} = \sum_{k=1}^{M} \lambda_k \boldsymbol{e}_k \boldsymbol{e}_k^\dagger \quad , \tag{11}$$

where $\lambda_k, k = 1, 2, \cdots, M$ are the eigenvalues of $\boldsymbol{R}$, and $\boldsymbol{e}_k, k = 1, 2, \cdots, M$ are the associated eigenvectors of $\boldsymbol{R}$. The interference covariance matrix eigenvalues $(\lambda_1, \lambda_2, \cdots, \lambda_M)$ are a convenient quantitative measure of the use of the adaptive array degrees of freedom [43]. The amplitude spread between the largest and smallest eigenvalues is a quantitative measure of the dynamic range of the interference (hot spot) signals.

2.2.2 Gradient Search Algorithm

Under conditions where only the probe-received voltage amplitude is measured (as in the BSD-2000 system previously mentioned), it is appropriate to consider a gradient search algorithm to minimize the interference power at selected positions. The gradient search is used to iteratively control the transmit weights so that the RF signal received by the probe array is minimized. The transmit array weights (gain and phase) are adaptively changed in small increments and the probe array output power is monitored to determine weight settings that reduce the output power most rapidly to a null. The mathematical formulation for the gradient search is developed in a straightforward manner [44] and will not be considered further here. Only the sample covariance matrix inversion algorithm has been implemented in the focused near-field nulling simulation presented in this report.

2.3 MOMENT-METHOD FORMULATION

This section describes a method of moments formulation to compute the probe-received voltages in Equation (2) due to the transmitting hyperthermia phased-array antenna in an infinite homogeneous conducting medium. The medium is described by the three parameters μ, ϵ, and σ, which are discussed in the next section. The formulation given here is analogous to that developed under array-receiving conditions for an adaptive radar [29]. The software used to analyze a hyperthermia array is based on the receive-array analogy but the theory presented below is given in the context of a transmit array. Referring to Figure 9, assume that each transmit element is fed with a generator having a known impedance Z_L. Let $v_{n,j}^{o.c.}$ represent the open-circuit voltage at the jth probe due to the nth transmit-array element. Here, the jth probe can denote either the focal point calibration probe or one of the auxiliary probes used to null a sidelobe. The number of auxiliary probes is denoted by J_{aux}. The receive probe is assumed to be terminated in an impedance Z_r.

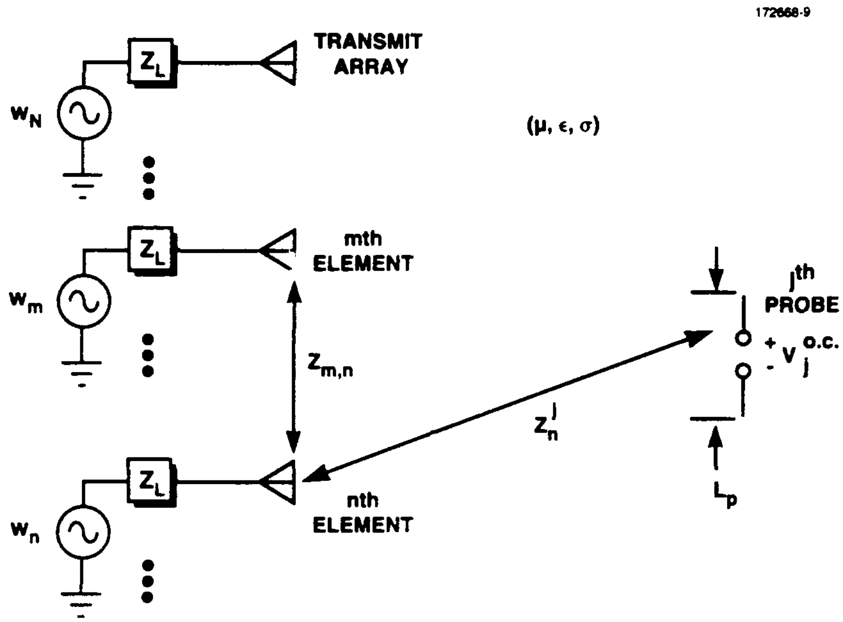


Figure 9. Transmit array and receive antenna probe. Open-circuit mutual impedance between array elements is denoted $Z_{m,n}$. The open-circuit voltage at the probe is computed from the array terminal currents and from Z_n^j, the open-circuit mutual impedance between the nth array element and the jth probe antenna.

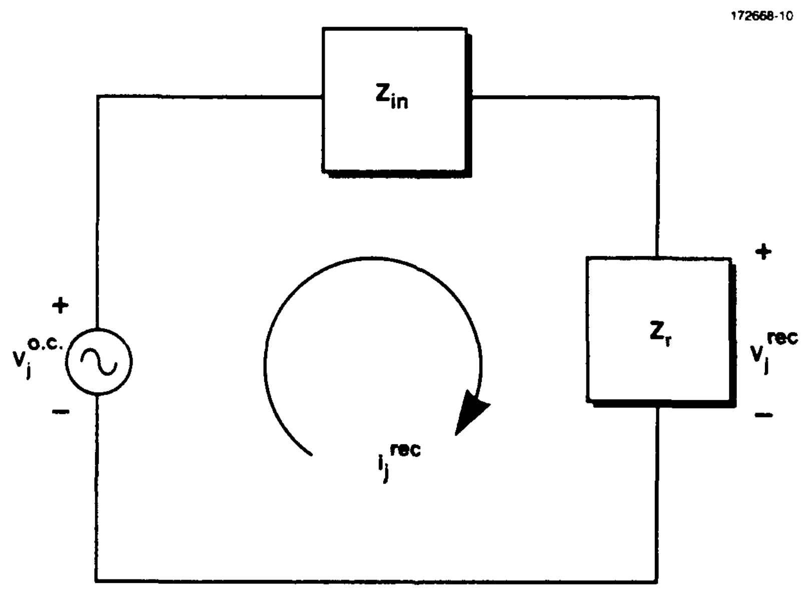


Figure 10. Equivalent circuit model for receive antenna probe.

Next, let $\boldsymbol{Z}$ denote the open-circuit mutual impedance matrix (with dimensions $N \times N$ for the N-element array). The open-circuit mutual impedance between array elements m and n is denoted $Z_{m,n}$. It is assumed that multiple interaction between the hyperthermia array and the probe sensor can be neglected. Thus, the hyperthermia array terminal current vector $\boldsymbol{i}$ can be computed in terms of the transmit weights $\boldsymbol{w}$ as

$$\boldsymbol{i} = [\boldsymbol{Z} + Z_L \boldsymbol{I}]^{-1} \boldsymbol{w} \quad . \tag{12}$$

Next, let Z_n^j be the open-circuit mutual impedance between the jth probe and the nth array element. The induced open-circuit voltage $v_{n,j}^{o.c.}$ at the jth receive probe, due to the nth array element transmit current i_n, can then be expressed as

$$v_{n,j}^{o.c.} = Z_n^j \cdot i_n \quad . \tag{13}$$

In matrix form, the induced open-circuit probe-voltage matrix $\boldsymbol{v}_{probe}^{o.c.}$ is

$$\boldsymbol{v}_{probe}^{o.c.} = \boldsymbol{Z}_{probe,array} \boldsymbol{i} \tag{14}$$

or

$$\boldsymbol{v}_{probe}^{o.c.} = \boldsymbol{Z}_{probe,array} [\boldsymbol{Z} + Z_L \boldsymbol{I}]^{-1} \boldsymbol{w} \quad , \tag{15}$$

where $\boldsymbol{Z}_{probe,array}$ is a rectangular matrix of order $J_{aux} \times N$ for the open-circuit mutual impedance between the probe array and the hyperthermia array. Note: the jth row of the matrix $\boldsymbol{Z}_{probe,array}$ is written as $(Z_1^j, Z_2^j, \cdots, Z_N^j)$, where $j = 1, 2, \cdots, J_{aux}$. The receive voltage matrix is then computed by the receiving circuit equivalence theorem for an antenna [50]. The receive-antenna equivalent circuit is depicted in Figure 10, where it is readily determined that

$$\boldsymbol{v}_{probe}^{rec} = \boldsymbol{v}_{probe}^{o.c.} \frac{Z_r}{Z_{in} + Z_r} \quad , \tag{16}$$

where Z_{in} is the input impedance of the probe. It should be noted that the $\boldsymbol{v}_{probe}^{rec}$ matrix is a column vector of length J_{aux} and v_j^{rec} is the jth element of the matrix. The probe-receive current matrix is given by

$$\boldsymbol{i}_{probe}^{rec} = \boldsymbol{v}_{probe}^{o.c.} \frac{1}{Z_{in} + Z_r} \quad . \tag{17}$$

The jth element of the column vector $\boldsymbol{i}_{probe}^{rec}$ is denoted $i_j^{rec}, j = 1, 2, \cdots, J_{aux}$. Finally, the power received by the jth probe is

$$p_j^{rec} = \frac{1}{2} Re(v_j^{rec} \cdot i_j^{rec^*}) \qquad , \tag{18}$$

where Re means real part. Substituting Equations (16) and (17) into Equation (18) yields

$$p_j^{rec} = \frac{1}{2} |v_j^{o.c.}|^2 \frac{Re(Z_r)}{|Z_{in} + Z_r|^2} \qquad . \tag{19}$$

The total interference power received by the auxiliary probe array is given by

$$p^{rec} = \sum_{j=1}^{J_{aux}} p_j^{rec} \qquad . \tag{20}$$

The incident electric field E is related to the open-circuit voltage $v^{o.c.}$ by the effective height h of the probe antenna [48] as

$$v^{o.c.} = hE \qquad . \tag{21}$$

If the length L_p of the probe antenna is approximately 0.1λ or less, the current distribution is triangular and the effective height is $h = 0.5L_p$ [48]. Thus, for a short-dipole probe the open-circuit voltage can be expressed as

$$v^{o.c.} = \frac{L_p}{2} E \qquad . \tag{22}$$

It then follows from Equation (22) that the E field for a short-dipole probe at position (x, y, z) is given by

$$E(x, y, z) = \frac{2v^{o.c.}(x, y, z)}{L_p} \qquad . \tag{23}$$

Finally, the quiescent and adapted E-field radiation patterns are computed using the quiescent and adapted weight vectors $\boldsymbol{w}_q$ and $\boldsymbol{w}_a$, respectively, in Equations (15) and (23).

The moment-method expansion and testing functions are assumed to be sinusoidal. The open-circuit mutual impedances in Equation (15) between thin-wire dipoles in a homogeneous conducting medium are computed based on subroutines from a well-known moment-method computer code [38]. In evaluating Z_n^j for the jth auxiliary probe, double precision computations are used.

As mentioned previously, the array is calibrated (phased focused) initially using a short dipole at the focal point. To accomplish this numerically, having computed $\boldsymbol{v}_{focus}^{rec}$, the receive array weight vector $\boldsymbol{w}$ will have its phase commands set equal to the conjugate of the corresponding phases in $\boldsymbol{v}_{focus}^{rec}$. Transmit antenna radiation patterns are obtained by scanning (moving) a dipole probe with half-length l in the near-field and computing the receive probe-voltage response.

The received voltage matrix for the jth probe (denoted $\boldsymbol{v}_j^{rec}$) is computed at K frequencies across the nulling bandwidth. Thus, $\boldsymbol{v}_j^{rec}(f_1), \boldsymbol{v}_j^{rec}(f_2), \cdots, \boldsymbol{v}_j^{rec}(f_K)$ are needed. In this report, the impedance matrix is computed at K frequencies and is inverted K times. The probe covariance matrix elements are computed by evaluating Equation (2) numerically, using Simpson's rule numerical integration. For multiple auxiliary probes, the covariance matrix is evaluated using Equation (3). Adaptive array radiation patterns are computed by superimposing the quiescent radiation pattern with the weighted sum of auxiliary-channel-received voltages.

2.3.1 WAVE PROPAGATION IN CONDUCTING MEDIA

To gain insight into the effect of a lossy medium on the propagation of an electromagnetic wave, it is useful to review certain fundamental equations which govern the field characteristics. In a conducting medium, Maxwell's curl equations in time-harmonic form are

$$\nabla \times \boldsymbol{H} = \boldsymbol{J} + j\omega\epsilon \boldsymbol{E} \tag{24}$$

$$\nabla \times \boldsymbol{E} = -j\omega\mu \boldsymbol{H} \quad , \tag{25}$$

where $\boldsymbol{E}$ and $\boldsymbol{H}$ are the electric and magnetic fields, respectively, $\boldsymbol{J}$ is the conduction current density, $\omega = 2\pi f$ is the radian frequency, ϵ is the permittivity of the medium, and μ is the permeability of the medium. The permittivity is expressed as $\epsilon = \epsilon_r \epsilon_o$, where ϵ_r is the dielectric constant (relative permittivity) and ϵ_o is the permittivity of free space. Similarly, $\mu = \mu_r \mu_o$, where μ_r is the relative permeability and μ_o is the permeability of free space. For a medium with electrical conductivity σ, $\boldsymbol{J}$ and $\boldsymbol{E}$ are related as

$$\boldsymbol{J} = \sigma \boldsymbol{E} \quad . \tag{26}$$

Substituting Equation (26) into Equation (24) yields

$$\nabla \times \boldsymbol{H} = (\sigma + j\omega\epsilon) \boldsymbol{E} \quad . \tag{27}$$

From Equations (24) and (25), the vector wave equation in terms of $\boldsymbol{E}$ is derived as

$$\nabla^2 \boldsymbol{E} - \gamma^2 \boldsymbol{E} = 0 \quad . \tag{28}$$

It is readily shown [49] that

$$\gamma = \pm\sqrt{j\omega\mu(\sigma + j\omega\epsilon)} = \pm j\omega\sqrt{\mu\epsilon}\sqrt{1 - j\frac{\sigma}{\omega\epsilon}} \quad . \tag{29}$$

The quantity $\sigma/\omega\epsilon$ is referred to as the loss tangent. It is common to express the complex propagation constant as

$$\gamma = \alpha + j\beta \quad , \tag{30}$$

where α is the attenuation constant and β is the phase constant. The constants α and β are found by setting Equation (29) equal to Equation (30) and then squaring both sides, equating the real and imaginary parts, and solving the pair of simultaneous equations, with the result

$$\alpha = \frac{\omega\sqrt{\mu\epsilon}}{\sqrt{2}}\{\sqrt{1 + (\frac{\sigma}{\omega\epsilon})^2} - 1\}^{1/2} \tag{31}$$

$$\beta = \frac{\omega\sqrt{\mu\epsilon}}{\sqrt{2}}\{\sqrt{1 + (\frac{\sigma}{\omega\epsilon})^2} + 1\}^{1/2} \quad . \tag{32}$$

The wavelength λ in the lossy dielectric is then computed from

$$\lambda = \frac{2\pi}{\beta} \quad . \tag{33}$$

The intrinsic wave impedance η is given by

$$\eta = \sqrt{\frac{j\omega\mu}{\sigma + j\omega\epsilon}} = \sqrt{\frac{\mu}{\epsilon}}\frac{1}{\sqrt{1 - j\frac{\sigma}{\omega\epsilon}}} \quad . \tag{34}$$

The instantaneous power density of the electromagnetic field is given by Poynting's vector, denoted $\boldsymbol{P}$,

$$\boldsymbol{P} = \frac{1}{2}\boldsymbol{E}\times\boldsymbol{H}^* \quad , \tag{35}$$

which has units of (W/m^2). The time-average power flow density is equal to the real part of the complex Poynting's vector. The time-average power dissipation per unit volume P_d (W/m^3) is derived from Maxwell's equations, with the result

$$P_d = \frac{1}{2}\boldsymbol{E} \cdot \boldsymbol{J}^* = \frac{1}{2}\sigma|\boldsymbol{E}|^2 \quad . \tag{36}$$

The specific absorption rate (SAR) [2] is the power dissipated or absorbed per unit mass (W/kg) of the medium (tissue), or

$$SAR = \frac{P_d}{\rho} = \frac{\sigma}{2\rho}|\boldsymbol{E}|^2 \quad , \tag{37}$$

where ρ is the density of the medium in kg/m^3.

It is convenient to have a simple equation for computing the propagation loss between any two points in the near field of an isolated transmitting antenna. Thus, mutual coupling effects are ignored for the time being. Consider a time-harmonic source radiating a spherical wave into an infinite homogeneous conducting medium. For an isotropic radiator, and suppressing the $e^{j\omega t}$ time dependence, the electric field as a function of range r can be expressed as

$$E(r) = E_o\frac{e^{-\gamma r}}{r} \quad , \tag{38}$$

where E_o is a constant.

For a source at the origin, the amplitude of the electric field at range r_1 is given by

$$|E(r_1)| = E_o\frac{e^{-\alpha r_1}}{r_1} \quad , \tag{39}$$

and at range r_2 by

$$|E(r_2)| = E_o\frac{e^{-\alpha r_2}}{r_2} \quad . \tag{40}$$

The total propagation loss between ranges r_1 and r_2 is found by taking the ratio of Equations (40) and (39), or

$$\frac{|E(r_2)|}{|E(r_1)|} = \frac{r_1}{r_2}e^{-\alpha(r_2-r_1)} \quad . \tag{41}$$

The field attenuation A_α in dB from range r_1 to range r_2 due to the lossy dielectric is simply

$$A_\alpha = 20 \log_{10}(e^{-\alpha(r_2 - r_1)}) \quad . \tag{42}$$

Similarly, the $1/r$ attenuation loss A_r in dB is

$$A_r = 20 \log_{10} \frac{r_1}{r_2} \quad . \tag{43}$$

2.3.2 THERMAL MODELING OF AN INHOMOGENEOUS TARGET

A thermal analysis computer program called the transient thermal analyzer (TTA), developed by Arthur D. Little, Inc., has been used at Lincoln Laboratory for space payload analysis since 1969. Recent documentation for TTA exists [51,52]. The important details of the analysis are described in this section. The TTA software is used in the next section to accomplish the thermal modeling of homogeneous muscle tissue surrounded by a constant-temperature water bolus. Thermal modeling techniques have been described in detail [53].

The TTA program uses the finite-difference technique to solve a set of nonlinear energy balance equations. Consider a system of interconnected nodes that model an inhomogeneous volume for which the temperature T_i of the ith node is to be determined. The heat-balance equation, which is solved by TTA, is expressed as

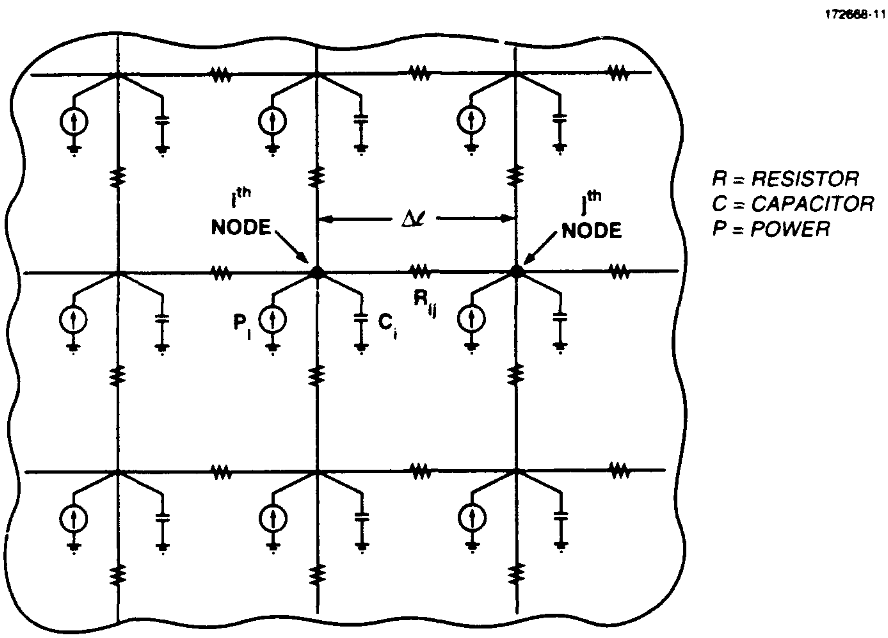
$$\sum_{i=1}^{N} Q_{i,j} - P_i(t) + M_i \frac{dT_i}{dt} = 0 \quad , \tag{44}$$

where $Q_{i,j}$ is the net outward heat flow from node i in the direction of node j, $P_i(t)$ is the power into node i at time t, and M_i is the thermal mass (mass times specific heat) of node i. Figure 11 shows an electric circuit analog which is used to model the two-dimensional thermal characteristics of the material volume. Power P_i in watts is delivered to the ith node. Capacitor C_i (with units Joules/°C) is used to model the thermal capacitance at the ith node. Resistor $R_{i,j}$ (with units °C/W) is used to model the heat resistance between nodes i and j.

With a spacing of Δl between nodes (assuming cubic cells), the values of $R_{i,j}$, C_i and P_i are computed as

$$R_{i,j} = \frac{1}{k_{i,j} \Delta l} \quad , \tag{45}$$

where $k_{i,j}$ is the thermal conductivity (with units W/m°C) between nodes i and j;



$R = \frac{1}{k\Delta\ell}$ [°C / W], k = THERMAL CONDUCTIVITY

$C = \rho C_P (\Delta\ell)^3$ [J/°C], ρ = DENSITY, C_P = SPECIFIC HEAT

$P = SAR\ \rho\ (\Delta\ell)^3$ [W], $SAR = \frac{\sigma}{2\rho} |E|^2$ (Specific Absorption Rate)

σ = ELECTRICAL CONDUCTIVITY

$|E|$ = MAGNITUDE OF ELECTRIC FIELD

Figure 11. Equivalent circuit model used in thermal analysis. The inhomogeneous body of interest is modeled with a grid of nodes interconnected by resistors. Each node has an associated power source and capacitor.

$$C_i = \rho_i C_{pi} (\Delta l)^3 \quad , \tag{46}$$

where C_{pi} is the specific heat at the ith node and ρ_i is the density (kg/m^3) at the ith node; and

$$P_i = (SAR)_i \rho_i (\Delta l)^3 \quad , \tag{47}$$

where $(SAR)_i$ is the SAR for the ith node, which is given by

$$(SAR)_i = \frac{\sigma_i}{2\rho_i} |E_i|^2 \quad , \tag{48}$$

where σ_i is the electrical conductivity of the ith node and $|E_i|$ is the magnitude of the electric field delivered by the hyperthermia array to the ith node. Note: in substituting Equation (48) into Equation (47), the density ρ_i cancels. Thus, an equivalent approach to computing the power delivered to the ith node is written in terms of the time-average power dissipated per unit volume of the ith node (denoted P_{di}) as

$$P_i = P_{di} (\Delta l)^3 \quad . \tag{49}$$

A block diagram showing how TTA is used in the hyperthermia analysis is given in Figure 12. First, the method of moments, controlled by the SMI nulling algorithm, is used to compute the electric field radiation pattern throughout a homogeneous region (muscle tissue). This power distribution is then read into the TTA program, which computes the temperature distribution inside an elliptical muscle-tissue target surrounded with a constant-temperature water bolus.

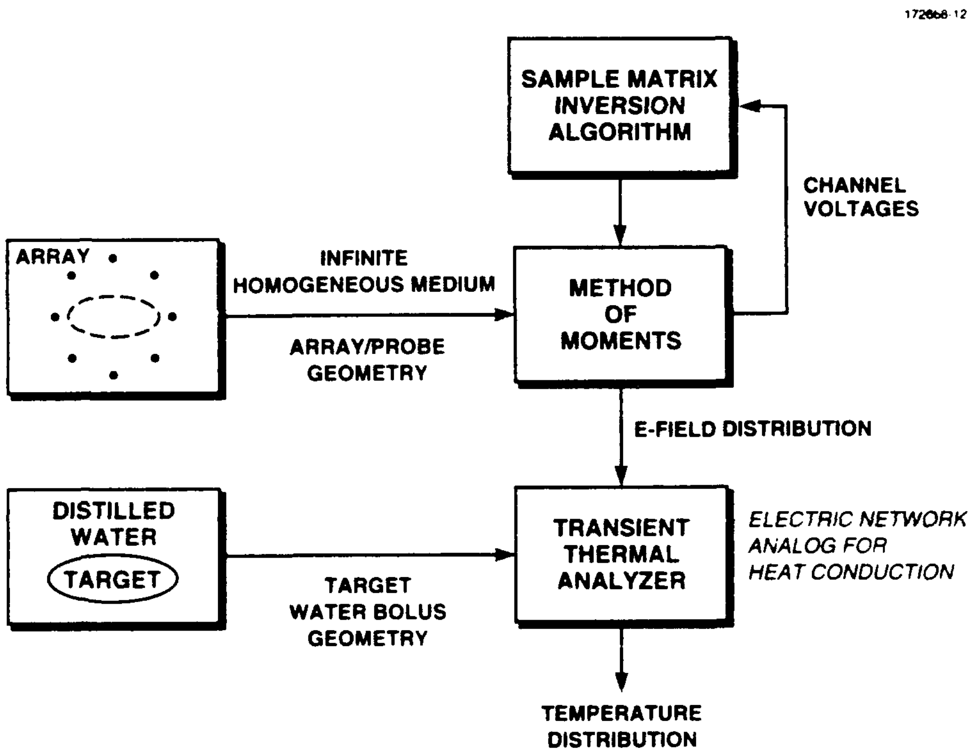


Figure 12. The moment-method simulation's use with the transient thermal analysis. RF power distribution over the target region is converted into a time-dependent temperature distribution.

3. SIMULATION RESULTS

To demonstrate the effectiveness of focused near-field adaptive nulling in reducing undesired hot spots, several computer simulations are presented. The E-field simulations are for the signal received by a short-dipole probe due to a transmitting phased array embedded in an infinite homogeneous lossy dielectric (muscle tissue). However, the thermal simulations are for an elliptical target (muscle tissue) surrounded by a constant-temperature water bolus. Because the RF wavelengths in the target and water bolus are similar, the E-field simulations are believed to give a reasonable approximation to the field distribution inside the elliptical target.

The E-field calculation in the assumed infinite homogeneous medium introduces additional field attenuation not present in a clinical hyperthermia system with an annular array transmitting through a water bolus into a patient. (As mentioned earlier, the water bolus has very little RF propagation loss.) In addition, the transmit array weights are normalized according to Equation (6). Thus, in this report no attempt is made to compute the absolute E-field strength in volts/meter in the elliptical target. Instead, the peak power in the elliptical target is adjusted (by a scale factor) to produce a desired maximum focal-region temperature (T_{max}) after t minutes. Note: an approximate absolute scale factor could be computed by making an initial computer simulation with an infinite homogeneous water bolus and then matching the target boundary field to the infinite homogeneous muscle tissue simulation.

The computer simulation model is related, in part, to the hyperthermia annular phased-array antenna system shown in Figure 1. This hyperthermia system uses a 60-cm array diameter with eight uniformly spaced dipole elements (fed as four pairs) which operate over the frequency band 60–120 MHz [21]. The simulated array is assumed to be fully adaptive, and with eight adaptive elements seven independent nulls can be formed while simultaneously focusing on a tumor. For comparison, the BSD-2000 hyperthermia system could be used to form up to three independent nulls.

In this section, we assume that the adaptive radiation pattern null-width characteristics in a homogeneous target will be similar to the characteristics observed in an inhomogeneous target. The null width is directly related to the wavelength and there is only a 5 percent change in wavelength between the assumed muscle tissue and water bolus. With this assumption, the transmit array is embedded in homogeneous tissue, which allows direct use of the thin-wire moment-method formulation in Section 2.3. After computing the two-dimensional electric-field distribution in the homogeneous medium, we then consider only an elliptical portion of the homogeneous region and use the ellipse as the homogeneous target. In the thermal analysis, the elliptical target is surrounded with a constant 10°C water bolus. The electric field amplitude is scaled to produce a 46°C peak temperature, at time $t = 20$ minutes, at the center of the elliptical phantom. The initial temperature of the phantom is assumed to be 25°C (room temperature).

All computer simulations presented in this section are at a 120-MHz operating frequency with four auxiliary nulling sensors; that is, $N_{aux} = 4$. Degraded system performance due to an insufficient number of auxiliary probes is considered in Appendix A. The parameters used in the electrical and thermal analyses are summarized in Table 1. These parameters, obtained in part

TABLE 1

Parameters Used in Electrical/Thermal Analysis

171588-5

PARAMETER	PHANTOM MUSCLE TISSUE	DISTILLED WATER
DIELECTRIC CONSTANT @ 100 MHz	73.5	80.0
ELECTRICAL CONDUCTIVITY @ 100 MHz	0.5 S/m	0.0001 S/m
DENSITY	970.0 kg/m^3	1000.0 kg/m^3
SPECIFIC HEAT	3516.0 J/kg °C	4200.0 J/kg °C
THERMAL CONDUCTIVITY	0.544 W/m °C	0.6019 W/m °C

from Sullivan [9], are for a frequency of 100 MHz: we assume that similar values of the parameters will exist at 120 MHz. Notice that the relative dielectric constants of phantom muscle tissue and distilled water are very similar; however, the electrical conductivities are vastly different. The relevant thermal characteristics—density, specific heat, and thermal conductivity [35]—are very similar for phantom muscle tissue and distilled water.

3.1 ELECTRIC FIELD FOR ARRAY IN HOMOGENEOUS TISSUE

Substituting the values f = 120 MHz, σ = 0.5 S/m, and ϵ_r = 73.5 into Equation (29) yields $\gamma_m = 10.0 + j23.8$ for the muscle tissue. With β_m = 23.8 radians/m, the wavelength in the phantom muscle tissue is $\lambda_m = 26.5$ cm. The attenuation constant for the muscle tissue is $\alpha_m = 10.0$ radians/m. Similarly, for distilled water $\gamma_w = 0.0021 + j22.5$, so the wavelength is $\lambda_w = 27.9$ cm. The attenuation constant for the distilled water medium is $\alpha_w = 0.0021$ radians/m. The propagation loss in the phantom muscle tissue is $20\log_{10} e^{-10.0}$, or -0.87 dB/cm. Similarly, the propagation loss in the distilled water is found to be -0.0002 dB/cm. Thus, the total loss due to propagation through 15 cm of distilled water is 0.003 dB. For 15 cm of muscle tissue the corresponding loss is 13.1 dB. The wave impedance in the muscle tissue is computed from Equation (34) as $\eta_m = 33.9 + \mathrm{j}\ 14.2\ \Omega$, and similarly in the distilled water $\eta_w = 42.1 + \mathrm{j}\ 0.004\ \Omega$.

The geometry used in the simulations is shown in Figure 13. A 60-cm-diameter ring array of eight dipoles uniformly surrounds a fictitious elliptical target zone with major axis 30 cm and minor axis 20 cm. The length of each perfectly conducting center-fed dipole array element at 120 MHz in the infinite homogeneous muscle tissue is $\lambda/2$, or 13.25 cm. The array focus is assumed at the origin, and four auxiliary short-dipole sensors with length 1.27 cm (0.05λ) are positioned in (x, y, z) coordinates at (15 cm, 0, 0), (−15 cm, 0, 0), (0, 0, 10 cm), and (0, 0, −10 cm); that is, the auxiliary E-field sensors are located every 90° in azimuth on the perimeter of the target. In rectangular coordinates, each dipole is oriented in the $\hat{y}$ direction and the feed terminals of each dipole are located at $y = 0$. The moment-method computer simulations were run on a Sun 3/260 workstation. The total CPU time for a complete moment-method run is 19.2 minutes. This CPU time includes computing the quiescent and adaptive radiation patterns on a 41 by 41 grid of points. The CPU time without radiation pattern calculations is 33 seconds.

The two-dimensional radiation pattern in the plane $y = 0$, before nulling, at 120 MHz with uniform amplitude and phase illumination, is shown in Figure 14. The calculated data are collected on a 41 by 41 grid of points over a square region, with side length 76.2 cm, centered at the focus. The spacing between data points is 1.905 cm, or 0.072λ, and the contour levels are displayed in 10-dB steps. The E-field data are computed for the case of a 1.27-cm short-dipole observation probe. The positions of the eight dipole radiators are clearly evident by the −20-dB contours surrounding each element. The radiation pattern is symmetric because of the symmetry of the array and the assumed homogeneous medium.

Finer contour levels (1-dB steps) for the quiescent radiation pattern are displayed in Figure 15. Here, it is evident that the focused main beam of the ring array is increasing in amplitude as the observation point moves closer to the focus. Away from the main beam region, the pattern amplitude is seen to increase as the observation position moves away from the focus.

A quiescent radiation pattern cut at $z = 0$ is shown in Figure 16. The large amplitude that occurs at ±30 cm is due to the E-field probe's close proximity to the transmitting elements. The large attenuation that occurs from the array diameter to the focus is due to the $1/r$ attenuation loss and the loss in the uniform homogeneous muscle tissue. The radiation pattern cut for $x = 0$ in Figure 17 is identical to the pattern in Figure 16 due to the symmetry of the array. In both Figures 16 and 17 the boundary of the fictitious elliptical target zone is indicated. The increasing radiation pattern amplitude near the left and right sides of the elliptical target (Figure 16) will be

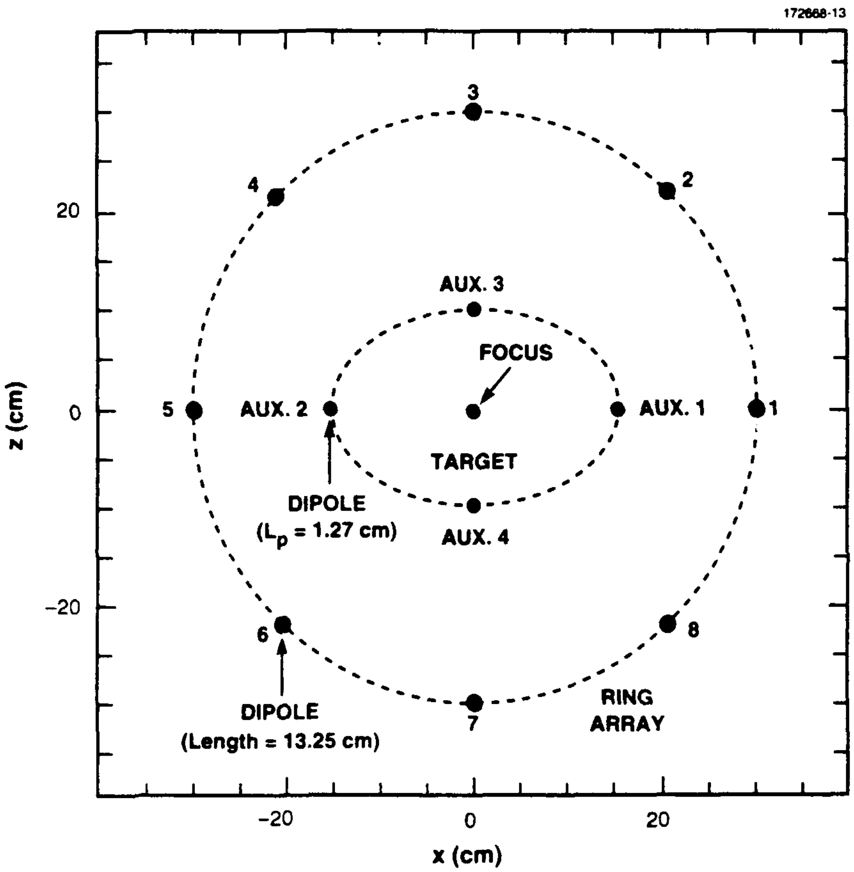


Figure 13. Geometry for eight-element ring array and four E-field auxiliary sensors.

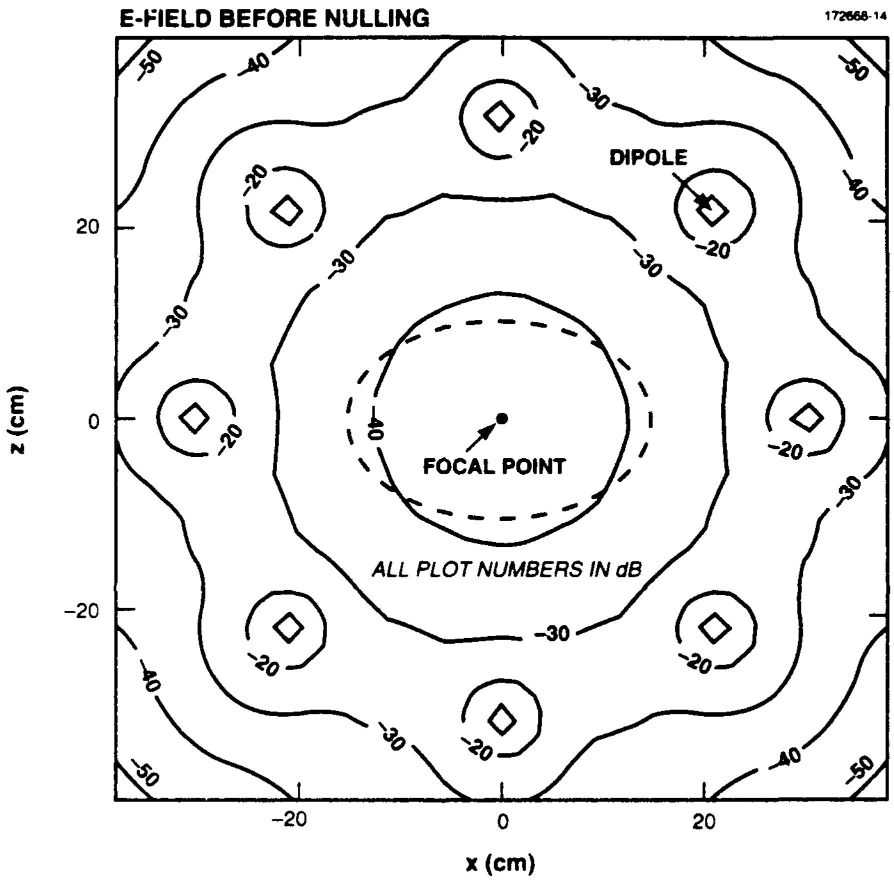


Figure 14. Simulated two-dimensional quiescent radiation pattern at 120 MHz for uniformly illuminated eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$). A fictitious elliptical target region is indicated by the dashed curve. Radiation contour levels are in 10-dB steps.

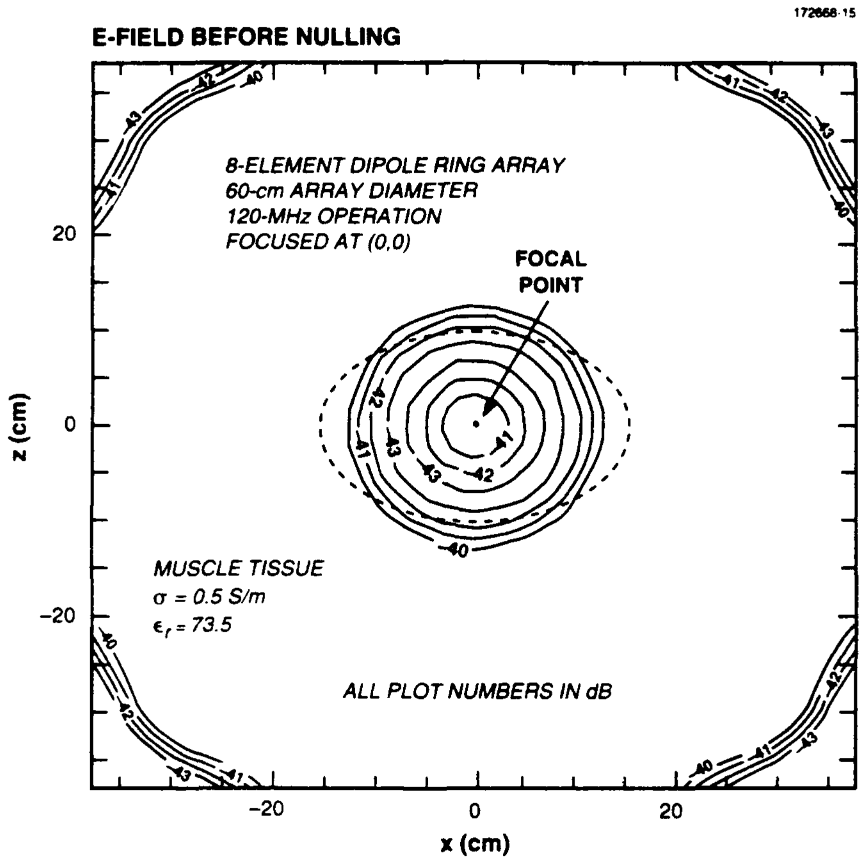


Figure 15. Simulated two-dimensional quiescent radiation pattern at 120 MHz for uniformly illuminated eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$). A fictitious elliptical target region is indicated by the dashed curve. Radiation contour levels are given in 1-dB steps.

shown to produce hot spots in the thermal distribution. As shown in the next section, because the top (anterior) and bottom (posterior) of the elliptical target are not as strongly illuminated as the left and right sides, no quiescent hot spots occur at the top or bottom. From Figure 16, the

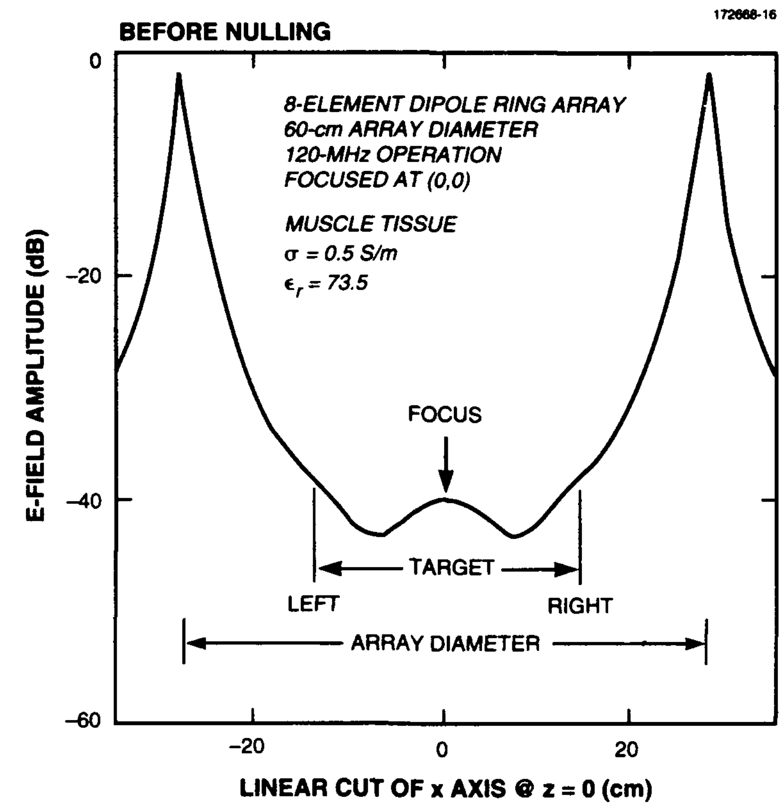


Figure 16. Simulated one-dimensional quiescent radiation pattern cut ($z = 0$) at 120 MHz for uniformly illuminated eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$).

ring-array half-power beamwidth in the target region is approximately 13 cm, or approximately one-half the wavelength (26.5 cm) in the phantom muscle tissue. [2] The adaptive nulling resolution

[2]Sullivan [54] has recently published simulations and measurements of SAR patterns at 70–110 MHz. using a homogeneous 35 × 25-cm elliptical target surrounded with a layer of fat and a water

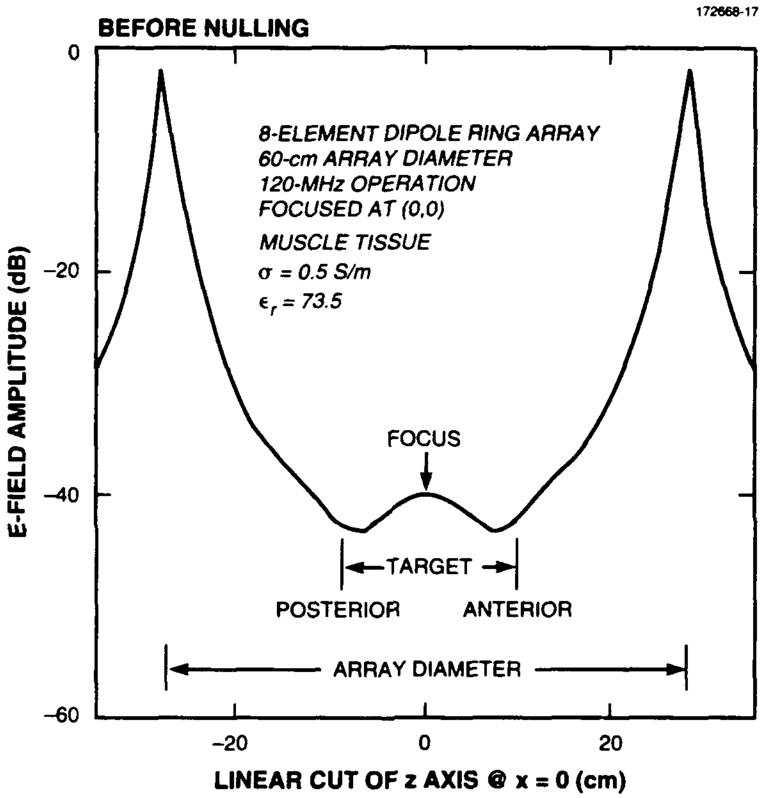


Figure 17. Simulated one-dimensional quiescent radiation pattern cut ($x = 0$) at 120 MHz for uniformly illuminated eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$).

or closest allowed spacing between a deep adaptive null and the main beam is equal to the half-power beamwidth of the antenna [55]. Thus, the closest allowed null position is 13 cm from the focus. Since the target width is 30 cm, two nulls can be formed at ($x = \pm 15$ cm, $z = 0$) without disturbing the focus. However, if two deep nulls are formed at ($x = 0, z = \pm 10$ cm) the focus will be compromised. In practice, the water bolus would restrict the placement of short-dipole sensors to the surface of the target. Thus, only weak nulls can be formed at ($x = 0, z = \pm 10$ cm) so that the focus will not be affected by the adaptive nulling process.

Next, adaptive radiation patterns are computed with four auxiliary dipole sensors; their positions are shown in Figure 13. The value of the receiving gain for auxiliary dipoles 1 and 2 is adjusted to produce a greater-than-35-dB SNR. This amount of SNR results in greater than 35 dB of nulling in the direction of auxiliary dipoles 1 and 2. In contrast, the gain values for auxiliary dipoles 3 and 4 are turned down to produce about a 3-dB SNR. Thus, only about 3 dB of nulling will occur at sensor positions 3 and 4 as the adaptive algorithm reduces the interference to the noise level of the receiver. The reason for choosing these null depths will become apparent with the data that follow. Figure 18 shows the two-dimensional radiation pattern after nulling with four auxiliary sensors. Two deep adaptive nulls at $x = \pm 15$ cm occur as expected, and weak nulling occurs at $z = \pm 10$ cm, also as expected. The two deep nulls in the $z = 0$ cut are quantified in Figure 19, where greater than 35 dB of interference nulling or pattern reduction occurs at $x = \pm 15$ cm. The peak level at the focus is adjusted to 0 dB for both the quiescent and adaptive patterns. Two weak adaptive nulls are in effect in the $x = 0$ radiation pattern cut shown in Figure 20; however, weak nulls are desired in this cut due to temperature considerations, as shown in the next section. The weak nulls in effect in the adaptive patterns reduce variation from the quiescent radiation pattern. Finally, the transmit array weights before and after nulling and the covariance matrix eigenvalues are shown in Figures 21 and 22, respectively. The adaptive transmit weights exhibit a 5-dB dynamic range in Figure 21(a). There are two large eigenvalues and two weak (nonzero) eigenvalues shown in Figure 22. These eigenvalues are directly associated with the two high-SNR auxiliary sensors and the two weak-SNR auxiliary sensors. Note that the 0-dB level in Figure 22 is equal to the noise level. The probe-array output power before and after adaptive nulling is 31.4 dB and 0.9 dB, respectively. This difference in power before and after nulling means that the adaptive cancellation is −30.5 dB. The output file for the moment-method simulation of this section is given in Appendix B.

---

bolus. A comparison of Sullivan's 110-MHz SAR simulation (Figure 5(a) of his paper) and the results (at 120 MHz) presented in Figures 16 and 17 of this report indicate reasonable agreement over the elliptical target region.

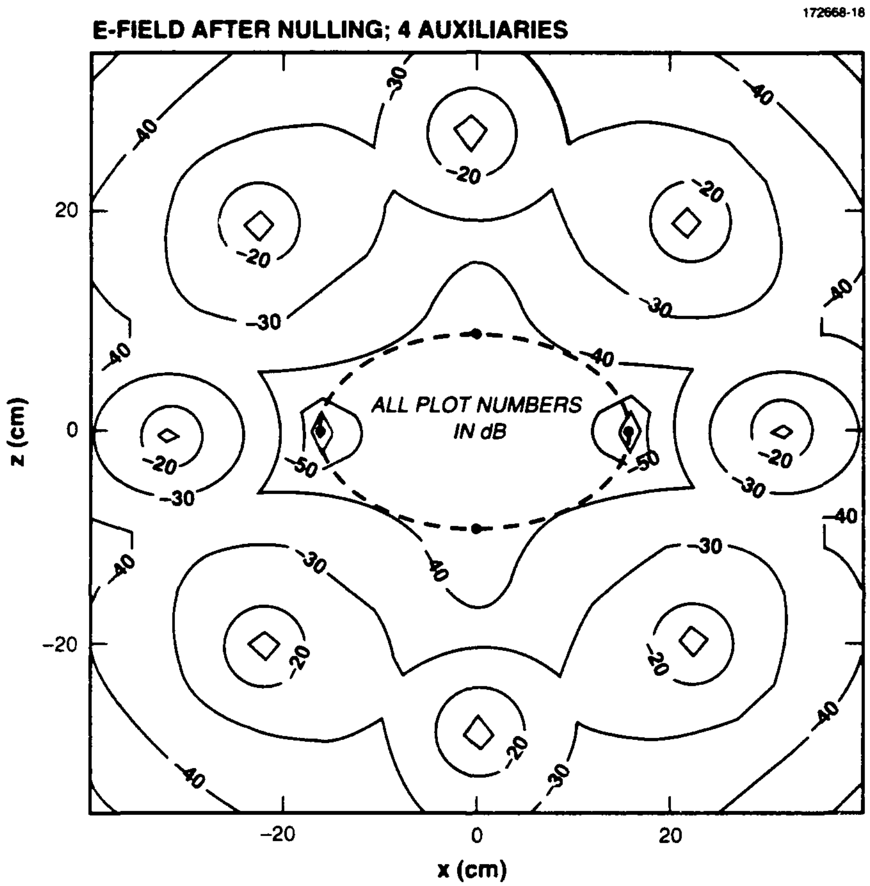


Figure 18. Simulated two-dimensional adaptive radiation pattern at 120 MHz for eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$). Radiation contour levels are given in 10-dB steps. Four auxiliary sensors are used in forming the adaptive pattern. The quiescent focus is at (0,0).

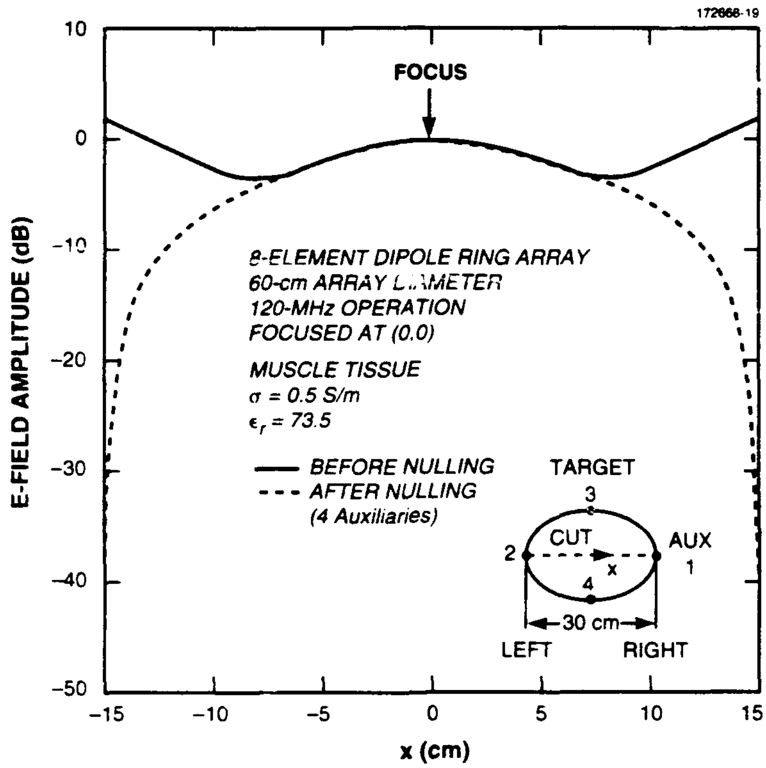


Figure 19. Simulated one-dimensional quiescent and adaptive radiation patterns in the $z = 0$ cut at 120 MHz for eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$). Four auxiliary sensors are used in forming the adaptive pattern.

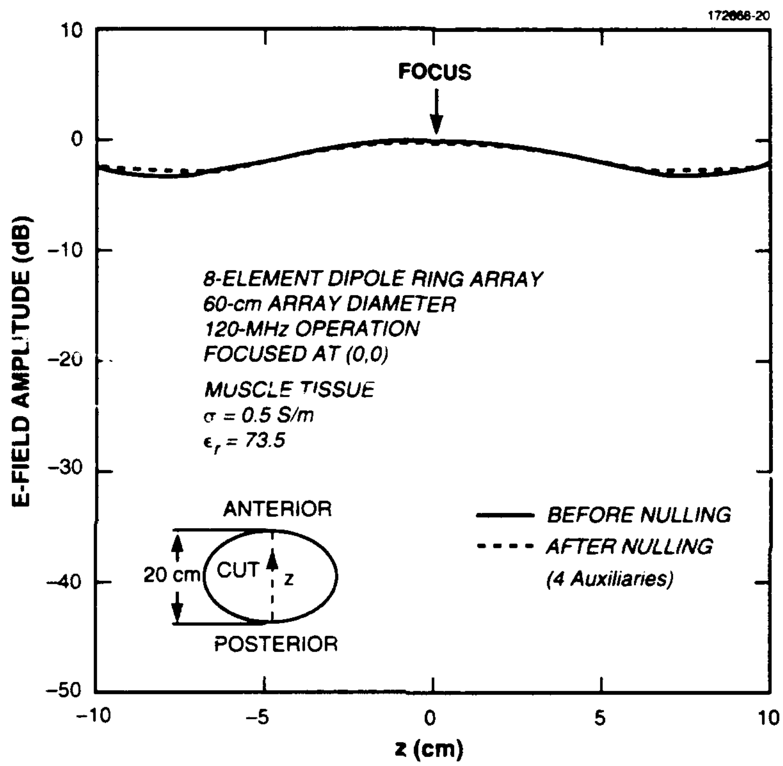


Figure 20. Simulated one-dimensional quiescent and adaptive radiation patterns in the $x = 0$ cut at 120 MHz for eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$). Four auxiliary sensors are used in forming the adaptive pattern.

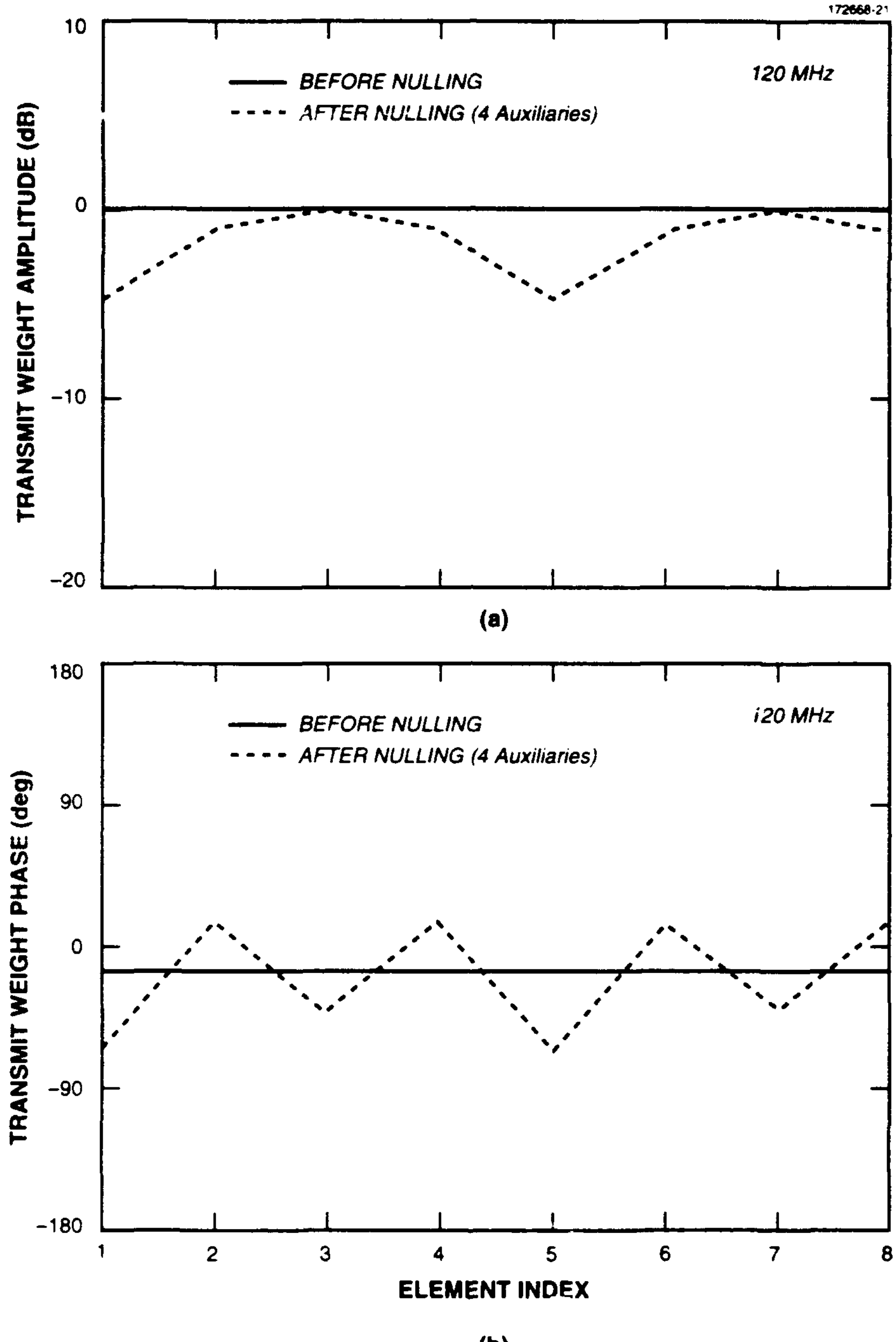


Figure 21. Transmit array weights before and after adaptive nulling. A dynamic range of about 5 dB is evident for the adaptive weights. Four auxiliary sensors are used in the adaptive process. (a) Amplitude: (b) Phase.

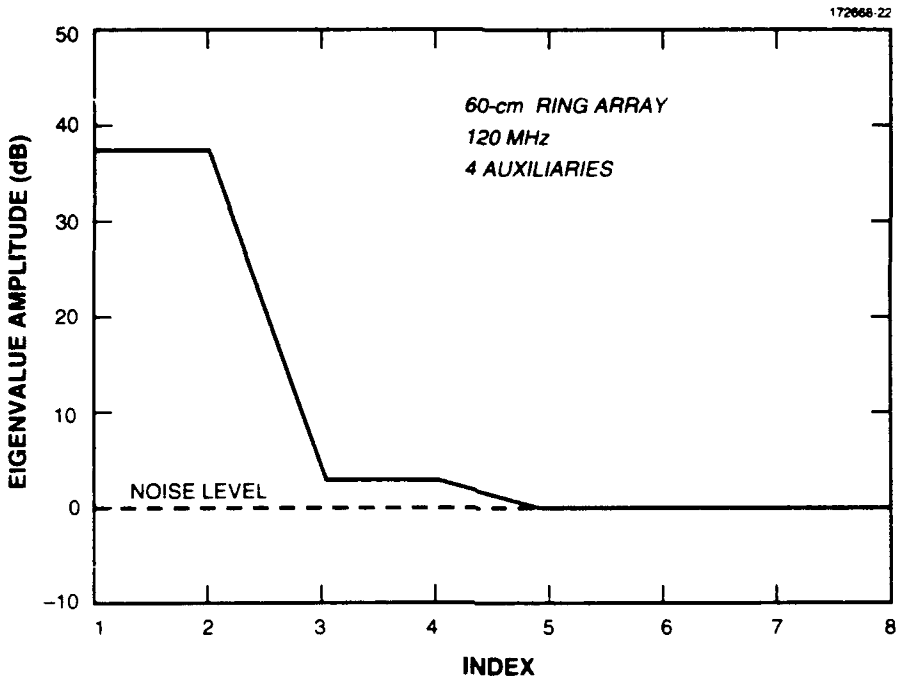


Figure 22. Channel covariance matrix eigenvalues (degrees of freedom) used in the adaptive process with four auxiliary sensors.

3.2 TEMPERATURE DISTRIBUTION IN ELLIPTICAL PHANTOM

In this section, the transient thermal analysis software is used to compute the temperature distribution in an elliptical phantom surrounded with a constant-temperature water bolus. The 41 by 41 two-dimensional E-field radiation pattern data from the previous section are used as the power source for the thermal node network. Two spacings of nodes are considered here: initially, the node spacing $\Delta x = \Delta z = \Delta l = 1.905$ cm (coarse grid) is used; to check convergence, the node spacing is then decreased by a factor of two to $\Delta l = 0.9525$ cm (fine grid). The coarser spacing is shown to be adequate.

3.2.1 1.905-cm Thermal Node Spacing

Note: the scale factors used to convert the normalized E-field distributions to a power level that induces a 46°C peak temperature at $t = 20$ minutes are 94.1 dB and 96.0 dB for the quiescent and adaptive patterns, respectively. These scale factors are determined by trial and error. From equations given in Section 2.3.2 and the parameter values in Table 1, all resistors in the phantom muscle tissue had a value of 96.5°C/W and all resistors in the water bolus had a value of 87.2°C/W. The value of the capacitors in the phantom muscle tissue is 23.6 J/°C; values for capacitors in the water-bolus region are not used in the input to the transient thermal analysis software. Instead, a constant temperature of 10°C is enforced at each water-bolus node. With a 41 by 41 grid, a total of 3280 resistors and 1681 capacitors are used in the thermal simulation. The CPU time required to compute the temperature distribution on a 41 by 41 grid is under four minutes.

Figure 23 shows the two-dimensional temperature distribution produced at time $t = 20$ minutes in the elliptical phantom muscle tissue target without adaptive nulling. To generate Figure 23, the power source used in the transient thermal analysis is the quiescent radiation pattern given in Figure 14. The initial temperature (at time $t = 0$) is 25°C. Notice the occurrence of two hot spots on the left and right sides of the elliptical phantom. The peak temperature on-focus is 46°C, which is achieved by scaling the normalized quiescent E-field as described earlier. The two hot spots are quantified in the $z = 0$ temperature pattern cut shown in Figure 24. While the peak temperature at each hot spot in Figure 24 is only 41°C, the temperature profile for $x = 0$ in Figure 25 shows no hot spots. As any undesired hot spot is a potential source for compromising the therapy session, adaptive nulling is used to reduce the sidelobes corresponding to the hot spots.

Figure 26 shows the simulated two-dimensional thermal distribution at time $t = 20$ minutes, with adaptive nulling at four auxiliary sensors in effect. The focal-spot diameter with adaptive nulling is equivalent to the focal-spot diameter before adaptive nulling. Hot spots on the left and right sides of the target are eliminated. A comparison of the temperature distribution before and after nulling along the major axis ($z = 0$) of the target ellipse is made in Figure 27. Similarly, the temperature distribution before and after nulling along the minor axis of the target ellipse is shown in Figure 28.

3.2.2 Convergence Check: 0.9525-cm Thermal Node Spacing

The convergence of the previous thermal simulations was verified by increasing the density of E-field observation probe positions by a factor of two, with a new spacing between points of 0.9525 cm, still with a 41 by 41 grid. The ring array operates as before at 120 MHz, and there are four auxiliary sensors laid out as shown in Figure 13. As the auxiliary positions are the same, the adaptive weights and covariance matrix eigenvalues in Figures 21 and 22, respectively, remain the same. From the parameter values in Table 1, all resistors in the finer-grid muscle-tissue phantom had a value of 193.0°C/W and all resistors in the water bolus had a value of 174.4°C/W. The value of the capacitors in the phantom muscle-tissue is 2.95 J/°C. Again, a constant temperature of 10°C is enforced at each water-bolus node. The E-field scaling factors to raise the focal-point temperature to 46°C before and after nulling are 76.5 dB and 78.4 dB, respectively. The finer-grid two-dimensional thermal distributions before and after nulling are shown in Figures 29 and 30, respectively. Although the temperature contours are smoother, the general agreement between these patterns and the coarser-grid patterns in Figures 23 and 26 are evident. Similarly, one-dimensional thermal pattern cuts with the finer grid are shown in Figures 31 (x axis) and 32 (z axis); good agreement with the coarse-grid patterns (Figures 27 and 28) is observed. In particular, the finer detail in Figure 31 shows that the hot spots are at a 42°C level compared to the 41°C level observed for the coarse grid (Figure 24). Thus, convergence of the coarse-grid thermal patterns is demonstrated.

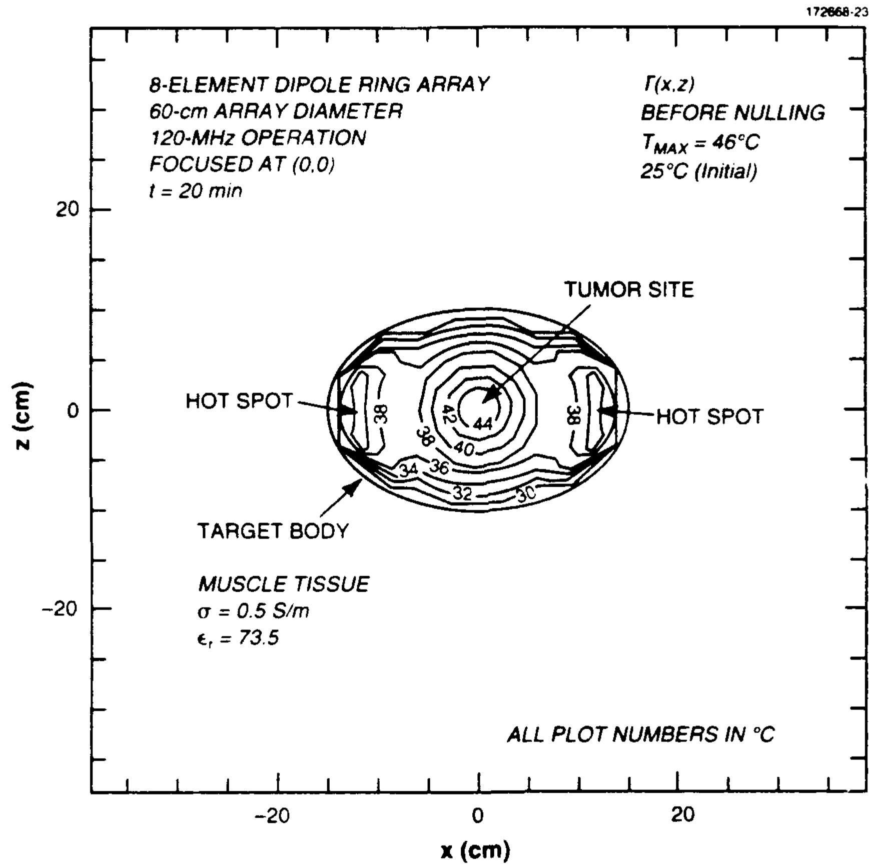


Figure 23. Simulated two-dimensional thermal pattern at time t = 20 minutes before nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The incident RF power distribution, from Figure 14, is at 120 MHz. Temperature contour levels are given in 2° C steps. Hot spots on the left and right sides of the target are observed.

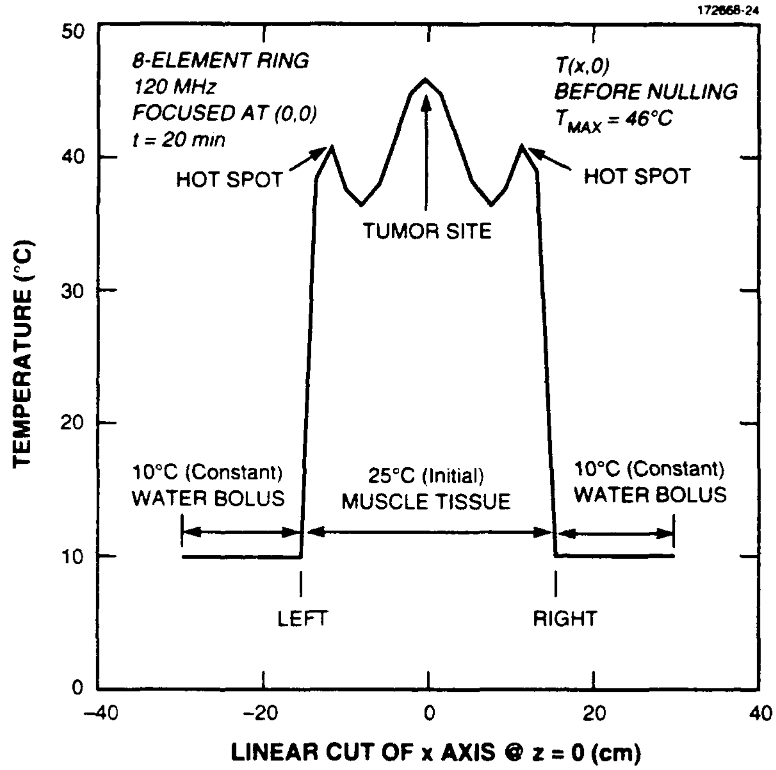


Figure 24. Simulated one-dimensional ($z = 0$) thermal pattern at time $t = 20$ minutes before nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution, from Figure 14, is at 120 MHz. Hot spots on the left and right sides of the target are observed.

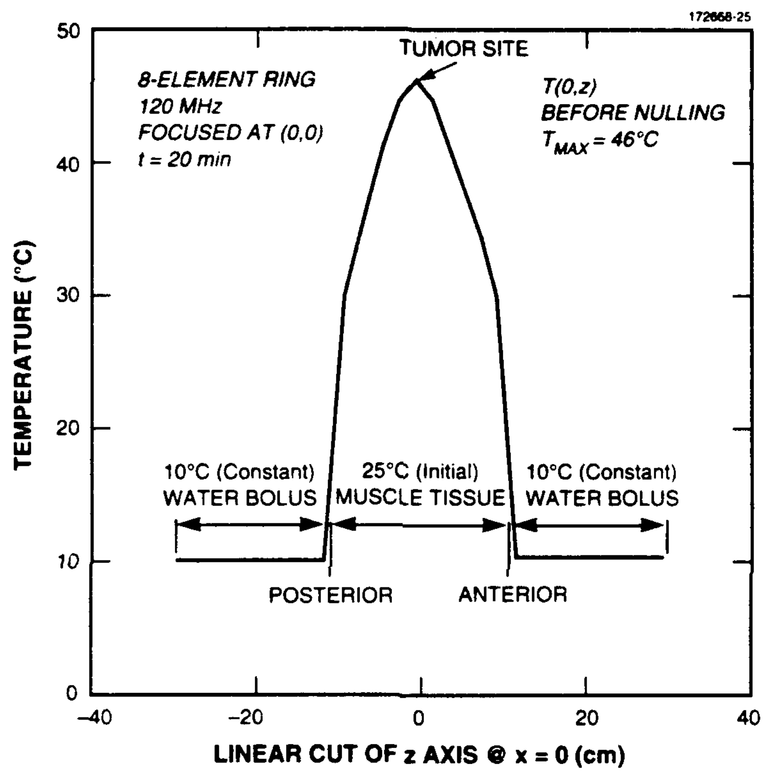


Figure 25. Simulated one-dimensional ($x = 0$) thermal pattern at time $t = 20$ minutes before nulling in elliptical phantom muscle-tissue target surrounded with 10°C constant-temperature water bolus. The quiescent incident RF power distribution, from Figure 14, is at 120 MHz. No undesired hot spots are present.

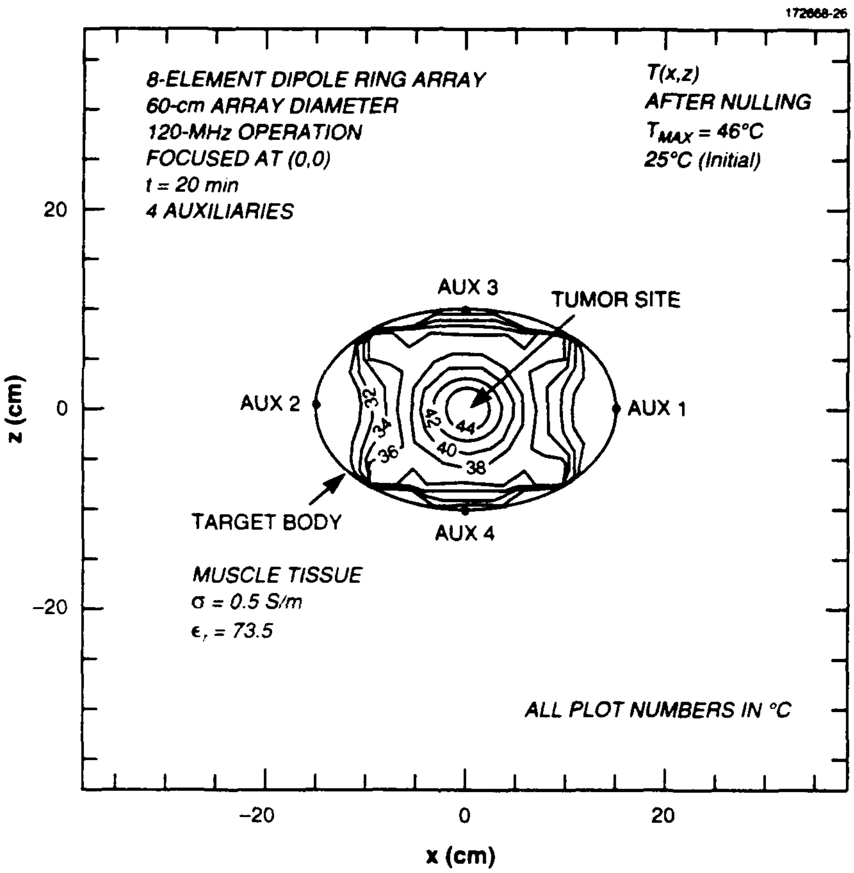


Figure 26. Simulated two-dimensional thermal pattern at time $t = 20$ minutes (with adaptive nulling at four auxiliary sensors in effect) in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The adapted incident RF power distribution, from Figure 18, is at 120 MHz. Temperature contour levels are given in 2° C steps. Hot spots on the left and right sides of the target are eliminated.

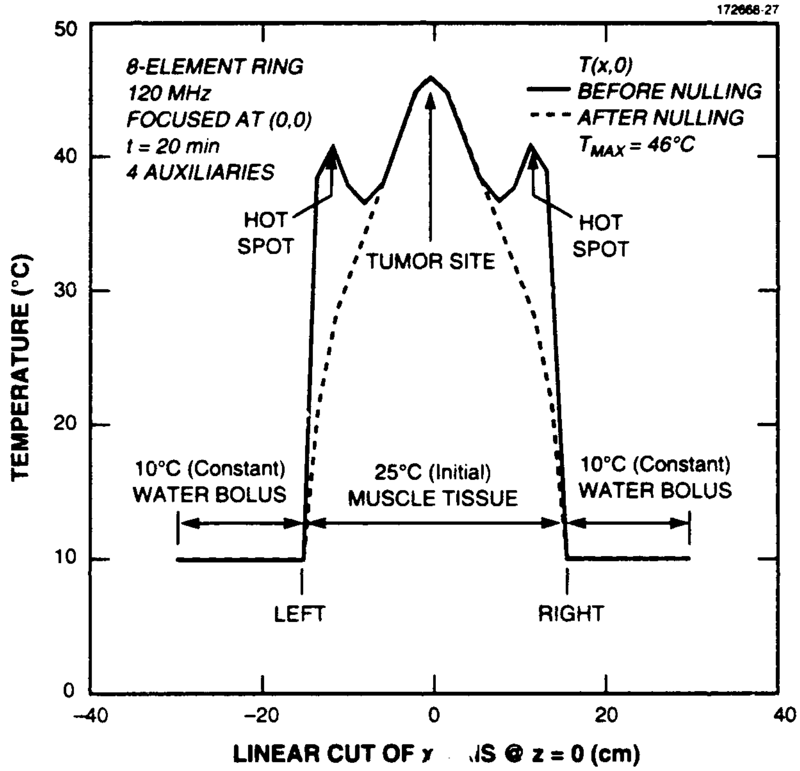


Figure 27. Simulated one-dimensional ($z = 0$) thermal patterns at time $t = 20$ minutes before and after nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution, from Figure 14, is at 120 MHz. Hot spots on the left and right sides of the target are clearly eliminated by the adaptive nulling process. Four auxiliary sensors are used in the adaptive process.

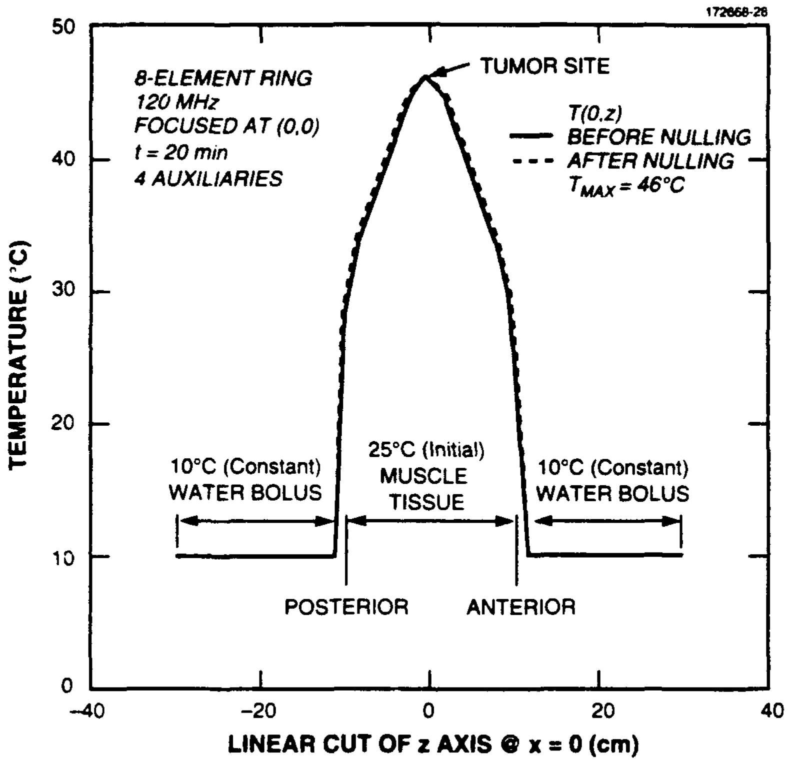


Figure 28. Simulated one-dimensional ($x = 0$) thermal patterns at time $t = 20$ minutes before and after nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution, from Figure 14, is at 120 MHz. No undesired hot spots are present. Four auxiliary sensors are used in the adaptive process.

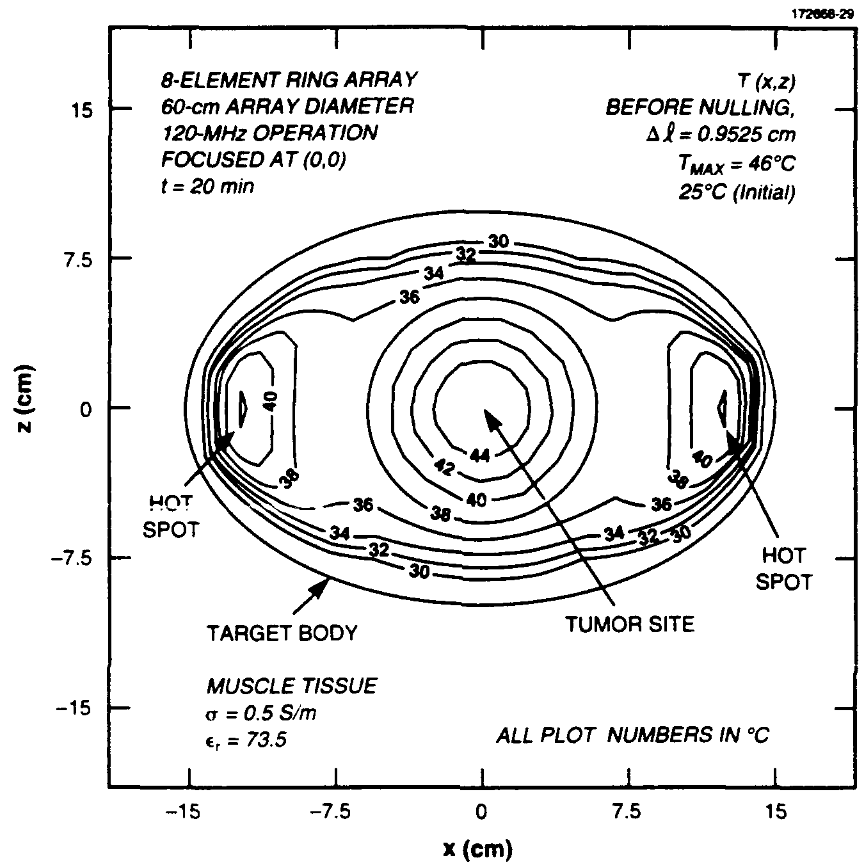


Figure 29. E-field probe-sample-spacing convergence check for simulated two-dimensional thermal pattern at time t = 20 minutes (before adaptive nulling) in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution is at 120 MHz. Temperature contour levels are given in 2° C steps. The grid spacing is one-half the spacing of that in Figure 23. Hot spots on the left and right sides of the target are present, as previously observed for the coarser probe-sample spacing.

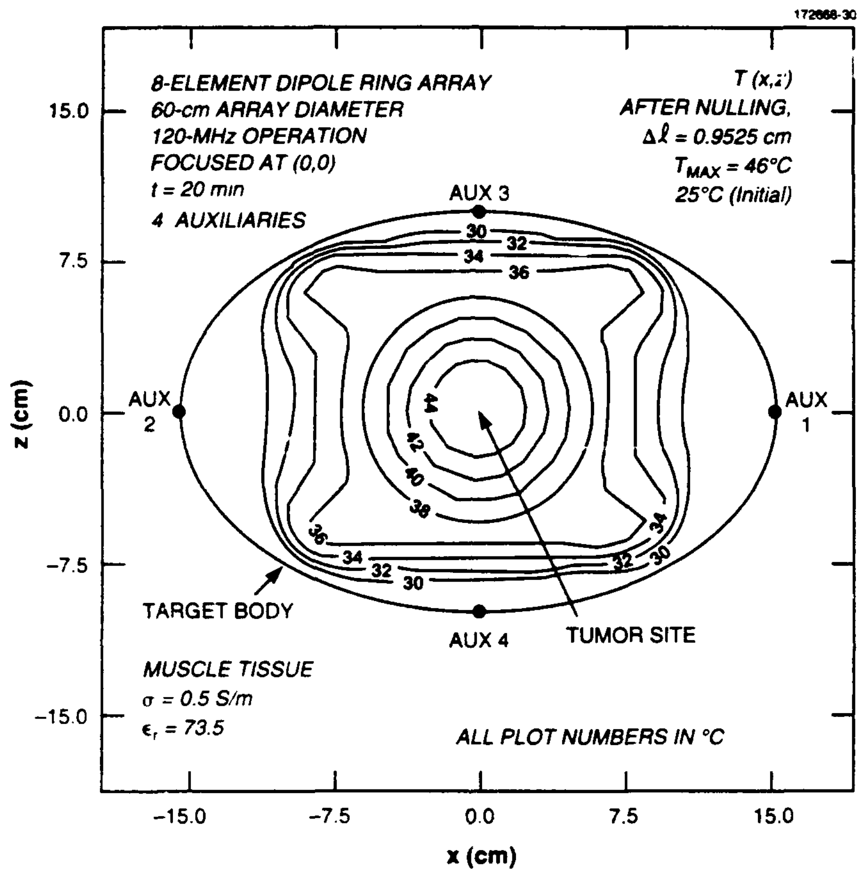


Figure 30. E-field probe-sample-spacing convergence check for simulated two-dimensional thermal pattern at time t = 20 minutes (with adaptive nulling in effect) in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The adapted incident RF power distribution is at 120 MHz. Temperature contour levels are given in 2° C steps. The grid spacing is one-half the spacing of that in Figure 26. Hot spots on the left and right sides of the target are eliminated, as previously observed for the coarser probe-sample spacing.

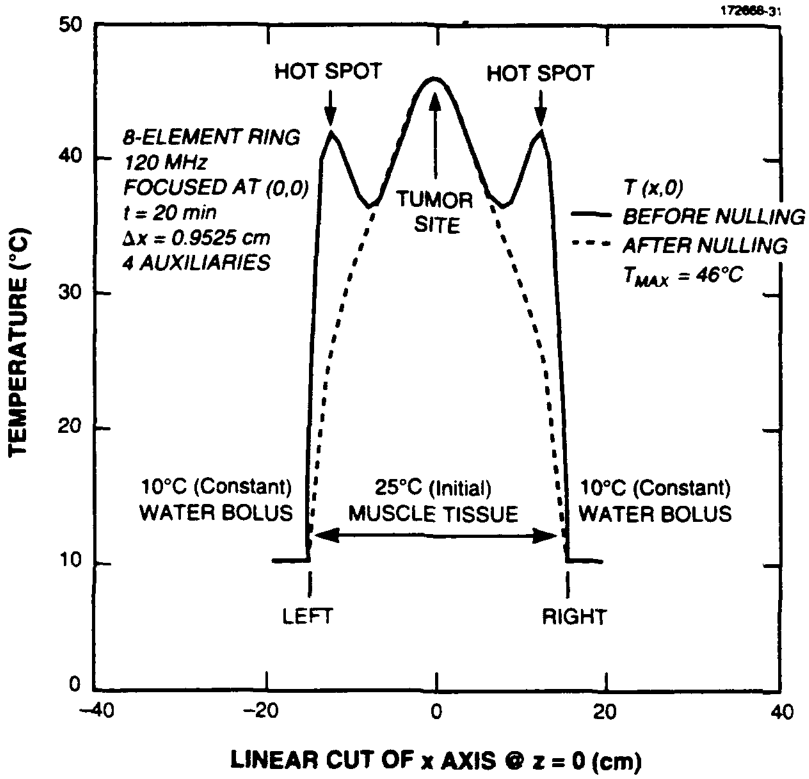


Figure 31. E-field probe-sample-spacing convergence check for simulated one-dimensional ($z = 0$) thermal patterns at time $t = 20$ minutes before and after nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution is at 120 MHz with 0.9525-cm sample spacing. Hot spots on the left and right sides of the target are eliminated by the adaptive nulling process, as previously observed in Figure 27. Four auxiliary sensors are used in the adaptive process.

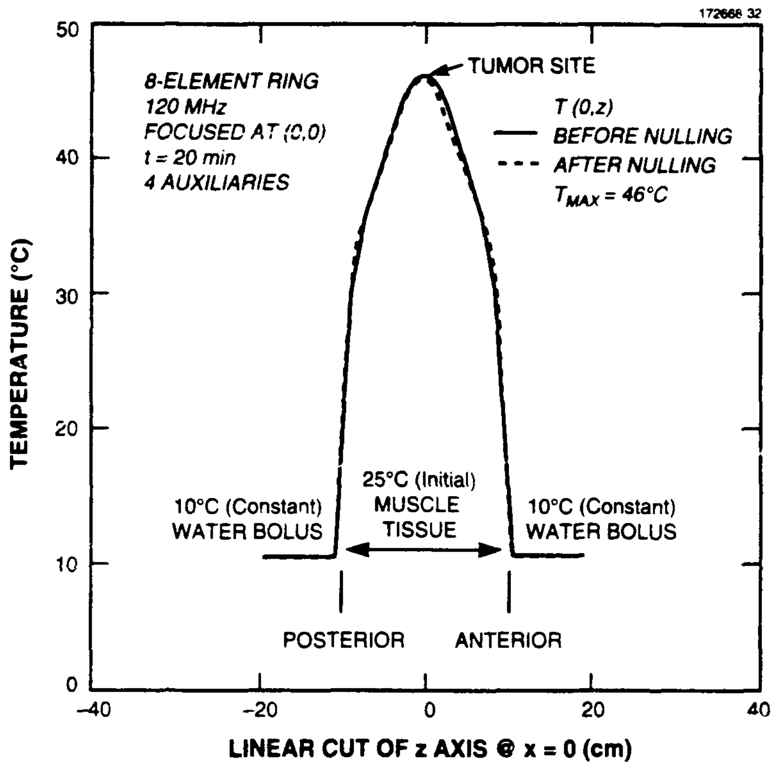


Figure 32. E-field probe-sample-spacing convergence check for simulated one-dimensional ($x = 0$) thermal patterns at time $t = 20$ minutes before and after nulling in elliptical phantom muscle-tissue target surrounded with 10°C constant-temperature water bolus. The quiescent incident RF power distribution is at 120 MHz with 0.9525-cm sample spacing. No undesired hot spots are present. Four auxiliary sensors are used in the adaptive process.

4. CONCLUSION

Focused near-field adaptive nulling has been applied to the problem of generating therapeutic temperature distributions, free of undesired hot spots, for deep-regional cancer treatment in a target body. A noninvasive adaptive hyperthermia system concept has been described and is analyzed in detail. Auxiliary short-dipole nulling sensors are positioned on the surface of the target. Due to the finite width of an electric-field null, a null formed at the target surface extends into the interior region of the target. When the width of the null is properly chosen, undesired hot spots inside the target are eliminated while simultaneously focusing on a deep-seated tumor. The resolution between a deep null and focus is limited by the half-power beamwidth of the hyperthermia array. The resolution can be somewhat enhanced by using weak nulls whenever the spacing between the null and focus is less than one half-power beamwidth. The depth of null is controlled by the SNR at the auxiliary probe position.

A theory for analyzing adaptive nulling with a phased array in an infinite homogeneous conducting medium has been developed. The method of moments is used to compute the electric field at a short-dipole sensor due to a thin-wire dipole ring array. The SMI algorithm is used to adaptively control the transmit array weights and to form radiation pattern nulls. The method used for analyzing the transient thermal behavior of an RF-illuminated target by using an equivalent electric circuit network has been described.

Computer simulations of the electric field and induced temperature distribution for an eight-element dipole ring array at 120 MHz have been presented. Four auxiliary E-field short-dipole sensors are used in eliminating two widely separated hot spots surrounding the array focus at the center of an elliptical target with dimensions 30 cm by 20 cm. Calculated data indicate that simultaneous nulling and focusing can yield desirable electric-field distributions for hyperthermia. Undesired hot spots controlled with an insufficient number of auxiliary nulling probes have been shown to simply redistribute within the target. Thus, an adequate number of auxiliary probes must be provided to ensure that no undesired hot spots are formed in deep-regional hyperthermia therapy. Although the computer simulations are implemented for a phantom target, adaptive nulling may prove beneficial in clinical trials of hyperthermia with living targets. Further adaptive-nulling simulations with inhomogeneous targets are desirable, as are experimental measurements of adaptive hyperthermia in phantom targets. A study of proper number and positioning of the non-invasive sensors for treating arbitrary tumor positions needs to be performed. The adaptive-nulling technique described here should be applicable to both invasive and noninvasive RF hyperthermia systems, as well as ultrasound hyperthermia systems.

APPENDIX A
SYSTEM DEGRADATION DUE TO INSUFFICIENT NUMBER OF AUXILIARY PROBES

This section shows simulation results which indicate that an insufficient number of auxiliary nulling probes will degrade adaptive hyperthermia system performance. The simulation parameters are the same as presented in Section 3, except only two auxiliary probes ($N_{aux} = 2$) are used to form nulls at the surface of the elliptical muscle-tissue target. The geometry used in the simulations is shown in Figure A-1. A 60-cm diameter ring array of eight dipoles uniformly surround a fictitious elliptical target zone with major axis 30 cm and minor axis 20 cm. The length of each perfectly conducting center-fed dipole array element, at 120 MHz in the infinite homogeneous muscle tissue, is $\lambda/2$ or 13.25 cm. The array focus is assumed to be at the origin, and two auxiliary short-dipole sensors with length 1.27 cm are positioned at (15 cm, 0) and (-15 cm, 0). In rectangular coordinates, each dipole is oriented in the $\hat{y}$ direction and the feed terminals of each dipole are located at $y = 0$. The quiescent radiation patterns and thermal distributions are the same as

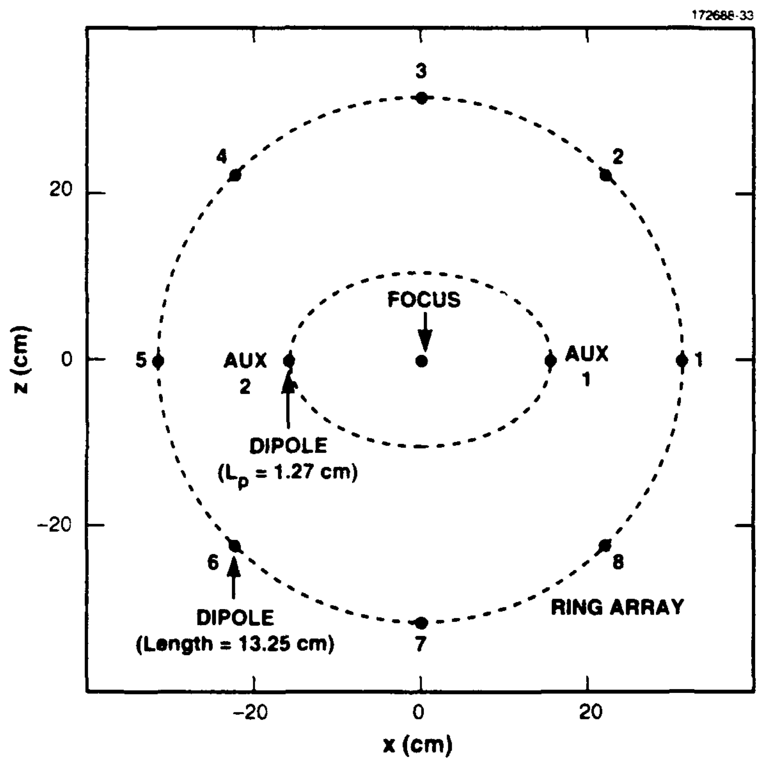


Figure A-1. Geometry for eight-element ring array and two E-field auxiliary sensors.

presented in Figures 14, 15, 16, 17, 23, 24, and 25.

With only two auxiliary sensors, the two-dimensional adapted electric-field radiation pattern shown in Figure A-2 indicates that both auxiliary positions are nulled. However, the two-dimensional adaptive thermal distribution in Figure A-3 shows that the quiescent hot spots (see Figure 23) have only been redistributed. The one-dimensional thermal distribution in Figure A-4 shows that the hot spots are reduced on the left and right sides of the elliptical target, and Figure A-5 shows that the hot spots are redistributed to lie along the z axis. Thus, two additional auxiliary probes (probes 3 and 4 in Figure 13) are necessary to avoid the possibility of redistributed hot spots. Finally, the adaptive weights and covariance matrix eigenvalues are shown in Figures A-6 and A-7, respectively. There are only small differences between the weights and eigenvalues compared to the four-auxiliary probe case (see Figures 21 and 22). The transmit weights cover a 7-dB dynamic range and there are only two nonzero eigenvalues. The probe-array power outputs before and after adaptive nulling are 31.4 dB and 0.0 dB, respectively. Thus, the interference cancellation is to the noise level, or -31.4 dB. An output file for the moment-method simulation for this section is given in Appendix B.

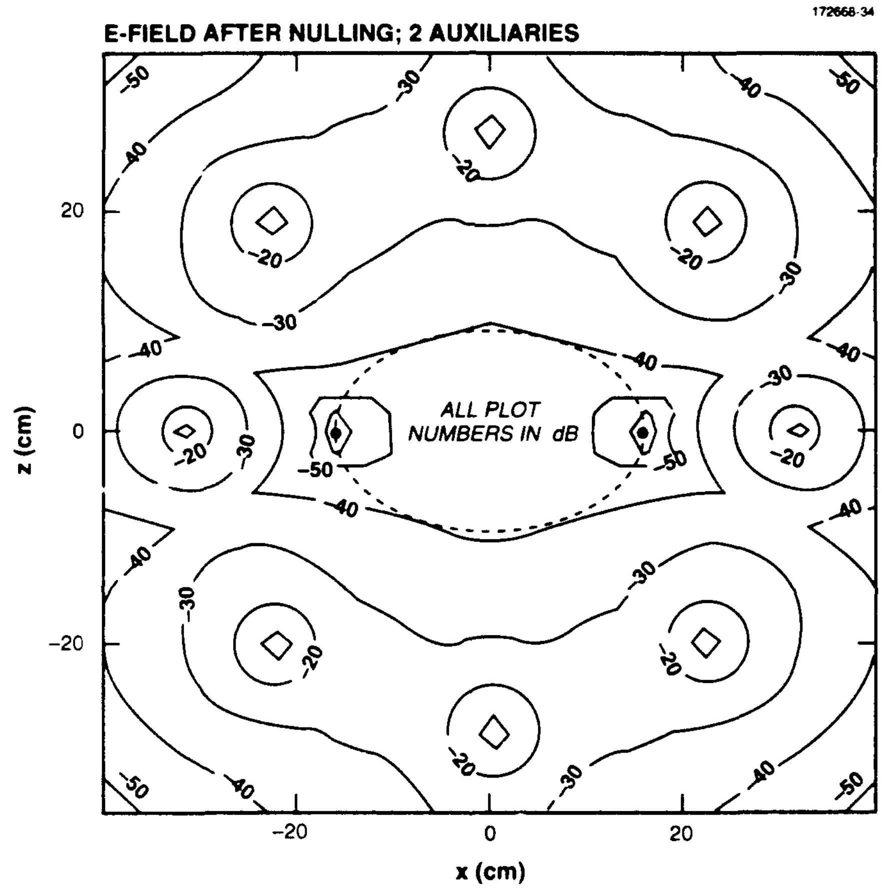


Figure A-2. Simulated two-dimensional adaptive radiation pattern at 120 MHz for eight-element ring array in infinite homogeneous conducting medium (phantom muscle tissue: $\epsilon_r = 73.5, \sigma = 0.5$). Radiation contour levels are given in 10-dB steps. Two auxiliary sensors are used in forming the adaptive pattern. The quiescent focus is at (0,0).

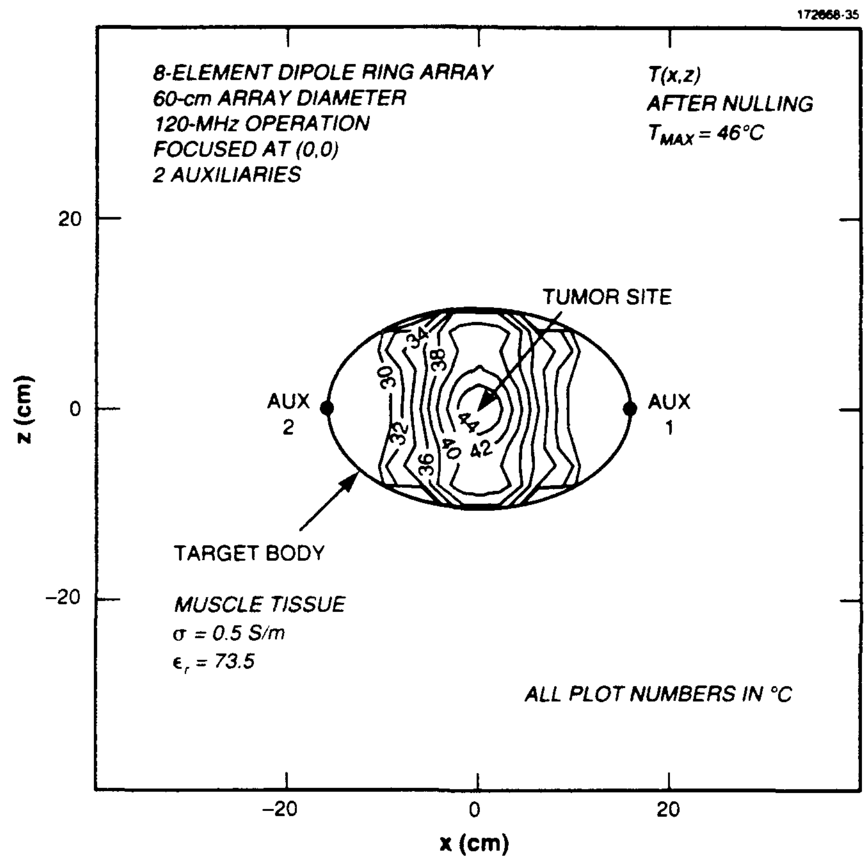


Figure A-3. Simulated two-dimensional thermal pattern at time t = 20 minutes (with adaptive nulling at only two auxiliary sensors in effect) in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The adapted incident RF power distribution, from Figure 18, is at 120 MHz. Temperature contour levels are given in 2° C steps. Initial hot spots (before nulling) on the left and right sides of the target (see Figure 23) are redistributed to the top and bottom (anterior and posterior positions).

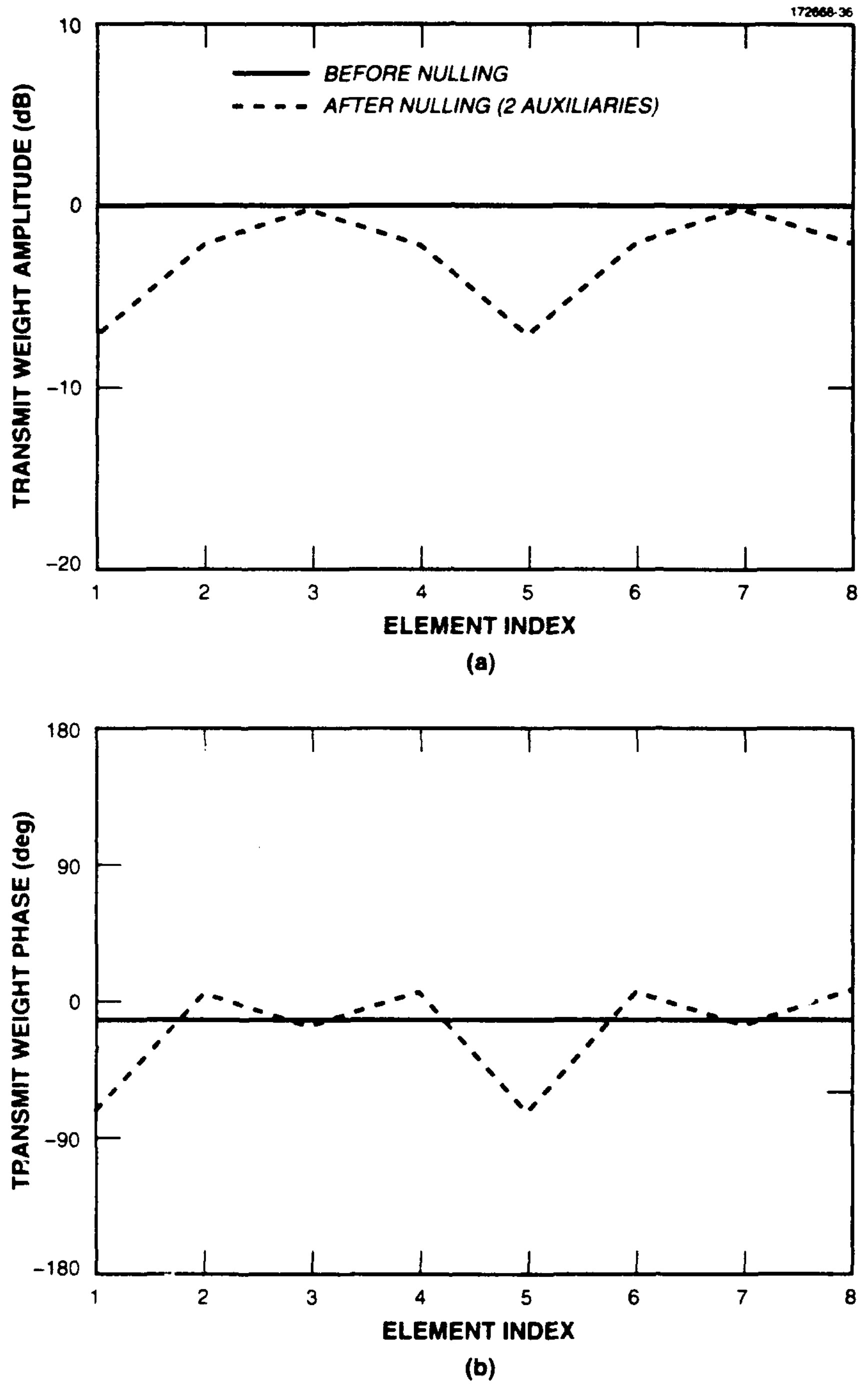


Figure A-4. Simulated one-dimensional ($z = 0$) thermal patterns at time $t = 20$ minutes before and after nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution, from Figure 14, is at 120 MHz. Hot spots on the left and right sides of the target are eliminated by the adaptive nulling process. Two auxiliary sensors are used in the adaptive process.

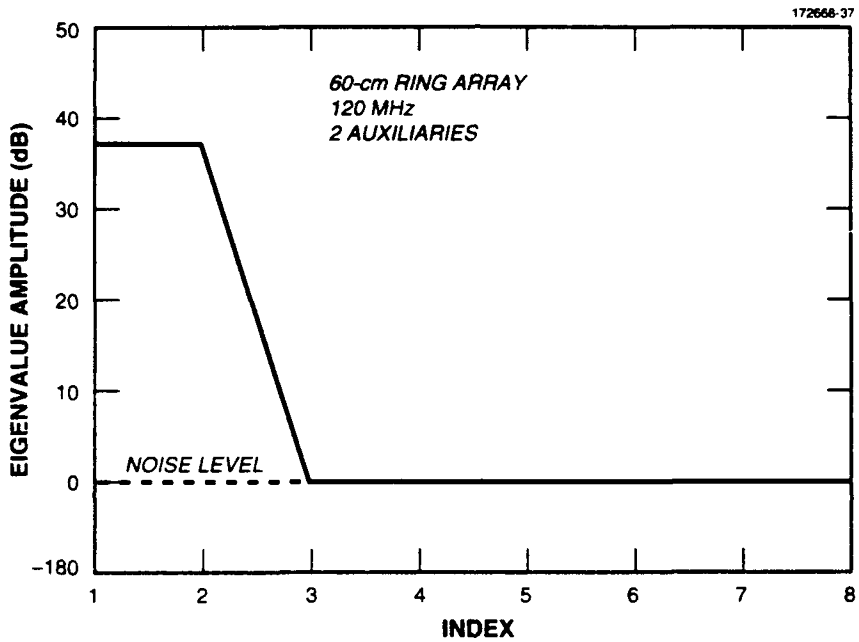


Figure A-5. Simulated one-dimensional ($x = 0$) thermal patterns at time t = 20 minutes before and after nulling in elliptical phantom muscle-tissue target surrounded with 10° C constant-temperature water bolus. The quiescent incident RF power distribution, from Figure 14, is at 120 MHz. Two undesired hot spots are present in the adaptive pattern. Two auxiliary sensors are used in the adaptive process.

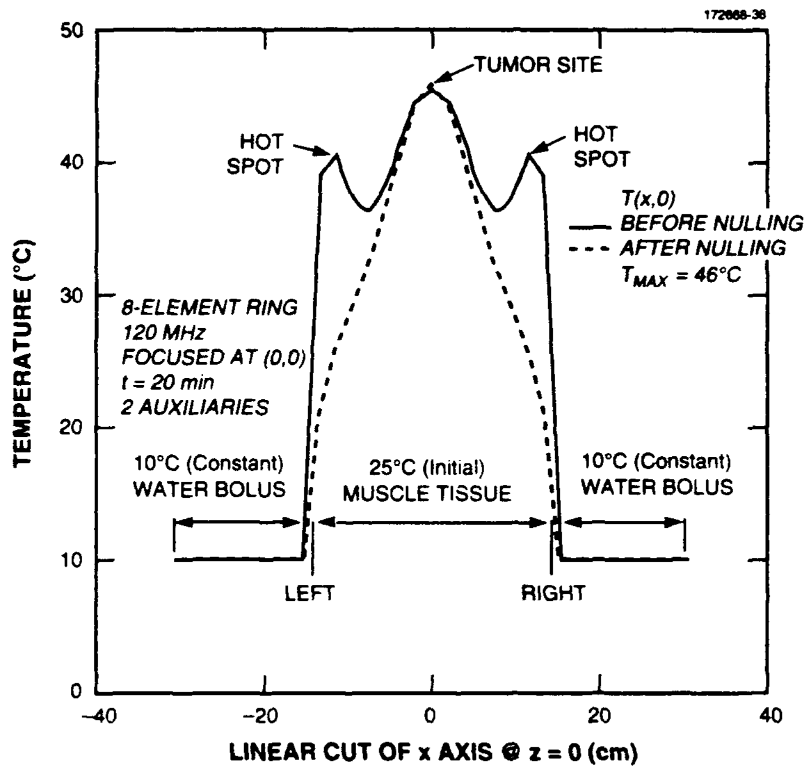


Figure A-6. Transmit array weights before and after adaptive nulling. A dynamic range of about 7 dB is evident for the adaptive weights. Two auxiliary sensors are used in the adaptive process. (a) Amplitude; (b) Phase.

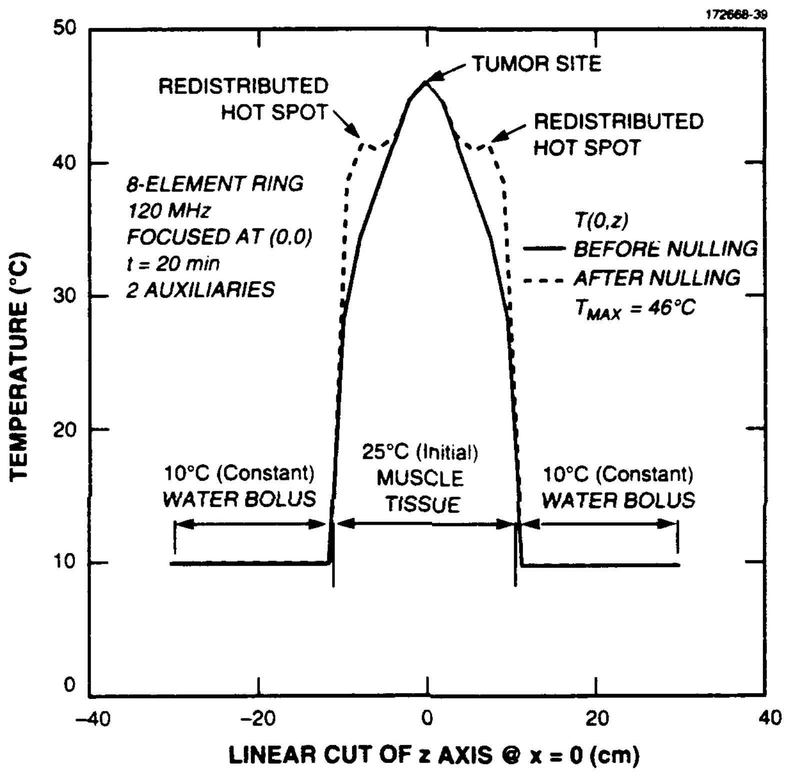


Figure A-7. Channel covariance matrix eigenvalues (degrees of freedom) used in the adaptive process with two auxiliary sensors.

APPENDIX B
SOFTWARE DOCUMENTATION

This appendix lists sample input and output files for the adaptive hyperthermia simulation. Listings of the key FORTRAN programs that implement adaptive nulling in a homogeneous conducting medium are also given. Additional moment-method software not listed here is contained in Richmond [37].

The first data file (four auxiliary sensors) is used to generate the E-field results presented in Section 3.1. The second data file (two auxiliary sensors) was used in calculating the electric-field distribution presented in Appendix A. The corresponding output files give the values for the array mutual coupling. quiescent and adaptive transmit weights, covariance matrix, eigenvalues, and cancellation.

```
***Input data file for adaptive nulling with four auxiliary probes.
***sdipjamhyper data file, filename sdipjamhyper.datacirconicrr4
 &DIPOLE NCOLX=8,NROWY=1,HZIN=0.001,HLIN=2.6,
  DXIN=6.888,DYIN=6.888,ARADIN=0.0039,
  ICIRC=1,CRADIN=11.81,HLSIN=0.25,
  IWL=0,FCHZ=120.0E6,BWFHZ=1.0E0,NFREQ=5,IWR=0,
  NPDX=0,NXDUM=0,NYDUM=0,ER3=73.5,SIG3=0.5,TD3=-1.0,
  ZLOAD=50.0,ZCHAR=0.0,NGEN=8,IGEN=0,
  NSCANS=1,
  THSINC=5.0,
  IMUT=1,IBLTSL=0,
  IENORM=1,ICHEB=0,SLLDB=20.,EDGTDB=0.,
  IPATRN=1,IPRCOM=1,IANGLP=0,
  NPHCT=0,NTHPT=499,
  THDR=180.,THDMIN=-90.,
  NCOLXN=121,NROWYN=1,NCOLZN=1,
  RLSYIN=0.0,RLSZIN=0.0,
  NCOLXN=117,NROWYN=1,NCOLZN=1,
  INEAR=1,
  IPOL=2,IGRNDP=0,
  ITLTPR=0,ITLTDP=0,
  NFCOLX=1,NFROWY=1,
  IANTX=1,IANTY=0,NPOWER=0,
  IPCONN=1,IPCONF=0,IPCUTF=0,IPCFX=0,IPCFY=0,IPCFZ=0,
  ITEK=0,
  IQUAN=0,IRNERR=0,ELERDB=0.02,ELERDG=0.2,NBMOD=12,
  NRAN=1,NBADWT=32,AWERDB=0.0,AWERDG=0.0,
  NAUX=7,IAUXA(1)=1,2,3,4,5,6,7,
  IATTEN=1,
  NJAMS=7,ISLC=0,AUXADB(1)=8*0.0,
  PWRJDB(1)=40.,40.,15.,15.,3*-99.0,
  YJAMIN(1)=7*0.0,
  XFOCIN=0.0,ZFOCIN=0.0,
  XJAMIN(1)=5.9,-5.9,2*0.0,3*-0.0,
  ZJAMIN(1)=0.0,0.0,4.0,-4.C,3*0.0
  YNIN=0.0,
  NCOLXN=101,NTHPT=101,
  XNIN=-15.,ZNIN=-15.,RLSXIN=30.,RLSZIN=30.,NCOLXN=41,NCOLZN=41,
 &END
```

```
***Output data file for adaptive nulling with four auxiliary probes.
***sdipjamhyper data file, filename sdipjamhyper.datacirconicrr4
 XJAMIN(1),YJAMIN(1),ZJAMIN(1)=      5.900       0.000       0.000
 FGHZ=     0.1200000
 DX,DY,HL,ARAD=       0.17496       0.17496       0.06604       0.00010
 IV,VTA=   1    0.5916E-02  0.2211E-02
 IV,VTA=   2    0.5916E-02  0.2211E-02
 IV,VTA=   3    0.5916E-02  0.2211E-02
 IV,VTA=   4    0.5916E-02  0.2211E-02
 IV,VTA=   5    0.5916E-02  0.2211E-02
 IV,VTA=   6    0.5916E-02  0.2211E-02
 IV,VTA=   7    0.5916E-02  0.2211E-02
 IV,VTA=   8    0.5916E-02  0.2211E-02
**********array mutual impedance matrix (first row)*******
 Z(1,   1)= 0.91705E+02 0.12403E+02
 Z(1,   2)=-0.33552E-01 0.19136E+00
 Z(1,   3)=-0.14488E-01-0.54125E-02
 Z(1,   4)= 0.29420E-02 0.13780E-02
 Z(1,   5)= 0.15283E-02-0.11371E-02
 Z(1,   6)= 0.29420E-02 0.13780E-02
 Z(1,   7)=-0.14488E-01-0.54125E-02
 Z(1,   8)=-0.33552E-01 0.19136E+00
 NEL,NCOLX,NROWY=    8    8    1
 CURRENTS
     1  1.000000         0.683E-04       13.
     2  1.000000         0.683E-04       13.
     3  1.000000         0.683E-04       13.
     4  1.000000         0.683E-04       13.
     5  1.000000         0.683E-04       13.
     6  1.000000         0.683E-04       13.
     7  1.000000         0.683E-04       13.
     8  1.000000         0.683E-04       13.
 IC,RWTA=   1    0.35355E+00
 IC,RWTA=   2    0.35355E+00
 IC,RWTA=   3    0.35355E+00
 IC,RWTA=   4    0.35355E+00
 IC,RWTA=   5    0.35355E+00
 IC,RWTA=   6    0.35355E+00
 IC,RWTA=   7    0.35355E+00
 IC,RWTA=   8    0.35355E+00
 I,CWTA(I,1)=   1    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   2    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   3    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   4    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   5    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   6    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   7    0.34510E+00-0.76830E-01
 I,CWTA(I,1)=   8    0.34510E+00-0.76830E-01
*******ring array weights before nulling (amp,phase)*************
 I=   1  CWTADB,CWTADG=        0.00000      -12.55098
 I=   2  CWTADB,CWTADG=        0.00000      -12.55099
 I=   3  CWTADB,CWTADG=        0.00000      -12.55098
 I=   4  CWTADB,CWTADG=        0.00000      -12.55097
 I=   5  CWTADB,CWTADG=        0.00000      -12.55098
 I=   6  CWTADB,CWTADG=        0.00000      -12.55097
 I=   7  CWTADB,CWTADG=        0.00000      -12.55098
 I=   8  CWTADB,CWTADG=        0.00000      -12.55097
 Z(1,   1)= 0.91705E+02 0.12403E+02
 Z(1,   2)=-0.33552E-01 0.19136E+00
```

```
Z(1,    3)=-0.14488E-01-0.54125E-02
Z(1,    4)= 0.29420E-02 0.13780E-02
Z(1,    5)= 0.15283E-02-0.11371E-02
Z(1,    6)= 0.29420E-02 0.13780E-02
Z(1,    7)=-0.14488E-01-0.54125E-02
Z(1,    8)=-0.33552E-01 0.19136E+00
NEL,NCOLX,NROWY=    8     8     1
II,PWRJ IN POWER=   1    0.10000E+05
II,PWRJ IN POWER=   2    0.10000E+05
II,PWRJ IN POWER=   3    0.31623E+02
II,PWRJ IN POWER=   4    0.31623E+02
II,PWRJ IN POWER=   5    0.12589E-09
II,PWRJ IN POWER=   6    0.12589E-09
II,PWRJ IN POWER=   7    0.12589E-09
******covariance matrix***************************
I,J,CNDB,PHASEN=    1    1          36.35          0.00
I,J,CNDB,PHASEN=    1    2          31.68         91.33
I,J,CNDB,PHASEN=    1    3          24.99       -117.95
I,J,CNDB,PHASEN=    1    4          20.88          6.26
I,J,CNDB,PHASEN=    1    5          20.72         -0.14
I,J,CNDB,PHASEN=    1    6          20.88          6.26
I,J,CNDB,PHASEN=    1    7          24.99       -117.95
I,J,CNDB,PHASEN=    1    8          31.68         91.33
I,J,CNDB,PHASEN=    2    1          31.68        -91.33
I,J,CNDB,PHASEN=    2    2          27.05          0.00
I,J,CNDB,PHASEN=    2    3          20.23        155.55
I,J,CNDB,PHASEN=    2    4           8.12        177.91
I,J,CNDB,PHASEN=    2    5          20.90         -6.14
I,J,CNDB,PHASEN=    2    6           8.28        177.98
I,J,CNDB,PHASEN=    2    7          20.24        155.46
I,J,CNDB,PHASEN=    2    8          27.04          0.00
I,J,CNDB,PHASEN=    3    1          24.99        117.95
I,J,CNDB,PHASEN=    3    2          20.23       -155.55
I,J,CNDB,PHASEN=    3    3          16.67          0.00
I,J,CNDB,PHASEN=    3    4          20.26       -155.49
I,J,CNDB,PHASEN=    3    5          25.01        117.95
I,J,CNDB,PHASEN=    3    6          20.27       -155.41
I,J,CNDB,PHASEN=    3    7          16.52          0.00
I,J,CNDB,PHASEN=    3    8          20.24       -155.46
I,J,CNDB,PHASEN=    4    1          20.88         -6.26
I,J,CNDB,PHASEN=    4    2           8.12       -177.91
I,J,CNDB,PHASEN=    4    3          20.26        155.49
I,J,CNDB,PHASEN=    4    4          27.08          0.00
I,J,CNDB,PHASEN=    4    5          31.71        -91.33
I,J,CNDB,PHASEN=    4    6          27.07          0.00
I,J,CNDB,PHASEN=    4    7          20.27        155.41
I,J,CNDB,PHASEN=    4    8           8.28       -177.98
I,J,CNDB,PHASEN=    5    1          20.72          0.14
I,J,CNDB,PHASEN=    5    2          20.90          6.14
I,J,CNDB,PHASEN=    5    3          25.01       -117.95
I,J,CNDB,PHASEN=    5    4          31.71         91.33
I,J,CNDB,PHASEN=    5    5          36.38          0.00
I,J,CNDB,PHASEN=    5    6          31.71         91.33
I,J,CNDB,PHASEN=    5    7          25.01       -117.95
I,J,CNDB,PHASEN=    5    8          20.90          6.14
I,J,CNDB,PHASEN=    6    1          20.88         -6.26
I,J,CNDB,PHASEN=    6    2           8.28       -177.98
I,J,CNDB,PHASEN=    6    3          20.27        155.41
I,J,CNDB,PHASEN=    6    4          27.07          0.00
```

```
I,J,CNDB,PHASEN=   6   5        31.71      -91.33
I,J,CNDB,PHASEN=   6   6        27.08        0.00
I,J,CNDB,PHASEN=   6   7        20.26      155.49
I,J,CNDB,PHASEN=   6   8         8.12     -177.91
I,J,CNDB,PHASEN=   7   1        24.99      117.95
I,J,CNDB,PHASEN=   7   2        20.24     -155.46
I,J,CNDB,PHASEN=   7   3        16.52        0.00
I,J,CNDB,PHASEN=   7   4        20.27     -155.41
I,J,CNDB,PHASEN=   7   5        25.01      117.95
I,J,CNDB,PHASEN=   7   6        20.26     -155.49
I,J,CNDB,PHASEN=   7   7        16.67        0.00
I,J,CNDB,PHASEN=   7   8        20.23     -155.55
I,J,CNDB,PHASEN=   8   1        31.68      -91.33
I,J,CNDB,PHASEN=   8   2        27.04        0.00
I,J,CNDB,PHASEN=   8   3        20.24      155.46
I,J,CNDB,PHASEN=   8   4         8.28      177.98
I,J,CNDB,PHASEN=   8   5        20.90       -6.14
I,J,CNDB,PHASEN=   8   6         8.12      177.91
I,J,CNDB,PHASEN=   8   7        20.23      155.55
I,J,CNDB,PHASEN=   8   8        27.05        0.00
*********eigenvalues*********************
 I,EVLDBN=   1        37.437
 I,EVLDBN=   2        37.195
 I,EVLDBN=   3         2.771
 I,EVLDBN=   4         3.103
 I,EVLDBN=   5         0.000
 I,EVLDBN=   6         0.000
 I,EVLDBN=   7         0.000
 I,EVLDBN=   8         0.000
 CPROD1,CPROD2= 0.13839E+04-0.85265E-13 0.10000E+01 0.00000E+00
 INR NORMALIZATION PARAMETER, INRNOR=    1
 I=   1  WANDB=      -12.318
 I=   2  WANDB=       -8.734
 I=   3  WANDB=       -7.602
 I=   4  WANDB=       -8.734
 I=   5  WANDB=      -12.318
 I=   6  WANDB=       -8.734
 I=   7  WANDB=       -7.602
 I=   8  WANDB=       -8.734
 I,WAN(I,1)=   1   0.11307E+00-0.21414E+00
 I,WAN(I,1)=   2   0.34937E+00 0.10851E+00
 I,WAN(I,1)=   3   0.32832E+00-0.25671E+00
 I,WAN(I,1)=   4   0.34937E+00 0.10851E+00
 I,WAN(I,1)=   5   0.11307E+00-0.21414E+00
 I,WAN(I,1)=   6   0.34937E+00 0.10851E+00
 I,WAN(I,1)=   7   0.32832E+00-0.25671E+00
 I,WAN(I,1)=   8   0.34937E+00 0.10851E+00
*********adaptive array weights (amp.,phase)*********
 I=   1  WANDB,WANDG=       -4.71576     -62.16515
 I=   2  WANDB,WANDG=       -1.13211      17.25469
 I=   3  WANDB,WANDG=        0.00000     -38.02192
 I=   4  WANDB,WANDG=       -1.13210      17.25474
 I=   5  WANDB,WANDG=       -4.71576     -62.16529
 I=   6  WANDB,WANDG=       -1.13210      17.25474
 I=   7  WANDB,WANDG=        0.00000     -38.02193
 I=   8  WANDB,WANDG=       -1.13210      17.25469
*********cancellation*******************
 CPROD1,CPROD2= 0.12271E+01 0.12185E-13 0.10000E+01 0.00000E+00
 INR NORMALIZATION PARAMETER, INRNOR=    1
```

```
 INR= QUI,ADAP ,CANCEL=      31.411      0.889    -30.522    DB
 CANCDB,NRAN,AVECAN=    -30.52207        1      -30.52207
***Input data file for adaptive nulling with two auxiliary probes.
***sdipjamhyper data file, filename sdipjamhyper.datacirconicrr3
 &DIPOLE NCOLX=8,NROWY=1,HZIN=0.001,HLIN=2.6,
  DXIN=6.888,DYIN=6.888,ARADIN=0.0039,
  ICIRC=1,CRADIN=11.81,HLSIN=0.25,
  IWL=0,FCHZ=120.0E6,BWFHZ=1.0E0,NFREQ=5,IWR=0,
  NPDX=0,NXDUM=0,NYDUM=0,ER3=73.5,SIG3=0.5,TD3=-1.0,
  ZLOAD=50.0,ZCHAR=0.0,NGEN=8,IGEN=0,
  NSCANS=1,
  THSINC=5.0,
  IMUT=1,IBLTSL=0,
  IENORM=1,ICHEB=0,SLLDB=20.,EDGTDB=0.,
  IPATRN=1,IPRCOM=1,IANGLP=0,
  NPHCT=0,NTHPT=499,
  THDR=180.,THDMIN=-90.,
  RLSYIN=0.0,RLSZIN=0.0,
  NCOLXN=117,NROWYN=1,NCOLZN=1,
  INEAR=1,
  IPOL=2,IGRNDP=0,
  ITLTPR=0,ITLTDP=0,
  NFCOLX=1,NFROWY=1,
  IANTX=1,IANTY=0,NPOWER=0,
  IPCONN=1,IPCONF=0,IPCUTF=0,IPCFX=0,IPCFY=0,IPCFZ=0,
  ITEK=0,
  IQUAN=0,IRNERR=0,ELERDB=0.02,ELERDG=0.2,NBMOD=12,
  NRAN=1,NBADWT=32,AWERDB=0.0,AWERDG=0.0,
  NAUX=7,IAUXA(1)=1,2,3,4,5,6,7,
  IATTEN=1,
  INRNOR=1,
  NJAMS=7,ISLC=0,AUXADB(1)=8*0.0,
  PWRJDB(1)=40.,40.,5*-99.0,
  YJAMIN(1)=7*0.0,
  XFOCIN=0.0,ZFOCIN=0.0,
  XJAMIN(1)=5.9,-5.9,2*0.0,2*-4.0,1*0.0,
  ZJAMIN(1)=0.0,0.0,4.0,-4.0,3.0,-3.0,1*0.0
  YNIN=0.0,
  NCOLXN=101,NTHPT=101,
  XNIN=-15.,ZNIN=-15.,RLSXIN=30.,RLSZIN=30.,NCOLXN=41,NCOLZN=41,
 &END
***Output data file for adadptive nulling with two auxiliary probes.
***sdipjamhyper data file, filename sdipjamhyper.datacirconicrr3
 XJAMIN(1),YJAMIN(1),ZJAMIN(1)=       5.900       0.000       0.000

 FGHZ=       0.1200000
 DX,DY,HL,ARAD=          0.17496       0.17496       0.06604       0.00010
 IV,VTA=   1     0.5916E-02  0.2211E-02
 IV,VTA=   2     0.5916E-02  0.2211E-02
 IV,VTA=   3     0.5916E-02  0.2211E-02
 IV,VTA=   4     0.5916E-02  0.2211E-02
 IV,VTA=   5     0.5916E-02  0.2211E-02
 IV,VTA=   6     0.5916E-02  0.2211E-02
 IV,VTA=   7     0.5916E-02  0.2211E-02
 IV,VTA=   8     0.5916E-02  0.2211E-02
**********array mutual impedance matrix (first row)*******
 Z(1,   1)= 0.91705E+02 0.12403E+02
 Z(1,   2)=-0.33552E-01 0.19136E+00
 Z(1,   3)=-0.14488E-01-0.54125E-02
```

```
Z(1,   4)= 0.29420E-02 0.13780E-02
Z(1,   5)= 0.15283E-02-0.11371E-02
Z(1,   6)= 0.29420E-02 0.13780E-02
Z(1,   7)=-0.14488E-01-0.54125E-02
Z(1,   8)=-0.33552E-01 0.19136E+00
NEL,NCOLX,NROWY=    8    8    1
CURRENTS
    1  1.000000          0.683E-04          13.
    2  1.000000          0.683E-04          13.
    3  1.000000          0.683E-04          13.
    4  1.000000          0.683E-04          13.
    5  1.000000          0.683E-04          13.
    6  1.000000          0.683E-04          13.
    7  1.000000          0.683E-04          13.
    8  1.000000          0.683E-04          13.
IC,RWTA=    1    0.35355E+00
IC,RWTA=    2    0.35355E+00
IC,RWTA=    3    0.35355E+00
IC,RWTA=    4    0.35355E+00
IC,RWTA=    5    0.35355E+00
IC,RWTA=    6    0.35355E+00
IC,RWTA=    7    0.35355E+00
IC,RWTA=    8    0.35355E+00
I,CWTA(I,1)=    1    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    2    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    3    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    4    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    5    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    6    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    7    0.34510E+00-0.76830E-01
I,CWTA(I,1)=    8    0.34510E+00-0.76830E-01
******ring array weights before nulling (amp,phase)************
I=   1  CWTADB,CWTADG=          0.00000      -12.55098
I=   2  CWTADB,CWTADG=          0.00000      -12.55099
I=   3  CWTADB,CWTADG=          0.00000      -12.55098
I=   4  CWTADB,CWTADG=          0.00000      -12.55097
I=   5  CWTADB,CWTADG=          0.00000      -12.55098
I=   6  CWTADB,CWTADG=          0.00000      -12.55097
I=   7  CWTADB,CWTADG=          0.00000      -12.55098
I=   8  CWTADB,CWTADG=          0.00000      -12.55097
Z(1,   1)= 0.91705E+02 0.12403E+02
Z(1,   2)=-0.33552E-01 0.19136E+00
Z(1,   3)=-0.14488E-01-0.54125E-02
Z(1,   4)= 0.29420E-02 0.13780E-02
Z(1,   5)= 0.15283E-02-0.11371E-02
Z(1,   6)= 0.29420E-02 0.13780E-02
Z(1,   7)=-0.14488E-01-0.54125E-02
Z(1,   8)=-0.33552E-01 0.19136E+00
NEL,NCOLX,NROWY=    8    8    1
II,PWRJ IN POWER=   1    0.10000E+05
II,PWRJ IN POWER=   2    0.10000E+05
II,PWRJ IN POWER=   3    0.12589E-09
II,PWRJ IN POWER=   4    0.12589E-09
II,PWRJ IN POWER=   5    0.12589E-09
II,PWRJ IN POWER=   6    0.12589E-09
II,PWRJ IN POWER=   7    0.12589E-09
******covariance matrix****************************
I,J,CNDB,PHASEN=   1   1          36.35         0.00
I,J,CNDB,PHASEN=   1   2          31.68        91.33
```

```
I,J,CNDB,PHASEN=   1   3        24.98     -117.94
I,J,CNDB,PHASEN=   1   4        20.88        6.28
I,J,CNDB,PHASEN=   1   5        20.72       -0.14
I,J,CNDB,PHASEN=   1   6        20.88        6.28
I,J,CNDB,PHASEN=   1   7        24.98     -117.94
I,J,CNDB,PHASEN=   1   8        31.68       91.33
I,J,CNDB,PHASEN=   2   1        31.68      -91.33
I,J,CNDB,PHASEN=   2   2        27.05        0.00
I,J,CNDB,PHASEN=   2   3        20.24      155.47
I,J,CNDB,PHASEN=   2   4         8.24      177.97
I,J,CNDB,PHASEN=   2   5        20.90       -6.17
I,J,CNDB,PHASEN=   2   6         8.24      177.97
I,J,CNDB,PHASEN=   2   7        20.24      155.47
I,J,CNDB,PHASEN=   2   8        27.04        0.00
I,J,CNDB,PHASEN=   3   1        24.98      117.94
I,J,CNDB,PHASEN=   3   2        20.24     -155.47
I,J,CNDB,PHASEN=   3   3        16.62        0.00
I,J,CNDB,PHASEN=   3   4        20.27     -155.41
I,J,CNDB,PHASEN=   3   5        25.01      117.93
I,J,CNDB,PHASEN=   3   6        20.27     -155.41
I,J,CNDB,PHASEN=   3   7        16.52        0.00
I,J,CNDB,PHASEN=   3   8        20.24     -155.47
I,J,CNDB,PHASEN=   4   1        20.88       -6.28
I,J,CNDB,PHASEN=   4   2         8.24     -177.97
I,J,CNDB,PHASEN=   4   3        20.27      155.41
I,J,CNDB,PHASEN=   4   4        27.08        0.00
I,J,CNDB,PHASEN=   4   5        31.71      -91.33
I,J,CNDB,PHASEN=   4   6        27.07        0.00
I,J,CNDB,PHASEN=   4   7        20.27      155.41
I,J,CNDB,PHASEN=   4   8         8.24     -177.97
I,J,CNDB,PHASEN=   5   1        20.72        0.14
I,J,CNDB,PHASEN=   5   2        20.90        6.17
I,J,CNDB,PHASEN=   5   3        25.01     -117.93
I,J,CNDB,PHASEN=   5   4        31.71       91.33
I,J,CNDB,PHASEN=   5   5        36.38        0.00
I,J,CNDB,PHASEN=   5   6        31.71       91.33
I,J,CNDB,PHASEN=   5   7        25.01     -117.93
I,J,CNDB,PHASEN=   5   8        20.90        6.17
I,J,CNDB,PHASEN=   6   1        20.88       -6.28
I,J,CNDB,PHASEN=   6   2         8.24     -177.97
I,J,CNDB,PHASEN=   6   3        20.27      155.41
I,J,CNDB,PHASEN=   6   4        27.07        0.00
I,J,CNDB,PHASEN=   6   5        31.71      -91.33
I,J,CNDB,PHASEN=   6   6        27.08        0.00
I,J,CNDB,PHASEN=   6   7        20.27      155.41
I,J,CNDB,PHASEN=   6   8         8.24     -177.97
I,J,CNDB,PHASEN=   7   1        24.98      117.94
I,J,CNDB,PHASEN=   7   2        20.24     -155.47
I,J,CNDB,PHASEN=   7   3        16.52        0.00
I,J,CNDB,PHASEN=   7   4        20.27     -155.41
I,J,CNDB,PHASEN=   7   5        25.01      117.93
I,J,CNDB,PHASEN=   7   6        20.27     -155.41
I,J,CNDB,PHASEN=   7   7        16.62        0.00
I,J,CNDB,PHASEN=   7   8        20.24     -155.47
I,J,CNDB,PHASEN=   8   1        31.68      -91.33
I,J,CNDB,PHASEN=   8   2        27.04        0.00
I,J,CNDB,PHASEN=   8   3        20.24      155.47
I,J,CNDB,PHASEN=   8   4         8.24      177.97
I,J,CNDB,PHASEN=   8   5        20.90       -6.17
```

```
I,J,CNDB,PHASEN=   8   6         8.24       177.97
I,J,CNDB,PHASEN=   8   7        20.24       155.47
I,J,CNDB,PHASEN=   8   8        27.05         0.00
*********eigenvalues**********************
I,EVLDBN=   1         37.195
I,EVLDBN=   2         37.437
I,EVLDBN=   3          0.000
I,EVLDBN=   4          0.000
I,EVLDBN=   5          0.000
I,EVLDBN=   6          0.000
I,EVLDBN=   7          0.000
I,EVLDBN=   8          0.000
CPROD1,CPROD2= 0.13835E+04-0.56843E-13 0.10000E+01 0.00000E+00
INR NORMALIZATION PARAMETER, INRNOR=   1
I=   1  WANDB=      -14.003
I=   2  WANDB=       -8.939
I=   3  WANDB=       -6.886
I=   4  WANDB=       -8.939
I=   5  WANDB=      -14.003
I=   6  WANDB=       -8.939
I=   7  WANDB=       -6.886
I=   8  WANDB=       -8.939
I,WAN(I,1)=   1   0.48427E-01-0.19349E+00
I,WAN(I,1)=   2   0.35635E+00 0.26440E-01
I,WAN(I,1)=   3   0.43481E+00-0.12564E+00
I,WAN(I,1)=   4   0.35635E+00 0.26441E-01
I,WAN(I,1)=   5   0.48427E-01-0.19349E+00
I,WAN(I,1)=   6   0.35635E+00 0.26441E-01
I,WAN(I,1)=   7   0.43481E+00-0.12564E+00
I,WAN(I,1)=   8   0.35635E+00 0.26441E-01
*********adaptive array weights (amp.,phase)*********
I=   1  WANDB,WANDG=       -7.11720     -75.94859
I=   2  WANDB,WANDG=       -2.05298       4.24348
I=   3  WANDB,WANDG=        0.00000     -16.11658
I=   4  WANDB,WANDG=       -2.05297       4.24352
I=   5  WANDB,WANDG=       -7.11719     -75.94872
I=   6  WANDB,WANDG=       -2.05298       4.24352
I=   7  WANDB,WANDG=        0.00000     -16.11658
I=   8  WANDB,WANDG=       -2.05297       4.24348
*********cancellation*******************
CPROD1,CPROD2= 0.10001E+01-0.63768E-14 0.10000E+01 0.00000E+00
INR NORMALIZATION PARAMETER, INRNOR=   1
INR= QUI,ADAP ,CANCEL=      31.410     0.000    -31.410    DB
CANCDB,NRAN,AVECAN=   -31.40966      1     -31.40966
*****************makefile for moment method software*******
****file sdipjamhyperMake*****
EXECUTABLE = sdipjamhyper.out
OBJS = sdipjamhyper.o dzabgnloss.o \
	dpack2.o zabgenloss.o pack2.o fwgh.o \
	chebaf.o wwquan.o taylor.o \
	eigenv.o reordr.o ydipsubloss.o \
	plothyper.o contek.o plabel.o \
	conturek.o circsubloss.o
# use tabs for continues
IL = /usr/lib/f68881/libm.il
FLAGS = -O1 -v -f68881
$(EXECUTABLE): $(OBJS)
	dislink -lF77 $(FLAGS) -o $(EXECUTABLE) $(OBJS) $(IL)
```

```
sdipjamhyper.o: /home/ajf/hyperthermia/sdipjamhyper.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/sdipjamhyper.f $(IL)

plothyper.o: /home/ajf/hyperthermia/plothyper.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/plothyper.f $(IL)

dzabgnloss.o: /home/ajf/hyperthermia/dzabgnloss.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/dzabgnloss.f $(IL)

dpack2.o: /home/ajf/monjam/dpack2.f
	f77 $(FLAGS) -c /home/ajf/monjam/dpack2.f $(IL)

contek.o: /home/ajf/monjam/contek.f
	f77 $(FLAGS) -c /home/ajf/monjam/contek.f $(IL)

plabel.o: /home/ajf/monjtr/plabel.f
	f77 $(FLAGS) -c /home/ajf/monjtr/plabel.f $(IL)

conturek.o: /home/ajf/hyperthermia/conturek.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/conturek.f $(IL)
zabgenloss.o: /home/ajf/hyperthermia/zabgenloss.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/zabgenloss.f $(IL)
pack2.o: /home/ajf/monjam/pack2.f
	f77 $(FLAGS) -c /home/ajf/monjam/pack2.f $(IL)
fwgh.o: /home/ajf/monjam/fwgh.f
	f77 $(FLAGS) -c /home/ajf/monjam/fwgh.f $(IL)

chebaf.o: /home/ajf/monjam/chebaf.f
	f77 $(FLAGS) -c /home/ajf/monjam/chebaf.f $(IL)

wwquan.o: /home/ajf/monjam/wwquan.f
	f77 $(FLAGS) -c /home/ajf/monjam/wwquan.f $(IL)

taylor.o: /home/ajf/monjam/taylor.f
	f77 $(FLAGS) -c /home/ajf/monjam/taylor.f $(IL)

eigenv.o: /home/ajf/monjam/eigenv.f
	f77 $(FLAGS) -c /home/ajf/monjam/eigenv.f $(IL)
reordr.o: /home/ajf/monjam/reordr.f
	f77 $(FLAGS) -c /home/ajf/monjam/reordr.f $(IL)

ydipsubloss.o: /home/ajf/hyperthermia/ydipsubloss.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/ydipsubloss.f $(IL)

circsubloss.o: /home/ajf/hyperthermia/circsubloss.f
	f77 $(FLAGS) -c /home/ajf/hyperthermia/circsubloss.f $(IL)
```



```
******file sdipjamhyper.f*************
C***PROGRAM SDIPJAMHYPER.F ---  ANALYZES FINITE ARRAYS OF DIPOLES
C***IN LOSSY DIELECTRIC OR FREE SPACE.
C***THE DIPOLES ARE ASSUMED TO BE ORIENTED PARALLEL
C***TO THE PLANE OF THE GRID FOR A PLANAR ARRAY, OR THEY CAN BE
C***ARRANGED IN AN ANNULAR (RING) ARRAY CONFIGURATION.
C***RECEIVING CONDITIONS ARE ASSUMED.
C***DOUBLE-PRECISION VERSION
      PARAMETER (NUMCHN=8)
      PARAMETER (NUMAUX=8)
```

```
      PARAMETER (NUMJAM=7)
      PARAMETER (NUMELM=8)
      PARAMETER (NUMFRQ=5)
      PARAMETER (NUMNPT=1681)
      COMPLEX PS(1825),CZ(900),VA(NUMELM),Z(NUMELM,NUMELM)
      COMPLEX VTA(NUMELM),VREFA,VRECVA(NUMELM)
      COMPLEX VCW(NUMELM),VRECVX(NUMELM),CJ,CSUMA,VXA
CC***THE ABOVE MATRICES ARE DIMENSIONED BY THE NUMBER OF ELEMENTS
      COMPLEX *16 COVNF(NUMCHN,NUMCHN),COVNFI(NUMCHN,NUMCHN)
      COMPLEX *16 COVAAJ(NUMCHN,NUMCHN),CINVCN(NUMCHN,NUMCHN)
      COMPLEX *16 VCHA(NUMCHN,NUMFRQ)
      COMPLEX *16 VMAJMA(NUMJAM,NUMFRQ)
      COMPLEX *16 VAUXJA(NUMAUX,NUMJAM,NUMFRQ)
      COMPLEX *8 VAXCWA(NUMAUX,NUMNPT)
C***NOTE DIMENS.  VCHA(NCHAN,NFREQ),VMAJMA(NJAM,NFREQ)                    DIP00200
C                 VAUXJA(NAUX,NJAM,NFREQ),VAXCWA(NAUX,NPTS)               DIP00210
      COMPLEX *8 VXMANA(NUMNPT),ETASP,GAMSP
      COMPLEX *16 EIGVAN(NUMCHN),EIGVEN(NUMCHN,NUMCHN)
      COMPLEX *16 WANA(NUMCHN,1)
      COMPLEX *16 WTCTR(1,NUMCHN),WQSLC(NUMCHN,1)
      COMPLEX *16 WQNA(NUMELM,1),CMPROD(1,NUMCHN),WAN(NUMCHN,1)
      COMPLEX *16 WANDMA(NUMELM,1)
      COMPLEX *16 CWTA(NUMELM,1),ETADP,GAMDP
      DIMENSION ACALPH(NUMELM),RWTA(NUMELM),RWTB(NUMELM)
      DIMENSION VXADB(NUMNPT),VXAPH(NUMNPT)
      DIMENSION XC(NUMNPT),YC(NUMNPT),ZC(NUMNPT)
      DIMENSION THSD(1),PHSD(1),THCT(1),PHCT(1)                           DIP00320
      DIMENSION XJAMIN(NUMJAM),YJAMIN(NUMJAM),ZJAMIN(NUMJAM)
      DIMENSION XJAM(NUMJAM),YJAM(NUMJAM),ZJAM(NUMJAM)
      DIMENSION PWRJDB(NUMJAM)                                            DIP00340
      DIMENSION PCHADB(NUMNPT)
      REAL *8 CNDB,PHASEN,DPDCR,WKE(144),EVLDBN                           DIP00370
      REAL *8 PREALN,PIMAGN                                               DIP00380
      REAL *8 QINRDB,AINRDB,CANCDB,EXCPCB                                 DIP00390
      REAL *8 QINRAA,AINRAA                                               DIP00400
      REAL *8 CNCLNA,ELSGDB,ELSGDG,AWSGDB,AWSGDG                          DIP00410
      REAL *8 SUMC,SUMAA,AVECAN,AVCAA,SAVE,SAVESR
      REAL *4 RX(100),RY(100),RZ(100),FM(41,41),DMF(41,41)
      REAL *4 AUXADB(NUMAUX),VATENA(NUMAUX)
      INTEGER IAUXA(NUMAUX),LTMP(NUMCHN),MTMP(NUMCHN)
      CHARACTER DATNAM*35, OUTNAM*35
      COMMON/A/DX,DY,NCOLX,NROWY,NEL,HZ,HL,ARAD,ZLOAD,ZCHAR               DIP00460
      COMMON /B/ NGEN,IGEN,THETAS,PHIS,IMUT,IBLTSL,IPATRN                 DIP00470
      COMMON /D/ FGHZ,RLAMDA,IWL,IS,NSCANS                                DIP00480
      COMMON /DEG/ THDR,THDMIN                                            DIP00490
      COMMON /PLT/ INEAR,IPLOTM                                           DIP00500
      COMMON/XLYL/XL,YL                                                   DIP00510
      COMMON/NEAR/IANTX,IANTY,NPOWER,IPCONN,IPCONF,IPCUTF                 DIP00520
      COMMON/NEAR2/IPCFX,IPCFY,IPCFZ                                      DIP00530
      COMMON /P/XN,YN,ZN,RLSX,RLSY,RLSZ,NCOLXN,NROWYN,NCOLZN              DIP00540
      COMMON /P2/EDGET,ICOMB,PUNFLX,PUNFLY                                DIP00550
      COMMON /CHEBY/ ICHEB,SLLDB                                          DIP00560
      COMMON /GROUND/ IGRNDP                                              DIP00570
      COMMON /NORM/ IENORM,BIGNDB                                         DIP00580
      COMMON /NORMAL/ INRNOR
      COMMON/PCENTR/PCDXIN                                                DIP00590
      COMMON /WRITE/ IWR                                                  DIP00600
      COMMON /SPLOSS/ ETASP,GAMSP
      COMMON /DPLOSS/ ETADP,GAMDP
```

```
      COMMON /F/ FHZ,ER3,SIG3,TD3
      COMMON /CIRCLE/ ICIRC,RADIUS,HLS
      NAMELIST/DIPOLE/NCOLX,NROWY,HZIN,HLIN,ARADIN,ZLOAD,NGEN,IGEN,
     1DXIN,DYIN,IMUT,NSCANS,THSD,PHSD,THSINC,IBLTSL,IPATRN,ER3,SIG3,TD3,
     2NPHCT,PHCT,NTHPT,ZCHAR,IWL,FCHZ,BWFHZ,NFREQ,THDR,THDMIN,ICIRC,
     3XNIN,YNIN,ZNIN,NCOLXN,NROWYN,NCOLZN,INEAR,IPRCOM,IANGLP,CRADIN,
     4IANTX,IANTY,NPOWER,NFCOLX,NFROWY,IPCONN,IPCONF,IPCUTF,IPOL,HLSIN,
     5IPCFX,IPCFY,IPCFZ,ITEK,IGRNDP,IGAIN,ICHEB,IWR,IENORM,BIGNDB,
     6NXDUM,NYDUM,NPDX,EDGTDB,RLSXIN,RLSYIN,RLSZIN,ISLC,INUNIF,
     7XFOCIN,ZFOCIN,XJAMIN,YJAMIN,ZJAMIN,IATTEN,AUXADB,INRNOR,
     8NJAMS,PWRCDB,PWRJDB,NAUX,IAUXA,IAUXB,ITLTPR,ITLTDP,
     9IQUAN,IRNERR,ELERDB,ELERDG,AWERDB,AWERDG,NBMOD,NBADWT,NRAN,SLLDB
C**NOTE; IF IWL=0 (INCHES), IWL=2 (METERS)
      WRITE(6,2959)
2959  FORMAT(1X,'ENTER INPUT DATA FILE NAME (typ. sdipjamhyper.data)')
      READ(5,*) DATNAM
      OPEN(4,FILE=DATNAM,FORM='FORMATTED')
      WRITE(6,3959)
3959  FORMAT(1X,'ENTER OUTPUT DATA FILE NAME (typ sdipjamhyper.output)')
      READ(5,*) OUTNAM
      OPEN(8,FILE=OUTNAM,FORM='FORMATTED')
      CALL GETCP2(CPU1)
      PI=3.141592654
      DCR=PI/180.
      DPDCR=DCR
      CJ=(0.,1.)
      CINMTR=0.0254
      IGRNDP=1
      IGAIN=0
      ZLOAD=0.0
      ICHEB=0
      IWR=0
      IENORM=1
      EXCPCB=-1.0D0
      IQUAN=0
      NRAN=1
      TILTPR=0.0
      TILTDP=0.0
      IATTEN=0
      IPOL=1
      ER3=1.0
      SIG3=0.0
      TD3=-1.0
      ICIRC=0
      CRADIN=0.0
      HLSIN=0.0
      ISLC=1
      INUNIF=0
      INRNOR=0
      READ(4,DIPOLE)
      IF(NFREQ.EQ.1) BWFHZ=0.0
      IF(ISLC.EQ.0) WRITE(6,81100)
81100 FORMAT(1X,'FULLY ADAPTIVE ARRAY')
      IF(ISLC.EQ.0.AND.ICHEB.EQ.0) INUNIF=1
      IF(ICIRC.EQ.1) WRITE(6,7000)
7000  FORMAT(1X,'RING ARRAY GEOMETRY')
      FHZ=FCHZ
      IXZ=0
```

```
      IXY=0
      IF(NROWYN.EQ.1) IXZ=1
      IF(NCOLZN.EQ.1) IXY=1
      IF(IPOL.EQ.2) IPRCOM=0                                            DIP00890
      IF(IPRCOM.EQ.0) WRITE(6,7898)                                     DIP00900
 7898 FORMAT(1X,'NO PROBE COMPENSATION')                                DIP00910
      WRITE(6,9276)IGRNDP                                               DIP00920
 9276 FORMAT(1X'GROUND PLANE PARAMETER, IGRNDP=',I4)                    DIP00930
CC    DO 1615 IX=1,NAUX                                                 DIP00940
CC    IF(IATTEN.EQ.0) AUXADB(IX)=0.0                                    DIP00950
CC    WRITE(6,2318)IX,AUXADB(IX)                                        DIP00960
C2318  FORMAT(1X,'IX,AUXADB(IX)=',I4,2X,F12.2)                          DIP00970
CC    VATENA(IX)=10.**(AUXADB(IX)/20.)                                  DIP00980
C1615  CONTINUE                                                         DIP00990
      IF(NJAMS.GT.1) WRITE(8,2009)XJAMIN(1),YJAMIN(1),ZJAMIN(1)         DIP01000
 2009 FORMAT(1X,'XJAMIN(1),YJAMIN(1),ZJAMIN(1)=',3F12.3)                DIP01010
      WRITE(6,6987) BWFHZ,NFREQ                                         DIP01020
 6987 FORMAT(1X,'BWFHZ,NFREQ=',E12.5,2X,I5)                             DIP01030
      IF(ITEK.EQ.0)CALL COMPRS                                          DIP01040
      IF(ITEK.EQ.1)CALL TEKALL(4014,480,0,1,0)                          DIP01050
CC    IF(INEAR EQ.0) CALL PRNTDA                                        DIP01060
      IF(IMUT.EQ.0) ZLOAD=1.0                                           DIP01070
      IF(ITLTPR.EQ.1) TILTPR=45.                                        DIP01080
      IF(ITLTDP.EQ.1) TILTDP=45.                                        DIP01090
      CTPR=COS(TILTPR*DCR)                                              DIP01100
      STPR=SIN(TILTPR*DCR)                                              DIP01110
      CTDP=COS(TILTDP*DCR)                                              DIP01120
      STDP=SIN(TILTDP*DCR)                                              DIP01130
      FGHZ=FCHZ/1.0E9                                                   DIP01140
      NEL=NCOLX*NROWY                                                   DIP01150
      RNEL=NEL
      XLIN=DXIN*(NCOLX-1)                                               DIP01160
      YLIN=DYIN*(NROWY-1)                                               DIP01170
      NACOLX=NCOLX-NPDX-2*NXDUM                                         DIP01180
      NAROWY=NROWY-2*NYDUM                                              DIP01190
      NAEL=NACOLX*NAROWY
      PCDXIN=NPDX*DXIN                                                  DIP01200
      NAUXP1=NAUX+1                                                     DIP01220
      NAUXP2=NAUX+2                                                     DIP01230
      IF(ISLC.EQ.1) NMAX=NAUXP1
      IF(ISLC.EQ.0) NMAX=NEL
      NMAXP1=NMAX+1                                                     DIP01250
      ELSGDB=ELERDB*SQRT(3.)                                            DIP01260
      ELSGDG=ELERDG*SQRT(3.)                                            DIP01270
      AWSGDB=AWERDB*SQRT(3.)                                            DIP01280
      AWSGDG=AWERDG*SQRT(3.)                                            DIP01290
      INITRN=1                                                          DIP01300
      IF(ISLC.EQ.0) THEN
        NAUX=NEL
        NAUXP1=NAUX+1
        NAUXP2=NAUX+2
        DO 79130 I=1,NEL
        IAUXA(I)=I
        WRITE(6,76767)I,IAUXA(I)
76767   FORMAT(1X,'I,IAUXA(I)=',2I5)
79130   CONTINUE
      ENDIF
      DO 1615 IX=1,NAUX
      IF(IATTEN.EQ.0) AUXADB(IX)=0.0
```

```
      WRITE(6,2318)IX,AUXADB(IX)
2318  FORMAT(1X,'IX,AUXADB(IX)=',I4,2X,F12.2)
      VATENA(IX)=10.**(AUXADB(IX)/20.)
1615  CONTINUE
30    CONTINUE
      NR=1
      IF(NGEN.EQ.1) NSCANS=0
      ICC=NEL
      ICC1=ICC
      IF(IBLTSL.EQ.1) ICC1=NROWY
CC    IF(IWL.EQ.1)GO TO 50
        WRITE(8,40)FGHZ
40      FORMAT(/,1X,'FGHZ=',F15.7)
C************************************************************************
C*****COMPUTE FREE SPACE LAMBDA*******
C*****ALL UNITS HAVE TO BE IN METERS
      RLAMDA=2.997925E10/FGHZ/2.54
C***PARAMETER CONVERSIONS TO PROPER UNITS
55    CONTINUE
      IF(IWL.EQ.2) GO TO 50
      DX=DXIN*CINMTR
      DY=DYIN*CINMTR
      HL=HLIN*CINMTR
      HLS=HLSIN*CINMTR
      ARAD=ARADIN*CINMTR
      HZ=HZIN*CINMTR
      RLSX=RLSXIN*CINMTR
      RLSY=RLSYIN*CINMTR
      RLSZ=RLSZIN*CINMTR
      XN=XNIN*CINMTR
      YN=YNIN*CINMTR
      ZN=ZNIN*CINMTR
      XFOC=XFOCIN*CINMTR
      ZFOC=ZFOCIN*CINMTR
      XL=XLIN*CINMTR
      YL=YLIN*CINMTR
      PCDX=PCDXIN*CINMTR
      RADIUS=CRADIN*CINMTR
50    CONTINUE
      WRITE(8,60)DX,DY,HL,ARAD
60    FORMAT(1X,'DX,DY,HL,ARAD=',2X,4F14.5)
C
C***COMPUTE CALIBRATION CONSTANTS (PHASE ONLY) TO
C***MAXIMIZE GAIN (FOCUS ANTENNA) TO NEAR FIELD RANGE
C
C***PHASE CENTER 'A' VOLTAGE EXCITATION
      XOA=-XL/2.+NXDUM*DX+(NACOLX-1)/2.*DX
      YP=0.0
      IF(ICIRC.EQ.0) WRITE(6,24690)
24690 FORMAT(1X,'CALLING NFDPX2')
      IF(ICIRC.EQ.0)
     2CALL NFDPX2(CTPR,STPR,CTDP,STDP,XFOC,YP,ZFOC,XOA,0,VA,VREFA)
      IF(ICIRC.EQ.1)
     2CALL NFDPC2(XFOC,YP,ZFOC,XOA,0,VA,VREFA)
C***SAVE INCIDENT VOLTAGES
      DO 65 IV=1,NEL
      VTA(IV)=VA(IV)
      WRITE(6,3757)IV,VTA(IV)
      WRITE(8,3757)IV,VTA(IV)
```

```
3757  FORMAT(1X,'IV,VTA=',I4,2X,2E12.4)                           DIP01760
65    CONTINUE                                                   DIP01770
      WRITE(6,9876)VTA(2)                                        DIP01780
9876  FORMAT(1X,'VTA(2)=',2E12.4)                                DIP01790
      NB=NCOLX                                                   DIP01800
      IDMB=NROWY                                                 DIP01810
      IDM=NEL                                                    DIP01820
      ICC=NEL                                                    DIP01830
      IBLT=IBLTSL                                                DIP01840
      IDM1=IDM3                                                  DIP01850
      IF(IBLT.EQ.0) IDM1=NEL                                     DIP01860
      I2=1                                                       DIP01890
8889  CONTINUE                                                   DIP02350
      CALL ZMATRX(CTDP,STDP,IBLTSL,ICC1,ICC,CZ,Z)
C***SOLVE SYSTEM OF EQUATIONS FOR THE UNKNOWN CURRENTS
      IF(IBLT.EQ.1) GO TO 240                                    DIP02370
      ISYM=0                                                     DIP02380
      I12=1                                                      DIP02390
      WRITE(6,9876)VTA(2)                                        DIP02400
      WRITE(6,6110)                                              DIP02410
6110  FORMAT(1X,'CALL CROUT')                                    DIP02420
      CALL CROUT(Z,VA,ICC1,ICC,ISYM,IWR,I12,NEL)                 DIP02430
      I12=2                                                      DIP02440
      WRITE(6,9876)VTA(2)                                        DIP02450
      GO TO 255                                                  DIP02460
240   IENTRY=4                                                   DIP02470
      CALL BLTSOL(CZ,VA,PS,NCOLX,IDMB,IENTRY)                    DIP02480
      IENTRY=3                                                   DIP02490
255   WRITE(8,270)                                               DIP02500
270   FORMAT(1X,'CURRENTS')                                      DIP02510
      IF(NEL.LT.40) CALL CNORM(VA,NEL)                           DIP02520
      WRITE(6,280)                                               DIP02530
280   FORMAT(1X,'AFTER CURRENTS SOLUTION')                       DIP02540
C***VA ARE CURRENTS (AMPERES) NOW                                DIP02550
C*                                                               DIP02560
C***COMPUTE RECEIVED VOLTAGES                                    DIP02570
      DO 285 IC=1,NEL                                            DIP02580
      WRITE(6,7531)ZLOAD                                         DIP02590
7531  FORMAT(1X,'ZLOAD=',F12.5)                                  DIP02600
      WRITE(6,1134)IC,VTA(IC)                                    DIP02610
      VRECVA(IC)=VA(IC)*ZLOAD                                    DIP02620
      WRITE(6,1174)IC,VRECVA(IC)                                 DIP02630
1134  FORMAT(1X,'IC,VTA=',I4,2X,2E12.4)                          DIP02640
1174  FORMAT(1X,'IC,VRECVA=',I4,2X,2E12.4)                       DIP02650
      WRITE(6,1135)IC,VA(IC)                                     DIP02660
1135  FORMAT(1X,'IC,VA=',I4,2X,2E12.4)                           DIP02670
      VRADB=20.*ALOG10(CABS(VRECVA(IC)))                         DIP02680
      VRAPH=ATAN2(AIMAG(VRECVA(IC)),REAL(VRECVA(IC)))/DCR        DIP02690
C***COMPUTE CALIBRATION CONSTANTS (PHASE ONLY)                   DIP02700
      ACALPH(IC)=-VRAPH                                          DIP02710
285   CONTINUE                                                   DIP02720
      DO 7777 IC=1,NEL                                           DIP02730
      WRITE(6,6667)IC,ACALPH(IC)                                 DIP02740
CC    WRITE(8,6667)IC,ACALPH(IC)                                 DIP02750
6667  FORMAT(1X,'IC,ACALPH=',I4,2X,F10.2,' DEGS')                DIP02760
7777  CONTINUE                                                   DIP02770
C***COMPUTE NEAR FIELD PATTERN OF FOCUSED ARRAY                  DIP02780
C                                                                DIP02790
C***COMPUTE BEAMFORMER WEIGHTS (I.E. TAPER)                      DIP02800
```

```
      IF(INUNIF.EQ.0)
      2CALL VRCVWT(NACOLX,NAROWY,NXDUM,NYDUM,EDGTDB,RWTA,RWTB)          DIP02810
      DO 1199 IC=1,NEL                                                 DIP02820
      IF(INUNIF.EQ.1.AND.ISLC.EQ.0) RWTA(IC)=1./SQRT(RNEL)
      WRITE(6,5111)IC,RWTA(IC)
     WRITE(8,5111)IC,RWTA(IC)
5111  FORMAT(1X,'IC,RWTA=',I4,2X,E12.5)
1199  CONTINUE                                                         DIP02850
      DO 5333 KC=1,NEL                                                 DIP02860
      CWTA(KC,1)=RWTA(KC)*CEXP(CJ*ACALPH(KC)*DCR)                      DIP02870
5333  CONTINUE                                                         DIP02880
      IF(IQUAN.EQ.1)                                                   DIP02890
     2CALL ADQUAN(NEL,CWTA,NBMOD,IRNERR,INITRN,ELSGDB,ELSGDG)          DIP02900
      BIGWDB=-299.0
      DO 91020 I=1,NEL
      WRITE(6,61910)I,CWTA(I,1)
      WRITE(8,61910)I,CWTA(I,1)
61910 FORMAT(1X,'I,CWTA(I,1)=',I4,2X,2E12.5)
      CWTADB=20.*DLOG10(CDABS(CWTA(I,1)))
      IF(CWTADB.GT.BIGWDB) BIGWDB=CWTADB
91020 CONTINUE
      DO 91120 I=1,NEL
      CWTADB=20.*DLOG10(CDABS(CWTA(I,1)))-BIGWDB
      CWTADG=DATAN2(DIMAG(CWTA(I,1)),DREAL(CWTA(I,1)))/DCR
      WRITE(6,35990)I,CWTADB,CWTADG
      WRITE(8,35990)I,CWTADB,CWTADG
35990 FORMAT(1X,'I=',I4,2X,'CWTADB,CWTADG=',1X,2F14.5)
91120 CONTINUE
      INITRN=2                                                         DIP02910
C***PERFORM NEAR FIELD SCAN WITH CW RADIATING DIPOLE                    DIP02920
      IF(INEAR.EQ.0) GO TO 390                                         DIP02930
      WRITE(6,350)                                                     DIP02940
      WRITE(6,360)XNIN,YNIN,ZNIN,RLSXIN,RLSYIN,RLSZIN,NCOLXN,NROWYN,   DIP02950
     2NCOLZN                                                           DIP02960
      WRITE(6,320)IWL                                                  DIP02970
320   FORMAT(1X,'IWL=',I4)                                             DIP02980
      WRITE(6,330)                                                     DIP02990
330   FORMAT(1X,'CHANGE NEAR FIELD SCAN PARAMETERS?, ICHANG=1')        DIP03000
      READ(5,*,END=370)ICHANG                                          DIP03010
      IF(ICHANG.EQ.0) GO TO 370                                        DIP03020
340   WRITE(6,350)                                                     DIP03030
350   FORMAT(1X,'XN,YN,ZN,RLSXIN,RLSYIN,RLSZIN,NCOLXN,NROWYN,NCOLZN=') DIP03040
      READ(5,*)XNIN,YNIN,ZNIN,RLSXIN,RLSYIN,RLSZIN,NCOLXN,NROWYN,NCOLZN DIP03050
      WRITE(6,360)XNIN,YNIN,ZNIN,RLSXIN,RLSYIN,RLSZIN,NCOLXN,NROWYN,   DIP03060
     2NCOLZN                                                           DIP03070
360   FORMAT(1X,6F10.3,2X,3I5)                                         DIP03080
      IF(IWL.EQ.2) GO TO 370                                           DIP03090
      XN=XNIN*CINMTR
      YN=YNIN*CINMTR
      ZN=ZNIN*CINMTR
      RLSX=RLSXIN*CINMTR
      RLSY=RLSYIN*CINMTR
      RLSZ=RLSZIN*CINMTR
370   CONTINUE                                                         DIP03160
C**ALL DIMENSIONS IN METERS
      NELN=NCOLXN*NROWYN                                               DIP03180
      NPTSN=NELN*NCOLZN                                                DIP03190
C***SET DEFAULT VALUES FOR DXN,DYN,DZN
      DXN=0.0                                                          DIP03200
```

```
      DYN=0.1                                                           DIP03210
      DZN=0.0                                                           DIP03220
      IF(NCOLXN.GT.1)DXN=RLSX/(NCOLXN-1)                                DIP03230
      IF(NROWYN.GT.1)DYN=RLSY/(NROWYN-1)                                DIP03240
      IF(NCOLZN.GT.1)DZN=RLSZ/(NCOLZN-1)                                DIP03250
     IC=0                                                              DIP03260
      BIGNDB=-299.0                                                     DIP03270
      DO 3000 ICOLZN=1,NCOLZN                                           DIP03280
      ZPOS=ZN+DZN*(ICOLZN-1)                                            DIP03290
      DO 3000 IROWYN=1,NROWYN                                           DIP03300
      Y=YN+DYN*(IROWYN-1)                                               DIP03310
      DO 3000 ICOLXN=1,NCOLXN                                           DIP03320
      CALL GETCP2(CPU2)
      CPUSUB=CPU2-CPU1
      WRITE(6,7319)CPUSUB
7319  FORMAT(1X,'CPU SUBTOTAL=',F14.2)
      IC=IC+1                                                           DIP03330
      X=XN+DXN*(ICOLXN-1)                                               DIP03340
      XC(IC)=X                                                          DIP03350
      YC(IC)=Y                                                          DIP03360
      ZC(IC)=ZPOS                                                       DIP03370
      WRITE(6,6969)IC,XC(IC),YC(IC),ZC(IC)                              DIP03410
6969  FORMAT(1X,'IC,XC,YC,ZC=',I4,2X,3F12.3)                            DIP03420
      IF(ICIRC.EQ.0)
     2CALL NFDPX2(CTPR,STPR,CTDP,STDP,X,Y,ZPOS,0.0,0,VCW,VREFCW)        DIP03430
      IF(ICIRC.EQ.1)
     2CALL NFDPC2(X,Y,ZPOS,0.0,0,VCW,VREFCW)
      IF(IMUT.EQ.0) GO TO 4255                                          DIP03440
C***SOLVE EACH SYSTEM OF EQUATIONS FOR THE UNKNOWN CURRENTS             DIP03450
      IF(IBLT.EQ.1) GO TO 4040                                          DIP03460
      I12=2                                                             DIP03470
      WRITE(6,6110)                                                     DIP03480
      CALL CROUT(Z,VCW,ICC1,ICC,ISYM,IWR,I12,NEL)                       DIP03490
      GO TO 4255                                                        DIP03500
4040  IENTRY=3                                                          DIP03510
      CALL BLTSOL(CZ,VCW,PS,NCOLX,IDMB,IENTRY)                          DIP03520
4255  CONTINUE                                                          DIP03530
C***COMPUTE RECEIVED VOLTAGES FOR PRESENT SCAN                          DIP03540
      DO 3285 IIIC=1,NEL                                                DIP03550
      VRECVX(IIIC)=VCW(IIIC)*ZLOAD                                      DIP03560
3285  CONTINUE                                                          DIP03570
C***STORE AUX. CHANNEL VOLTAGES                                         DIP03580
      DO 1681 IAX=1,NAUX                                                DIP03590
C**MODIFICATION TO INCLUDE AUX. ATTEN.                                  DIP03600
      VAXCWA(IAX,IC)=VRECVX(IAUXA(IAX))*VATENA(IAX)                     DIP03610
CC    WRITE(6,3231)IAX,IC,XC(IC),VAXCWA(IAX,IC)                         DIP03620
3231  FORMAT(1X,'IAX,IC,XC,VAXCWA(IAX,IC)=',2I4,2X,F12.3,2X,2E12.5)     DIP03630
1681  CONTINUE                                                          DIP03640
C***PERFORM BEAM FORMATION                                              DIP03650
      CSUMA=(0.,0.)                                                     DIP03660
      DO 5444 KC=1,NEL                                                  DIP03670
      CSUMA=CSUMA+VRECVX(KC)*CWTA(KC,1)                                 DIP03680
5444  CONTINUE                                                          DIP03690
      VXA=CSUMA                                                         DIP03700
      IF(CABS(VXA).EQ.0.)VXA=(1.E-10,0.)                                DIP03710
      VXADB(IC)=20.*ALOG10(CABS(VXA))                                   DIP03720
      IF(VXADB(IC).GT.BIGNDB) BIGNDB=VXADB(IC)                          DIP03730
      VXAPH(IC)=ATAN2(AIMAG(VXA),REAL(VXA))/DCR                         DIP03740
      VXMANA(IC)=10.**(VXADB(IC)/20.)*CEXP(CJ*VXAPH(IC)*DCR)            DIP03750
CC    WRITE(6,4457)IC,VXMANA(IC)                                        DIP03760
```

```
CC    WRITE(6,4457)IC,VXMANA(IC)                                      DIP03760
4457  FORMAT(1X,'IC,VXMANA=',I4,2X,2E12.5)                            DIP03770
      WRITE(6,6429)IC,XC(IC),VXADB(IC)                                DIP03780
6429  FORMAT(1X,'IC,X,VXADB=',I4,2X,F10.2,2X,F12.2)                   DIP03790
3000  CONTINUE                                                        DIP03800
     IF(IENORM.EQ.0) GO TO 2500                                       DIP03810
      WRITE(6,3765)BIGNDB                                             DIP03820
3765  FORMAT(1X,'BIGNDB=',F12.2)                                      DIP03830
      DO 3020 IC=1,NPTSN                                              DIP03840
CC    WRITE(6,3343)IC,VXADB(IC)                                       DIP03850
      VXADB(IC)=VXADB(IC)-BIGNDB                                      DIP03860
CC    WRITE(6,3343)IC,VXADB(IC)                                       DIP03870
3020  CONTINUE                                                        DIP03880
2500  WRITE(6,3030)                                                   DIP03890
3030  FORMAT(1X,'WANT TO PLOT NEAR FIELD CUTS, IPLOTN=1')             DIP03900
      READ(5,*)IPLOTN                                                 DIP03910
C***NEXT LINE ADDED TO AVOID RUN TIME ERROR                            DIP03920
      IF(IPCONN.EQ.1) IPLOTN=0                                        DIP03930
      IF(IPCONN.EQ.0) THEN
        WRITE(30,18889)
        WRITE(30,18888)XN,NCOLXN,DXN,ZN,NCOLZN,DZN
        WRITE(30,4547)
4547  FORMAT(1X,'IZ,IX,VXADB(IC)')
      ENDIF
      IPP=0
      DO 7788 IZ=1,NCOLZN
      DO 7788 IY=1,NROWYN
      DO 7788 IX=1,NCOLXN
      IPP=IPP+1
CC    DO 7788 IPP=1,NPTSN                                             DIP03950
CC    WRITE(6,3343)IPP,VXADB(IPP)                                     DIP03960
3343  FORMAT(1X,'IPP,VXADB=',I4,2X,F12.2)                             DIP03970
      IF(IPCONN.EQ.0) WRITE(30,*) IZ,IX,VXADB(IPP)
7788  CONTINUE                                                        DIP03980
3040  IF((NCOLXN.GT.1.OR.NROWYN.GT.1).AND.IPLOTN.EQ.1)                DIP03990
     1CALL PLOTTR(NCOLXN,NROWYN,NCOLZN,XC,YC,ZC,NPTSN,VXADB,VXAPH,    DIP04000
     2VXADB,VXAPH)                                                    DIP04010
      IF(NELN.EQ.1.AND.IPLOTN.EQ 1)                                   DIP04020
     1CALL PLOTAX(NPTSN,ZC,VXADB,VXAPH,VXADB,VXAPH)                   DIP04030
      IF(IPCONN.EQ.0) GO TO 3939                                      DIP04040
C***THIS SECTION FOR CONTOUR PLOTS                                     DIP04050
      NCLXN2=NCOLXN+2                                                 DIP04060
      NRWYN2=NROWYN+2                                                 DIP04070
      NCLZN2=NCOLZN+2
      DO 3777 IX=1,NCOLXN                                             DIP04080
      RX(IX)=(XN+DXN*(IX-1))/CINMTR
3777  CONTINUE                                                        DIP04100
      DO 3008 IY=1,NROWYN                                             DIP04110
      RY(IY)=(YN+DYN*(IY-1))/CINMTR
3008  CONTINUE                                                        DIP04130
      DO 3009 IZ=1,NCOLZN
      RZ(IZ)=(ZN+DZN*(IZ-1))/CINMTR
3009  CONTINUE
      IC=0                                                            DIP04140
      IF(IXZ.EQ.1) WRITE(30,18889)
18889 FORMAT(1X,'XN,NCOLXN,DXN,ZN,NCOLZN,DZN=')
      IF(IXZ.EQ.1) WRITE(30,18888)XN,NCOLXN,DXN,ZN,NCOLZN,DZN
18888 FORMAT(1X,E14.5,I5,2X,E14.5,2X,E14.5,I5,2X,E14.5)
      IF(IXZ.EQ.1) WRITE(30,4546)
```

```
4546  FORMAT(1X,'IZ,IX,FM(IZ,IX)')
      DO 3022 IZ=1,NCOLZN
      DO 3022 IY=1,NROWYN                                         DIP04150
      DO 3022 IX=1,NCOLXN                                         DIP04160
      IC=IC+1                                                     DIP04170
     IF(IXY.EQ.1) FM(IY,IX)=VXADB(IC)
      IF(IXZ.EQ.1) FM(IZ,IX)=VXADB(IC)
      IF(IXZ.EQ.1.AND.IPCONN.EQ.1) WRITE(30,*) IZ,IX,FM(IZ,IX)
3022  CONTINUE                                                    DIP04190
      IF(IXY.EQ.1)
     2CALL PLCONT(NCOLXN,NROWYN,NCLXN2,NRWYN2,RX,RY,FM,DMF,-50.,  DIP04200
     310.,5,1)                                                    DIP04210
      IF(IXZ.EQ.1)
     2CALL PLCONT(NCOLXN,NCOLZN,NCLXN2,NCLZN2,RX,RZ,FM,DMF,-50.,
     310.,5,1)
3939  CONTINUE                                                    DIP04220
CC    WRITE(25,*)NCOLXN                                           DIP04470
      DO 1767 IDDD=1,NCOLXN                                       DIP04480
CC    WRITE(25,*)VXADB(IDDD)
1767  CONTINUE                                                    DIP04500
      WRITE(6,3050)                                               DIP04510
3050  FORMAT(1X 'PLOT NEAR FIELD AGAIN?, IPLA=1')                 DIP04520
      READ(5,*)IPLA                                               DIP04530
      IF(IPLA.EQ.1) GO TO 3040                                    DIP04540
390   CONTINUE                                                    DIP04550
      ISTOP=0                                                     DIP04560
      IF(ISTOP.EQ.1) GO TO 9999                                   DIP04570
C***CALL PRNTDA (PRINT PARAMETERS)                                 DIP04580
C370   CALL PRNTDA                                                DIP04590
      IF(IPATRN.EQ.0) GO TO 440                                   DIP04630
410   WRITE(6,420)                                                DIP04640
420   FORMAT(1X,'ISYLMBL FOR PLOTTING, LT. 0 THEN NOT USED')      DIP04650
      READ(5,*)ISYMBL                                             DIP04660
      WRITE(6,430)                                                DIP04700
430   FORMAT(1X,'WANT TO PLOT PATTERNS AGAIN?, IPFNA=1')          DIP04710
      READ(5,*)IPFNA                                              DIP04720
      IF(IPFNA.EQ.1) GO TO 410                                    DIP04730
440   CONTINUE                                                    DIP04740
C***THIS SECTION FOR COVARIANCE MATRIX COMPUTATION                 DIP04750
      IF(NJAMS.EQ.0) GO TO 9999
      DO 7999 ICH=1,NMAX                                          DIP04770
      DO 7999 JCH=1,NMAX                                          DIP04780
      COVNF(ICH,JCH)=DCMPLX(0.0D0,0.0D0)                          DIP04790
7999  CONTINUE                                                    DIP04800
8888  IF(NJAMS.EQ.0) GO TO 4444                                   DIP04810
C***THIS SECTION FOR JAMMER COVARIANCE MATRIX                      DIP04820
      FMINHZ=FCHZ-BWFHZ/2.                                        DIP04830
      DELFHZ=0.0
      IF(NFREQ.GT.1) DELFHZ=BWFHZ/(NFREQ-1)
      DO 600 IFR=1,NFREQ                                          DIP04850
      FHZ=FMINHZ+DELFHZ*(IFR-1)                                   DIP04860
      FGHZ=FHZ/1.0E9                                              DIP04870
C***COMPUTE FREE SPACE WAVELENGTH AT EACH FREQUENCY
      RLAMDA=2.997925E10/FHZ/2.54
C***NOTE: GAMMA= ALPHA +J BETA
C***      AND RLAMDA=2 PI/ BETA   (REF. HAYT PG. 334)
C***THUS NEED TO COMPUTE GAMMA, AND ETA FOR EACH FREQ.
      IF(IWL.EQ.2) GO TO 8789
      DX=DXIN*CINMTR
```

```
      DY=DYIN*CINMTR
      HL=HLIN*CINMTR
      ARAD=ARADIN*CINMTR
      HZ=HZIN*CINMTR
8789  CONTINUE
      CALL ZMATRX(CTDP,STDP,IBLTSL,ICC1,ICC,CZ,Z)                       DIP04940
C***COMPUTE ELEMENT INDUCED VOLTAGES DUE TO JAMMER SOURCES               DIP04950
      CALL VJAMMR(NJAMS,NEL,PWRJDB,ICC1,ICC,PS,CZ,Z,CWTA                DIP04960
     1,NB,IDMB,NAUX,IAUXA,ZLOAD,XJAMIN,YJAMIN,ZJAMIN,                   DIP04970
     2IFR,NFREQ,CTPR,STPR,CTDP,STDP,VMAJMA,VAUXJA)                      DIP04980
      WRITE(6,1234)IFR,VMAJMA(1,IFR),VAUXJA(1,1,IFR)                    DIP04990
      WRITE(8,1234)IFR,VMAJMA(1,IFR),VAUXJA(1,1,IFR)                    DIP05000
1234  FORMAT(1X,'IFR,VMAJMA,VAUXA=',I4,2X,4E11.4)                       DIP05010
600   CONTINUE                                                          DIP05020
C***FORM RECEIVED VOLTAGE MATRIX                                        DIP05030
C***VRECVM(MAIN A, AUX A1, AUX A2,... AUX AN :)
      DO 9001 IJAM=1,NJAMS                                              DIP05050
      DO 9002 IFR=1,NFREQ                                               DIP05060
      IF(ISLC.EQ.1) VCHA(1,IFR)=VMAJMA(IJAM,IFR)
CC    WRITE(6,6789)IJAM,IFR,VMAJMA(IJAM,IFR)                            DIP05080
6789  FORMAT(1X,'IJAM,IFR,VMAJMA(IJAM,IFR)=',2I4,2X,2E12.5)             DIP05090
      DO 9003 IA=1,NAUX                                                 DIP05100
      IAP1=IA+1                                                         DIP05110
      IF(ISLC.EQ.1) VCHA(IAP1,IFR)=VAUXJA(IA,IJAM,IFR)*VATENA(IA)
      IF(ISLC.EQ.0) VCHA(IA,IFR)=VAUXJA(IA,IJAM,IFR)*VATENA(IA)
      WRITE(6,8876)IJAM,IFR,IA,VAUXJA(IA,IJAM,IFR)
      WRITE(8,8876)IJAM,IFR,IA,VAUXJA(IA,IJAM,IFR)
8876  FORMAT(1X,'IJAM,IFR,IA,VAUXJA=',3I4,2X,2E12.4)                    DIP05150
9003  CONTINUE                                                          DIP05160
9002  CONTINUE                                                          DIP05170
      DO 5555 KKK=1,4                                                   DIP05180
      DO 5555 LLL=1,NFREQ                                               DIP05190
CCC   WRITE(6,6655)KKK,LLL,VCHA(KKK,LLL)                                DIP05200
5555  CONTINUE                                                          DIP05210
      WRITE(6,5533)                                                     DIP05220
5533  FORMAT(1X,'NOW COMPUTE JAMMER COVARIANCE MATRIX')                 DIP05230
C***COMPUTE COVARIANCE MATRIX FOR JTH JAMMER SOURCE                     DIP05240
      IF(NFREQ.GT.1) CALL COVSWC(VCHA,VCHA,NMAX,NFREQ,BWFHZ,COVAAJ)
      DO 9005 ICH=1,NMAX
      DO 9005 JCH=1,NMAX
      COVNF(ICH,JCH)=COVNF(ICH,JCH)+COVAAJ(ICH,JCH)                     DIP05280
CC    WRITE(6,2299)ICH,JCH,COVNF(ICH,JCH)                               DIP05290
9005  CONTINUE                                                          DIP05300
9001  CONTINUE                                                          DIP05310
C***ADD RECEIVER NOISE TO DIAGONAL ELEMENTS                             DIP05320
4444  DO 8006 ICH=1,NMAX                                                DIP05330
      COVNF(ICH,ICH)=COVNF(ICH,ICH)+1.0D0                               DIP05340
8006  CONTINUE                                                          DIP05350
      DO 2727 I=1,NMAX                                                  DIP05360
      DO 2727 J=1,NMAX                                                  DIP05370
      IF(CDABS(COVNF(I,J)).EQ.0.0D0) GO TO 2727                         DIP05380
      CNDB=10.*DLOG10(CDABS(COVNF(I,J)))                                DIP05390
      PREALN=DREAL(COVNF(I,J))                                          DIP05400
      PIMAGN=DIMAG(COVNF(I,J))                                          DIP05410
      PHASEN=DATAN2(PIMAGN,PREALN)/DCR                                  DIP05420
      WRITE(8,4411)I,J,CNDB,PHASEN                                      DIP05430
      IF(I.EQ.1)WRITE(6,4411)I,J,CNDB,PHASEN                            DIP05440
4411  FORMAT(1X,'I,J,CNDB,PHASEN=',2I4,2X,2F12.2)                       DIP05450
2727  CONTINUE                                                          DIP05460
```

```
C***COMPUTE COVARIANCE MATRIX INVERSE                                   DIP05470
      DO 8007 ICH=1,NMAX                                                DIP05480
      DO 8007 JCH=1,NMAX                                                DIP05490
      COVNFI(ICH,JCH)=COVNF(ICH,JCH)                                    DIP05500
8007  CONTINUE                                                          DIP05510
      CALL DCMINV(COVNFI,LTMP,MTMP,NMAX,NMAX)                           DIP05520
C***CHECK MATRIX INVERSION ACCURACY  (CINVERSE*C=I)                     DIP05530
      CALL CMMULT(COVNFI,COVNF,NMAX,NMAX,NMAX,CINVCN)                   DIP05540
      DO 8008 ICH=1,NMAX                                                DIP05550
      DO 8008 JCH=1,NMAX                                                DIP05560
      IF(ICH.EQ.1)WRITE(6,8009)ICH,JCH,CINVCN(ICH,JCH)                  DIP05570
8009  FORMAT(1X,'ICH,JCH,CINVCN=',2I4,2X,2E12.5)                        DIP05580
8008  CONTINUE                                                          DIP05590
C***COMPUTE EIGENVALUES (ALSO EIGENVECTORS AND PERFORMANCE INDEX)       DIP05600
      IJOB=2                                                            DIP05610
      CALL EIGCC(COVNF,NMAX,NMAX,IJOB,EIGVAN,EIGVEN,NMAX,WKE,IER)       DIP05620
      WRITE(6,155)IER                                                   DIP05630
155   FORMAT(1X,'AFTER COMPUTE EIGENVALUES, IER=',I5)                   DIP05640
      DO 200 I=1,NMAX                                                   DIP05650
      WRITE(6,300)I,EIGVAN(I)                                           DIP05660
300   FORMAT(1X,'I,EIGVAN=',2X,I4,2E10.3)                               DIP05670
200   CONTINUE                                                          DIP05680
      DO 205 I=1,NMAX                                                   DIP05690
      EVLDBN=10.*DLOG10(CDABS(EIGVAN(I)))                               DIP05710
      WRITE(8,207)I,EVLDBN                                              DIP05720
      WRITE(6,207)I,EVLDBN                                              DIP05730
207   FORMAT(1X,'I,EVLDBN=',I4,2X,2F12.3)                               DIP05740
205   CONTINUE                                                          DIP05750
C***  CALL PRNTDA                                                       DIP05760
C***FILL-IN SIDELOBE CANCELLER QUIESCENT WEIGHTS                        DIP05770
      DO 8985 I=1,NMAX                                                  DIP05780
      WQSLC(I,1)=DCMPLX(0.0D0,0.0D0)                                    DIP05790
8985  CONTINUE                                                          DIP05800
      WQSLC(1,1)=DCMPLX(1.0D0,0.0D0)                                    DIP05810
      DO 7769 I=1,NMAX                                                  DIP05820
      WRITE(6,1212)I,WQSLC(I,1)                                         DIP05830
      WRITE(8,1212)I,WQSLC(I,1)                                         DIP05840
1212  FORMAT(1X,'I,WQSLC=',I4,2X,2E12.5)                                DIP05850
7769  CONTINUE                                                          DIP05860
C***COMPUTE QUIESCENT INR                                               DIP05870
      IF(ISLC.EQ.1) CALL INRTIO(WQSLC,COVNF,NMAX,WTCTR,CMPROD,QINRDB)
      IF(ISLC.EQ.0) CALL INRTIO(CWTA,COVNF,NMAX,WTCTR,CMPROD,QINRDB)
      QINRAA=10.*DLOG10(CDABS(COVNF(1,1)))                              DIP05890
C***COMPUTE AVERAGE CANCELLATION
      SUMC=0.0                                                          DIP05910
      SUMAA=0.0                                                         DIP05920
      DO 1829 IR=1,NRAN                                                 DIP05930
C***ZERO-OUT ADAPTIVE WEIGHTS INITIALLY                                 DIP05940
      DO 57 I=1,NMAX                                                    DIP05950
      WAN(I,1)=DCMPLX(0.0D0,0.0D0)                                      DIP05960
57    CONTINUE                                                          DIP05970
C***COMPUTE ADAPTIVE ARRAY WEIGHTS                                      DIP05980
      IF(ISLC.EQ.1) CALL CMMULT(COVNFI,WQSLC,NMAX,NMAX,1,WAN)
      IF(ISLC.EQ.0) CALL CMMULT(COVNFI,CWTA,NMAX,NMAX,1,WAN)
C***QUANTIZE AND RANDOMIZE ADAPTIVE WEIGHT SETTINGS                     DIP06000
      IF(IQUAN.EQ.1.AND.NBADWT.LT.20)                                   DIP06010
     2CALL ADQUAN(NMAX,WAN,NBADWT,IRNERR,INITRN,AWSGDB,AWSGDG)          DIP06020
C***NORMALIZE FULLY ADAPTIVE WEIGHTS
      IF(ISLC.EQ.0) THEN
```

```
        SAVE=0.0D0
        DO 33345 I=1,NEL
        SAVE=SAVE+CDABS(WAN(I,1))**2
33345   CONTINUE
        SAVESR=DSQRT(SAVE)
        DO 44456 I=1,NEL
        WAN(I,1)=WAN(I,1)/SAVESR
        WANDB=20.*DLOG10(CDABS(WAN(I,1)))
        IF(IR.EQ.1) WRITE(6,3599)I,WANDB
        IF(IR.EQ.1) WRITE(8,3599)I,WANDB
3599    FORMAT(1X,'I=',I4,2X,'WANDB=',1X,F12.3)
44456   CONTINUE
      ENDIF
C****TO PRINT NORMALIZED WEIGHTS                                          DIP06030
      BIGWDB=-299.0                                                       DIP06040
      DO 9102 I=1,NMAX                                                    DIP06050
      IF(IR.EQ.1) WRITE(6,6191)I,WAN(I,1)                                 DIP06100
      IF(IR.EQ.1) WRITE(8,6191)I,WAN(I,1)                                 DIP06110
6191  FORMAT(1X,'I,WAN(I,1)=',I4,2X,2E12.5)
      WANDBM=20.*DLOG10(CDABS(WAN(I,1)))                                  DIP06060
      IF(WANDBM.GT.BIGWDB) BIGWDB=WANDBM                                  DIP06070
9102  CONTINUE                                                            DIP06080
      DO 9112 I=1,NMAX                                                    DIP06090
      WANDB=20.*DLOG10(CDABS(WAN(I,1)))-BIGWDB                            DIP06130
      WANDG=DATAN2(DIMAG(WAN(I,1)),DREAL(WAN(I,1)))/DCR
      IF(IR.EQ.1) WRITE(6,35991)I,WANDB,WANDG
      IF(IR.EQ.1) WRITE(8,35991)I,WANDB,WANDG
35991 FORMAT(1X,'I=',I4,2X,'WANDB,WANDG=',1X,2F14.5)
9112  CONTINUE                                                            DIP06170
C***COMPUTE ADAPTED INR                                                   DIP06180
      CALL INRTIO(WAN,COVNF,NMAX,WTCTR,CMPROD,AINRDB)                     DIP06190
C***COMPUTE CANCELLATION                                                  DIP06200
      CANCDB=AINRDB-QINRDB                                                DIP06210
      WRITE(6,3007)QINRDB,AINRDB,CANCDB                                   DIP06220
      WRITE(8,3007)QINRDB,AINRDB,CANCDB                                   DIP06230
3007  FORMAT(1X,'INR= QUI,ADAP ,CANCEL=',2X,3F10.3,2X,' DB')              DIP06240
      SUMC=SUMC+CANCDB                                                    DIP06250
C***COMPUTE ADAPTED INR CH. A                                             DIP06260
      DO 2255 I=1,NMAX                                                    DIP06270
      WANDMA(I,1)=WAN(I,1)                                                DIP06280
      IF(I.GT.NMAX) WANDMA(I,1)=(0.0D0,0.0D0)                             DIP06290
2255  CONTINUE                                                            DIP06300
      CALL INRTIO(WANDMA,COVNF,NMAX,WTCTR,CMPROD,AINRAA)                  DIP06310
C***CANCELLATION CH. A                                                    DIP06320
      CNCLNA=AINRAA-QINRAA                                                DIP06330
      WRITE(6,3738)QINRAA,AINRAA,CNCLNA                                   DIP06340
      WRITE(8,3738)QINRAA,AINRAA,CNCLNA                                   DIP06350
3738  FORMAT(1X,'SIDELOBE CANCELLER CH. A INR= QUI,ADAP,CAN=',2X,3F10.3)
      SUMAA=SUMAA+CNCLNA                                                  DIP06370
1829  CONTINUE                                                            DIP06380
      AVECAN=SUMC/NRAN                                                    DIP06390
      WRITE(6,4456)CANCDB,NRAN,AVECAN                                     DIP06400
4456  FORMAT(1X,'CANCDB,NRAN,AVECAN=',F12.5,2X,I5,2X,F12.5)               DIP06410
      WRITE(8,4456)CANCDB,NRAN,AVECAN                                     DIP06420
      AVECAA=SUMAA/NRAN                                                   DIP06430
      WRITE(6,2220)AVECAA                                                 DIP06440
      WRITE(8,2220)AVECAA                                                 DIP06450
2220  FORMAT(1X,'AVE. CANEL, CH. A, =',2X,F12.5)                          DIP06460
C***SECTION TO COMPUTE ADAPTIVE ARRAY RADIATION PATTERNS                  DIP06470
```

```
CC    IF(INEAR.EQ.0.OR.IANGLP.EQ.0) GO TO 9990                          DIP06480
      IF(INEAR.EQ.0) GO TO 9990                                         DIP06490
      BIGADB=-299.                                                      DIP06500
      IC=0                                                              DIP06510
      DO 8919 IZ=1,NCOLZN
      DO 8919 IY=1,NROWYN                                               DIP06520
      DO 8919 IX=1,NCOLXN                                               DIP06530
      IC=IC+1                                                           DIP06540
      CSUMA=(0.,0.)                                                     DIP06550
      IF(ISLC.EQ.1) CSUMA=CSUMA+DCONJG(WAN(1,1))*VXMANA(IC)
      DO 7921 IAX=1,NAUX                                                DIP06570
      IAXP1=IAX+1                                                       DIP06580
      IF(ISLC.EQ.1) CSUMA=CSUMA+DCONJG(WAN(IAXP1,1))*VAXCWA(IAX,IC)
      IF(ISLC.EQ.0) CSUMA=CSUMA+DCONJG(WAN(IAX,1))*VAXCWA(IAX,IC)
7921  CONTINUE                                                          DIP06610
      IF(CABS(CSUMA).EQ.0.) CSUMA=(1.E-10,0.)                           DIP06620
      PCHADB(IC)=20.*ALOG10(CABS(CSUMA))                                DIP06630
      IF(PCHADB(IC).GT.BIGADB) BIGADB=PCHADB(IC)                        DIP06650
CC    WRITE(6,4999)IC,PCHADB(IC)                                        DIP06660
8919  CONTINUE                                                          DIP06670
C***NORMALIZE ADAPTIVE PATTERNS                                         DIP06680
      IF(IXZ.EQ.1) WRITE(31,18889)
      IF(IXZ.EQ.1) WRITE(31,18888)XN,NCOLXN,DXN,ZN,NCOLZN,DZN
      IF(IXZ.EQ.1) WRITE(31,4546)
      IC=0                                                              DIP06690
      DO 3459 IZ=1,NCOLZN
      DO 3459 IY=1,NROWYN                                               DIP06700
      DO 3459 IX=1,NCOLXN                                               DIP06710
      IC=IC+1                                                           DIP06720
      PCHADB(IC)=PCHADB(IC)-BIGADB                                      DIP06730
      IF(IPCONN.EQ.0) WRITE(31,*)IZ,IX,PCHADB(IC)
      IF(IPCONN.EQ.0) GO TO 3459                                        DIP06760
      IF(IXY.EQ.1) FM(IY,IX)=PCHADB(IC)
      IF(IXZ.EQ.1) FM(IZ,IX)=PCHADB(IC)
      IF(IXZ.EQ.1.AND.IPCONN.EQ.1) WRITE(31,*) IZ,IX,FM(IZ,IX)
3459  CONTINUE                                                          DIP06780
      WRITE(25,*)NCOLXN                                                 DIP06800
      DO 1879 IDDD=1,NCOLXN                                             DIP06810
      WRITE(25,*)PCHADB(IDDD)
1879  CONTINUE                                                          DIP06830
      IF((NCOLXN.GT.1.OR.NROWYN.GT.1).AND.IPLOTN.EQ.1.AND.IANGLP.EQ.0)  DIP06840
     2CALL PLOTTR(NCOLXN,NROWYN,NCOLZN,XC,YC,ZC,NPTSN,PCHADB,PCHADB,    DIP06850
     3PCHADB,PCHADB)                                                    DIP06860
      IF(IPCONN.EQ.1.AND.IXY.EQ.1)
     2CALL PLCONT(NCOLXN,NROWYN,NCLXN2,NRWYN2,RX,RY,
     3FM,DMF,-50.,10.,5,1)                                              DIP06880
      IF(IPCONN.EQ.1.AND.IXZ.EQ.1)
     2CALL PLCONT(NCOLXN,NCOLZN,NCLXN2,NCLZN2,RX,RZ,
     3FM,DMF,-50.,10.,5,1)
9990  CONTINUE                                                          DIP06890
9999  CONTINUE                                                          DIP06900
      CALL DONEPL                                                       DIP06910
      CALL GETCP2(CPUL)
      CPUTOT=CPUL-CPU1
      WRITE(6,2006)CPUTOT
2006  FORMAT(1X,'TOTAL CPU TIME=',F15.2)
      STOP                                                              DIP06920
      END                                                               DIP06930
C***SUBROUTINE TO COMPUTE IMPEDANCE MATRIX                              DIP07720
```

```
      SUBROUTINE ZMATRX(CTDP,STDP,IBLTSL,ICC1,ICC,CZ,Z)                 DIP07730
      COMPLEX Z(ICC1,ICC),CZ(1)                                         DIP07740
      COMMON/A/DX,DY,NCOLX,NROWY,NEL,HZ,HL,ARAD,ZLOAD,ZCHAR             DIP07750
      COMMON /GROUND/ IGRNDP                                            DIP07760
      COMMON /CIRCLE/ ICIRC,RADIUS,HLS
      NB=NCOLX                                                          DIP07770
      IDMB=NROWY                                                        DIP07780
C***COMPUTE MUTUAL IMPEDANCES Z(1,1),Z(1,2),Z(1,3),...,Z(1,NEL).       DIP07790
CC    CALL RGDZMN(IGRNDP,ICC1,ICC,Z)                                    DIP07800
CC    CALL RGDZAB(CTDP,STDP,IGRNDP,ICC1,ICC,Z)                          DIP07810
      IF(ICIRC.EQ.0) CALL RGDZA2(CTDP,STDP,IGRNDP,ICC1,ICC,Z)
      IF(ICIRC.EQ.1) CALL CADZA2(CTDP,STDP,IGRNDP,ICC1,ICC,Z)
      WRITE(8,10) NEL,NCOLX,NROWY                                       DIP07830
10    FORMAT(1X,'NEL,NCOLX,NROWY=',3I5)                                 DIP07840
      ICOUNT=0                                                          DIP07850
      DO 20 I=1,NCOLX                                                   DIP07860
      DO 20 J=1,NROWY                                                   DIP07870
      ICOUNT=ICOUNT+1                                                   DIP07880
      WRITE(6,80) I,J,Z(1,ICOUNT)                                       DIP07890
80    FORMAT(1X,'I,J,Z(1,ICOUNT)=',2I4,2X,2E12.5)                       DIP07900
20    CONTINUE                                                          DIP07910
      IDM=NEL                                                           DIP07920
      ICC=NEL                                                           DIP07930
C***FILL THE IMPEDANCE MATRIX                                           DIP07940
      IF(NCOLX.LE.1) GO TO 70                                           DIP07950
      IBLT=IBLTSL                                                       DIP07960
      IDM1=IDMB                                                         DIP07970
      IF(IBLT.EQ.0) IDM1=NEL                                            DIP07980
      IF(NROWY.GT.1) GO TO 40                                           DIP07990
C***FILL TOEPLITZ MATRIX                                                DIP08000
      DO 30 I=2,NEL                                                     DIP08010
      DO 30 J=I,NEL                                                     DIP08020
      K=1+J-I                                                           DIP08030
      Z(I,J)=Z(1,K)                                                     DIPC8040
30    CONTINUE                                                          DIP08050
      GO TO 50                                                          DIP08060
40    CALL BTOEPL(IBLT,NB,IDMB,IDM1,IDM,Z)                              DIP08070
50    CONTINUE                                                          DIP08080
      IF(IBLT.EQ.0) GO TO 70                                            DIP08090
      DO 60 I=1,IDMB                                                    DIP08100
      DO 60 J=1,IDM                                                     DIP08110
        IC=(J-1)*IDMB+I                                                 DIP08120
        CZ(IC)=Z(I,J)                                                   DIP08130
CC    WRITE(8,7878)I,J,Z(I,J)                                           DIP08140
7878  FORMAT(1X,'I,J,Z(I,J)=',2I4,2X,2E12.5)                            DIP08150
60    CONTINUE                                                          DIP08160
70    CONTINUE                                                          DIP08170
      RETURN                                                            DIP08180
      END                                                               DIP08190
C***SUBROUTINE TO COMPUTE RECEIVE BEAMFORMER WEIGHTS                    DIP09150
      SUBROUTINE VRCVWT(NACOLX,NAROWY,NXDUM,NYDUM,EDGTDB,WA,WB)         DIP09160
      DIMENSION WA(1),WB(1)                                             DIP09170
      DIMENSION WT(180)                                                 DIP09180
      COMMON/A/DX,DY,NCOLX,NROWY,NEL,HZ,HL,ARAD,ZLOAD,ZCHAR             DIP09200
      COMMON /CHEBY/ ICHEB,SLLDB                                        DIP09210
      PI=3.141592654                                                    DIP09220
      TPI=2.*PI                                                         DIP09230
      DCR=PI/180.                                                       DIP09240
CC    SLLDB=40.                                                         DIP09250
```

```
      TAP=10.**(EDGTDB/20.)
      WRITE(6,*)TAP
      AMP=(1.-TAP)/2.
      NPDX=NCOLX-NACOLX-2*NXDUM
      AXL=DX*(NACOLX-1)
      AYL=DY*(NAROWY-1)
      IF(TAP.NE.1.0)FX=AXL/2.*PI/ACOS(TAP)
      IF(TAP.NE.1.0)FY=AYL/2.*PI/ACOS(TAP)
      WRITE(6,20)
20    FORMAT(1X,'BEFORE CALL CHEBWT')
      WRITE(6,30)SLLDB
30    FORMAT(1X,'SLLDB=',2X,F10.2)
      IF(ICHEB.EQ.1)CALL CHEBWT(NACOLX,SLLDB,WT,RLOSS)
      WRITE(6,40)
40    FORMAT(1X,'AFTER CHEBWT')
      DO 1000 IC=1,NEL
      WA(IC)=0.0
      WB(IC)=0.0
1000  CONTINUE
C***COMPUTE EFFECTIVE DIPOLE CENTER COORDS. FOR BOTH PHASE CENTERS
      XO=-AXL/2.
      YO=-AYL/2.
      IC=0
      DO 80 I=1,NACOLX
      DO 80 J=1,NAROWY
      IC=IC+1
      X=XO+DX*(I-1)
      Y=YO+DY*(J-1)
      TAPERX=1.0
      TAPERY=1.0
      IF(ICHEB.EQ.1) GO TO 70
      IF(TAP.NE.1.0)TAPERX=COS(PI*X/FX)
      IF(TAP.NE.1.0.AND.FY.NE.0.0)TAPERY=COS(PI*Y/FY)
60    TAPER=TAPERX*TAPERY
      WT(IC)=TAPER
70    CONTINUE
80    CONTINUE
C***TRANSFORM FROM SUB-APERTURES TO FULL ARRAY
      IBGNA=NXDUM*NROWY+NYDUM
      IBGNB=IBGNA+NPDX*NROWY
      IC=0
      DO 1010 IX=1,NACOLX
      DO 1010 IY=1,NAROWY
      IC=IC+1
      IA=IBGNA+(IX-1)*NROWY+IY
      IB=IBGNB+(IX-1)*NROWY+IY
      WA(IA)=WT(IC)
      WB(IB)=WT(IC)
1010  CONTINUE
      RETURN
      END
C***SUBROUTINE TO COVARIANCE MATRIX BASED ON NUMERICAL INTEGRATION
C***IN THE FREQUENCY DOMAIN ACCORDING TO SIMPSON'S RULE
      SUBROUTINE COVSWC(VA,VB,NCHAN,NFREQ,BWFHZ,COVAB)
      COMPLEX *16 VA(NCHAN,NFREQ),VB(NCHAN,NFREQ)
      COMPLEX *16 COVAB(NCHAN,NCHAN),CSUM,DCABF
      REAL *8 SWC(101)
      DELTAF=BWFHZ/(NFREQ-1)
      CALL SIMWC(NFREQ,SWC)
```

```
      DO 10 ICH=1,NCHAN                                                 DIP09850
      DO 10 JCH=1,NCHAN                                                 DIP09860
      CSUM=(0.CD0,0.0D0)                                                DIP09870
      DO 20 IFR=1,NFREQ                                                 DIP09880
      DCABF=VA(ICH,IFR)*DCONJG(VB(JCH,IFR))                             DIP09890
      CSUM=CSUM+DCABF*SWC(IFR)                                          DIP09900
CC    WRITE(6,6767)IFR,CSUM                                             DIP09910
6767  FORMAT(1X,'IFR,CSUM=',I4,2X,2E12.5)                               DIP09920
20    CONTINUE                                                          DIP09930
      COVAB(ICH,JCH)=DELTAF/2.*CSUM                                     DIP09940
C***NEW LINE TO NORMALIZE COVAB                                         DIP09950
      COVAB(ICH,JCH)=COVAB(ICH,JCH)/BWFHZ                               DIP09960
CC    WRITE(6,4567)ICH,JCH,COVAB(ICH,JCH)                               DIP09970
4567  FORMAT(1X,'ICH,JCH,COVAB(ICH,JCH)=',2I4,2X,2E12.5)                DIP09980
10    CONTINUE                                                          DIP09990
      RETURN                                                            DIP10000
      END                                                               DIP10010
C***SUBROUTINE TO GENERATE SIMPSON'S 1/3 RULE WEIGHTING COEF.           DIP10020
C***INTGERAL F(X)DX=(DELTAX/3.)*(F(1)+4*F(2)+2*F(3)+4*F(4)+ ..+F(ODD))  DIP10030
C*** THE SERIES 1 4 2 4 2 4 .....1 ARE SIMPSON'S COEF.                  DIP10040
      SUBROUTINE SIMWC(NCOEF,SWC)                                       DIP10050
      REAL *8 SWC(NCOEF)                                                DIP10060
      DO 10 N=1,NCOEF                                                   DIP10070
      XNN=FLOAT(N)                                                      DIP10080
      NN=N/2                                                            DIP10090
      TT=XNN/2.                                                         DIP10100
      DIF=TT-FLOAT(NN)                                                  DIP10110
      NC=2                                                              DIP10120
      IF(DIF.EQ.0.) NC=4                                                DIP10130
      IF(N.EQ.1.OR.N.EQ.NCOEF) NC=1                                     DIP10140
      SWC(N)=NC                                                         DIP10150
10    CONTINUE                                                          DIP10160
      RETURN                                                            DIP10170
      END                                                               DIP10180
C***SUBROUTINE TO COMPUTE RECEIVED VOLTAGES DUE TO JAMMER SOURCES       DIP10190
      SUBROUTINE VJAMMR(NJM,NEL,PWRJDB,ICC1,ICC,PS,CZ,Z,CWTA            DIP10200
     1,NB,IDMB,NAUX,IAUXA,ZLOAD,XJAMIN,YJAMIN,ZJAMIN,                   DIP10210
     2IFR,NFR,CTPR,STPR,CTDP,STDP,VMAINA,VAUXA)                         DIP10220
      COMPLEX *16 CWTA(NEL,1)                                           DIP10230
      COMPLEX PS(1),CZ(1),Z(ICC1,ICC)                                   DIP10240
      COMPLEX VJM(180),VREFJM                                           DIP10250
      COMPLEX CJ,CSUMA                                                  DIP10260
      COMPLEX *16 VMAINA(NJM,NFR)                                       DIP10270
      COMPLEX *16 VAUXA(NAUX,NJM,NFR)                                   DIP10280
      DIMENSION PWRJDB(1),PWRJ(10)                                      DIP10290
      DIMENSION XJAMIN(1),YJAMIN(1),ZJAMIN(1)                           DIP10300
      INTEGER IAUXA(1)                                                  DIP10310
      COMMON/PCENTR/PCDXIN                                              DIP10320
      COMMON /B/ NGEN,IGEN,THETAS,PHIS,IMUT,IBLTSL,IPATRN               DIP10330
      COMMON /D/ FGHZ,RLAMDA,IWL,IS,NSCANS                              DIP10340
      COMMON /WRITE/ IWR                                                DIP10350
      COMMON /CIRCLE/ ICIRC,RADIUS,HLS
      PI=3.141592654                                                    DIP10360
      DCR=PI/180.                                                       DIP10370
      CJ=(0.,1.)                                                        DIP10380
      CINMTR=0.0254
      ISYM=0                                                            DIP10390
C***CONVERT DB TO POWER (RELATIVE TO NOISE)                             DIP10400
      DO 10 II=1,NJM                                                    DIP10410
```

```
      PWRJ(II)=10.**(PWRJDB(II)/10.)                                    DIP10420
      WRITE(6,66)II,PWRJ(II)                                            DIP10430
      WRITE(8,66)II,PWRJ(II)
66    FORMAT(1X,'II,PWRJ IN POWER=',I4,2X,E12.5)                        DIP10440
10    CONTINUE                                                          DIP10450
      IF(IFR.GT.1) GO TO 55                                             DIP10460
      WRITE(6,20)IWL                                                    DIP10470
      DO 5544 IJAM=1,NJM                                                DIP10480
      WRITE(6 40)                                                       DIP10490
      WRITE(6,50)XJAMIN(IJAM),YJAMIN(IJAM),ZJAMIN(IJAM)                 DIP10500
5544  CONTINUE                                                          DIP10510
20    FORMAT(1X,'IWL=',I4)                                              DIP10520
      WRITE(6,30)                                                       DIP10530
30     FORMAT(1X,'CHANGE NEAR FIELD JAMMER POSITIONS (INCHES)?, ICH=1')
      READ(5,*)ICH                                                      DIP10550
      IF(ICH.EQ.0) GO TO 55                                             DIP10560
      DO 8887 IJAM=1,NJM                                                DIP10570
       WRITE(6,40)                                                      DIP10580
40     FORMAT(1X,'XJAMIN,YJAMIN,ZJAMIN =')                              DIP10590
      READ(5,*,END=55)XJAMIN(IJAM),YJAMIN(IJAM),ZJAMIN(IJAM)            DIP10600
      WRITE(6,50)XJAMIN(IJAM),YJAMIN(IJAM),ZJAMIN(IJAM)                 DIP10610
50     FORMAT(1X,5F10.3,2X,2I5)                                         DIP10620
8887  CONTINUE                                                          DIP10630
55    CONTINUE                                                          DIP10640
      PCDX=PCDXIN*CINMTR
      WRITE(6,4757)IFR                                                  DIP10660
4757  FORMAT(1X,'FREQ. INDEX, IFR=',I4)                                 DIP10670
60    CONTINUE                                                          DIP10680
C**ALL DIMENSIONS IN METERS
      DO 180 IPHACN=1,1                                                 DIP10700
      WRITE(6,2223)IPHACN                                               DIP10710
2223  FORMAT(1X,'IPHACN=',I4)                                           DIP10720
      XREFDP=-PCDX/2.+(IPHACN-1)*PCDX                                   DIP10730
C***PERFORM JAMMER SOURCE SCAN                                          DIP10740
      DO 180 IJAM=1,NJM                                                 DIP10750
      X=XJAMIN(IJAM)*CINMTR+PCDX*(IPHACN-1)
      Y=YJAMIN(IJAM)*CINMTR
      ZPOS=ZJAMIN(IJAM)*CINMTR
      WRITE(6,6688)X,Y,ZPOS,XREFDP                                      DIP10790
6688  FORMAT(1X,'X,Y,ZPOS,XREFDP (METERS)=',4F12.4)
      IF(ICIRC.EQ.0)
     2CALL NFDPX2(CTPR,STPR,CTDP,STDP,X,Y,ZPOS,XREFDP,1,VJM,VREFJM)
      IF(ICIRC.EQ.1)
     2CALL NFDPC2(X,Y,ZPOS,XREFDP,1,VJM,VREFJM)
C***NORMALIZE INCIDENT JAMMER POWER                                     DIP10820
      DO 70 INORM=1,NEL                                                 DIP10830
      VJM(INORM)=VJM(INORM)/VREFJM*SQRT(PWRJ(IJAM))                     DIP10840
      WRITE(6,77)INORM,VJM(INORM)                                       DIP10850
77    FORMAT(1X,'INORM,VJM(INORM) VOLTAGE=',I4,2X,2E12.5)               DIP10860
70    CONTINUE                                                          DIP10870
      IF(IMUT.EQ.0) GO TO 90                                            DIP10880
C***SOLVE EACH SYSTEM OF EQUATIONS FOR THE UNKNOWN CURRENTS             DIP10890
      IF(IBLTSL.EQ.1) GO TO 80                                          DIP10900
      I12=1                                                             DIP10910
      IF(IJAM.GT.1.OR.IPHACN.GT.1) I12=2                                DIP10920
      WRITE(6,6110)                                                     DIP10930
6110  FORMAT(1X,'CALL CROUT IN VJAMMER')                                DIP10940
      CALL CROUT(Z,VJM,ICC1,ICC,ISYM,IWR,I12,NEL)                       DIP10950
      GO TO 90                                                          DIP10960
```

```
80    IENTRY=4                                                           DIP10970
      IF(IJAM.GT.1.OR.IPHACN.GT.1) IENTRY=3                              DIP10980
      CALL BLTSOL(CZ,VJM,PS.NB,IDMB,IENTRY)                              DIP10990
      IF(NEL.LT.40) CALL CNORM(VJM,NEL)                                  DIP11000
90    CONTINUE                                                           DIP11010
C***COMPUTE RECEIVED VOLTAGES FOR PRESENT SOURCE POSITION                 DIP11020
      DO 100 IEL=1,NEL                                                   DIP11030
      VJM(IEL)=VJM(IEL)*ZLOAD                                            DIP11040
CC    WRITE(6,4456)IEL,VJM(IEL)                                          DIP11050
4456  FORMAT(1X,'IEL,VJM(IEL) RECEIVED VOLT.=',I4,2X,2E12.5)             DIP11060
100   CONTINUE                                                           DIP11070
C***PERFORM BEAM FORMATION FOR MAIN A                                     DIP11080
C***PHASE CENTER A                                                        DIP11090
      CSUMA=(0.0D0,0.0D0)                                                DIP11100
      DO 110 KC=1,NEL                                                    DIP11110
      CSUMA=CSUMA+VJM(KC)*CWTA(KC,1)                                     DIP11120
110   CONTINUE                                                           DIP11130
      VMAINA(IJAM,IFR)=CSUMA                                             DIP11140
      WRITE(6,2222)IJAM,IFR,VMAINA(IJAM,IFR)                             DIP11150
2222  FORMAT(1X,'IJAM,IFR,VMAINA= (AFTER B.F.)',2I4,2X,2E12.5)           DIP11160
C***COMPUTE AUXILIARY CHANNEL VOLTAGES                                    DIP11170
      DO 7000 IAUX=1,NAUX                                                DIP11180
      VAUXA(IAUX,IJAM,IFR)=VJM(IAUXA(IAUX))                              DIP11190
7000  CONTINUE                                                           DIP11200
180   CONTINUE                                                           DIP11210
      DO 9000 IJAM=1,NJAMS                                               DIP11220
CCC   WRITE(6,4433)IJAM,IFR,VMAINA(IJAM,IFR)                             DIP11230
4433  FORMAT(1X,'VJAMMR:IJAM,IFR,VMAINA=',2I4,2X,2E12.5)                 DIP11240
      DO 9001 IAUX=1,NAUX                                                DIP11250
      WRITE(6,3333)IAUX,IJAM,IFR,VAUXA(IAUX,IJAM,IFR)                    DIP11260
3333  FORMAT(1X,'VJAMMR:IAUX,IJAM,IFR,VAUXA=',3I4,2X,2E12.5)             DIP11270
9001  CONTINUE                                                           DIP11280
9000  CONTINUE                                                           DIP11290
      RETURN                                                             DIP11300
      END                                                                DIP11310
C***SUBROUTINE TO COMPUTE INTERFERENCE TO NOISE RATIO                     DIP11320
      SUBROUTINE INRTIO(WT,COV,NEL,WTCTR,CMPROD,DBINR)                   DIP11330
      IMPLICIT REAL *8 (A-H,O-Z)                                         DIP11340
      COMPLEX *16 WT(NEL,1),COV(NEL,NEL),WTCTR(1,NEL)                    DIP11350
      COMPLEX *16 CMPROD(1,NEL),CPROD1(1,1),CPROD2(1,1),CINR             DIP11360
      COMMON /NORMAL/ INRNOR
      WRITE(12,2222)                                                     DIP11370
2222  FORMAT(1X,'INSIDE INRTIO SUBROUTINE')                              DIP11380
      DO 110 I=1,NEL                                                     DIP11390
CC    WRITE(12,445)I,WT(I,1)                                             DIP11400
445   FORMAT(1X,'I,WT(I,1)=',2X,I4,2X,2E12.5)                            DIP11410
110   CONTINUE                                                           DIP11420
      CALL CONJTR(WT,NEL,1,WTCTR)                                        DIP11430
      DO 111 I=1,NEL                                                     DIP11440
CC    WRITE(12,666)I,WT(I,1),WTCTR(1,I)                                  DIP11450
666   FORMAT(1X,'I,WT,WTCTR=',2X,I4,2X,4E12.5)                           DIP11460
111   CONTINUE                                                           DIP11470
      CALL CMMULT(WTCTR,COV,1,NEL,NEL,CMPROD)                            DIP11480
      CALL CMMULT(CMPROD,WT,1,NEL,1,CPROD1)                              DIP11490
      CALL CMMULT(WTCTR,WT,1,NEL,1,CPROD2)                               DIP11500
      WRITE(6,333)CPROD1(1,1),CPROD2(1,1)
      WRITE(8,333)CPROD1(1,1),CPROD2(1,1)
333   FORMAT(1X,'CPROD1,CPROD2=',4E12.5)                                 DIP11520
C***INR NORMALIZED                                                        DIP11530
```

```
CC    INRNOR=0
      WRITE(6,7739)INRNOR                                             DIP11550
      WRITE(8,7739)INRNOR                                             DIP11560
7739  FORMAT(1X,'INR NORMALIZATION PARAMETER, INRNOR=',I4)            DIP11570
      IF(INRNOR.EQ.1) CINR=CPROD1(1,1)/CPROD2(1,1)                    DIP11580
C***INR NOT NORMALIZED FOR INRNOR=0
      IF(INRNOR.EQ.0) CINR=CPROD1(1,1)                                DIP11600
      DBINR=10.*DLOG10(CDABS(CINR))                                   DIP11610
      RETURN                                                          DIP11620
      END                                                             DIP11630
      SUBROUTINE CMMULT(A,B,L,M,N,C)                                  DIP11640
      COMPLEX *16 A(L,M),B(M,N),C(L,N)                                DIP11650
      DO 20 I=1,L                                                     DIP11660
      DO 20 J=1,N                                                     DIP11670
      C(I,J)=DCMPLX(0.0D0,0.0D0)                                      DIP11680
      DO 20 K=1,M                                                     DIP11690
      C(I,J)=C(I,J)+(A(I,K)*B(K,J))                                   DIP11700
 20   CONTINUE                                                        DIP11710
      RETURN                                                          DIP11720
      END                                                             DIP11730
C***SUBROUTINE TO COMPUTE CONJUGATE TRANSPOSE OF A MATRIX              DIP11740
      SUBROUTINE CONJTR(A,L,M,ACTR)                                   DIP11750
      COMPLEX *16 A(L,M),ACTR(M,L)                                    DIP11760
      DO 10 I=1,M                                                     DIP11770
      DO 10 J=1,L                                                     DIP11780
      ACTR(I,J)=DCONJG(A(J,I))                                        DIP11790
CC    WRITE(12,7766)I,J,A(J,I),ACTR(I,J)                              DIP11800
7766  FORMAT(1X,'I,J,A(J,I),ACTR(I,J)=',2X,2I4,2X,4E12.5)             DIP11810
10    CONTINUE                                                        DIP11820
      RETURN                                                          DIP11830
      END                                                             DIP11840
      SUBROUTINE DCMINV(A,L,M,IDM,NEQ)                                DIP11850
      COMPLEX *16 A(IDM,IDM),BIGA,HOLD,DET                            DIP11860
      INTEGER L(IDM),M(IDM)                                           DIP11870
      N=NEQ                                                           DIP11880
      DET=DCMPLX(1.0D0,0.0D0)                                         DIP11890
      DO 80 K=1,N                                                     DIP11900
      L(K)=K                                                          DIP11910
      M(K)=K                                                          DIP11920
      BIGA=A(K,K)                                                     DIP11930
      DO 20 J=K,N                                                     DIP11940
      DO 20 I=K,N                                                     DIP11950
10    IF(CDABS(BIGA)-CDABS(A(I,J)))15,19,19                           DIP11960
15    BIGA=A(I,J)                                                     DIP11970
      L(K)=I                                                          DIP11980
      M(K)=J                                                          DIP11990
19    CONTINUE                                                        DIP12000
20    CONTINUE                                                        DIP12010
      J=L(K)                                                          DIP12020
      IF(J-K)35,35,25                                                 DIP12030
25    CONTINUE                                                        DIP12040
      DO 30 I=1,N                                                     DIP12050
      HOLD=-A(K,I)                                                    DIP12060
      A(K,I)=A(J,I)                                                   DIP12070
30    A(J,I)=HOLD                                                     DIP12080
35    I=M(K)                                                          DIP12090
      IF(I-K) 45,45,38                                                DIP12100
38    CONTINUE                                                        DIP12110
      DO 40 J=1,N                                                     DIP12120
```

```
      HOLD=-A(J,K)                                                      DIP12130
      A(J,K)=A(J,I)                                                     DIP12140
40    A(J,I)=HOLD                                                       DIP12150
45    CONTINUE                                                          DIP12160
      DO 55 I=1,N                                                       DIP12170
      IF(I-K)50,55,50                                                   DIP12180
50    A(I,K)=A(I,K)/(-BIGA)                                             DIP12190
55    CONTINUE                                                          DIP12200
      DO 65 I=1,N                                                       DIP12210
      DO 65 J=1,N                                                       DIP12220
      IF(I-K)60,64,60                                                   DIP12230
60    IF(J-K)62,64,62                                                   DIP12240
62    A(I,J)=A(I,K)*A(K,J)+A(I,J)                                       DIP12250
64    CONTINUE                                                          DIP12260
65    CONTINUE                                                          DIP12270
      DO 75 J=1,N                                                       DIP12280
      IF(J-K)70,75,70                                                   DIP12290
70    A(K,J)=A(K,J)/BIGA                                                DIP12300
75    CONTINUE                                                          DIP12310
      DET=DET*BIGA                                                      DIP12320
      A(K,K)=1.0D0/BIGA                                                 DIP12330
80    CONTINUE                                                          DIP12340
      K=N                                                               DIP12350
100   K=K-1                                                             DIP12360
      IF(K)150,150,105                                                  DIP12370
105   I=L(K)                                                            DIP12380
      IF(I-K)120,120,108                                                DIP12390
108   CONTINUE                                                          DIP12400
      DO 110 J=1,N                                                      DIP12410
      HOLD=A(J,K)                                                       DIP12420
      A(J,K)=-A(J,I)                                                    DIP12430
110   A(J,I)=HOLD                                                       DIP12440
120   J=M(K)                                                            DIP12450
      IF(J-K)100,100,125                                                DIP12460
125   CONTINUE                                                          DIP12470
      DO 130 I=1,N                                                      DIP12480
      HOLD=A(K,I)                                                       DIP12490
      A(K,I)=-A(J,I)                                                    DIP12500
130   A(J,I)=HOLD                                                       DIP12510
      GO TO 100                                                         DIP12520
150   RETURN                                                            DIP12530
      END                                                               DIP12540
C***SUBROUTINE CNORM                                                    DIP12550
C***COMPUTES A NORMALIZED COMPLEX COLUMN VECTOR SUCH THAT THE           DIP12560
C***MAXIMUM ELEMENT HAS UNITY MAGNITUDE.                                DIP12570
C                                                                       DIP12580
C***PRINTS MAGNITUDE AND PHASE OF NORMALIZED VECTOR                     DIP12590
C                                                                       DIP12600
      SUBROUTINE CNORM(V,N)                                             DIP12610
C***    V IS THE INPUT COMPLEX COLUMN VECTOR                            DIP12620
C***    N IS THE LENGTH OF V                                            DIP12630
      COMPLEX V(1),SS                                                   DIP12640
      CNOR=0.0                                                          DIP12650
      DO 10 K=1,N                                                       DIP12660
      SA=CABS(V(K))                                                     DIP12670
      IF(SA.GT.CNOR) CNOR=SA                                            DIP12680
10    CONTINUE                                                          DIP12690
      IF(CNOR.LE.0.) CNOR=1.0                                           DIP12700
      DO 30 K=1,N                                                       DIP12710
```

```
      SS=V(K)                                                           DIP12720
      SA=CABS(SS)                                                       DIP12730
      SNOR=SA/CNOR                                                      DIP12740
      PHR=0.                                                            DIP12750
      IF(SA.GT.0.) PHR=ATAN2(AIMAG(SS),REAL(SS))                        DIP12760
      PH=57.29578*PHR                                                   DIP12770
      WRITE(8,20) K,SNOR,SA,PH                                          DIP12780
20    FORMAT(1X,I5,F10.6,3X,E15.3,F10.0)                                DIP12790
30    CONTINUE                                                          DIP12800
      RETURN                                                            DIP12810
      END                                                               DIP12820
      SUBROUTINE GETCP2(RCPU)
      REAL ETIME,TARRAY(2),RCPU
      TIME=ETIME(TARRAY)
      RCPU=TARRAY(1)
      RETURN
      END
C***SUBROUTINE TO COMPUTE INDUCED VOLTAGE BETWEEN PROBE AND              YDI00010
C***DIPOLE ARRAY ELEMENTS
C***RING ARRAY
      SUBROUTINE NFDPC2(XP,YP,ZP,XREF,IREF,V,VREF)
C***DISTANCES ARE IN METERS
      COMPLEX V(1),VD,VR1,VREF
      COMPLEX *16 DZABG                                                 YDI00060
      COMMON/A/DX,DY,NCOLX,NROWY,NEL,HZ,HL,ARAD,ZLOAD,ZCHAR             YDI00070
      COMMON /CIRCLE/ ICIRC,RADIUS,HLS
C***NOTE: AA,BB,... FOR PROBE    11,22,... FOR DIPOLE                   YDI00120
      PI=3.1415926535
      DCR=PI/180.
C***PROBE DIMENSIONS
      XAA=XP                                                            YDI00260
      XBB=XP                                                            YDI00270
      XCC=XP                                                            YDI00280
      YAA=YP-HLS
      YBB=YP                                                            YDI00300
      YCC=YP+HLS
785   ZAA=ZP
      ZBB=ZP                                                            YDI00330
      ZCC=ZP
      IC=0                                                              YDI00430
C****CIRCULAR DIPOLE ARRAY ELEMENTS
      DELPHI=360./NEL
      DO 10 IX=1,NEL
      PHI=DELPHI*(IX-1)
      XD=RADIUS*COS(PHI*DCR)
      ZD=RADIUS*SIN(PHI*DCR)
      DIST=SQRT((XP-XD)**2+(ZP-ZD)**2)
      IF(DIST.LT.ARAD) XD=XD+ARAD
      X11=XD
      X22=XD                                                            YDI00490
      X33=XD
      Y11=YP-HL
      Y22=YP
      Y33=YP+HL
      Z11=ZD
      Z22=ZD
      Z33=ZD
9446  IC=IC+1                                                           YDI00650
CC    WRITE(6,2233)IX,IY,XD,YD                                          YDI00660
```

```
2233  FORMAT(1X,'IX,IY,XD,YD=',2X,2I4,2X,2F12.5)                   YDI00670
CC    WRITE(6,7854)XAA,XBB,XCC,YAA,YBB,YCC,ZAA,ZBB,ZCC
7854  FORMAT(1X,'XYZABC=',9E12.5)
CC    WRITE(6,7855)X11,X22,X33,Y11,Y22,Y33,Z11,Z22,Z33
7855  FORMAT(1X,'XYZ123=',9E12.5)
      CALL DSZABG(XAA,XBB,XCC,YAA,YBB,YCC,ZAA,ZBB,ZCC,X11,X22,X33,   YDI00680
     2Y11,Y22,Y33,Z11,Z22,Z33,DZABG)                                  YDI00690
      VD=DZABG                                                        YDI00700
CC    WRITE(6,33)IC,VD
33    FORMAT(1X,'IC,VD=',2X,I5,2E12.5)
      V(IC)=VD
      VAMPDB=20.*ALOG10(CABS(V(IC)))                                  YDI00780
      VPHASE=ATAN2(AIMAG(V(IC)),REAL(V(IC)))*180./3.141592654         YDI00790
CC    WRITE(6,4455)IC,VAMPDB,VPHASE                                   YDI00800
4455  FORMAT(1X,'IC,VAMDB,VPHASE=',2X,I4,2X,2F12.2)                   YDI00810
10    CONTINUE                                                        YDI00820
C***COMPUTE REFERENCE VOLTAGE (FICTITIOUS ELEMENT AT XREF, Y=0)       YDI00830
C***SET XREF=RADIUS, THIS MAKES ELEMENT 1 THE REFERENCE
      XREF=RADIUS
C***SET XREF=ARAD, (REF. CLOSE TO THE ORIGIN)
      XREF=ARAD
      X11=XREF
      X22=XREF                                                        YDI00870
      X33=XREF
      Y11=-HL
      Y22=0.0                                                         YDI00980
      Y33=HL
      Z11=0.0
      Z22=0.0
      Z33=0.0
9267  CALL DSZABG(XAA,XBB,XCC,YAA,YBB,YCC,ZAA,ZBB,ZCC,X11,X22,X33,   YDI01000
     2Y11,Y22,Y33,Z11,Z22,Z33,DZABG)                                  YDI01010
      VR1=DZABG                                                       YDI01020
      VREF=VR1
99    RETURN                                                          YDI01080
      END                                                             YDI01090
C***SUBROUTINE TO COMPUTE MUT. IMPED. BETWEEN STRAIGHT DIPOLES
C***ARRANGED IN A RING ARRAY (CIRCLE)
      SUBROUTINE CADZA2(CTDP,STDP,IGRNDP,ICC1,ICC,Z)
      COMPLEX ZMA,ZABG,Z(ICC1,ICC)
      COMMON /A/ DX,DY,NCOLX,NROWY,NEL,HZ,HL,ARAD,ZLOAD,ZCHAR
      COMMON /CIRCLE/ ICIRC,RADIUS,HLS
C***ALL DIMENSIONS IN METERS
      PI=3.1415926535
      DCR=PI/180.
C***FIXED POSITION FOR ELEMENT 1
      XAA=RADIUS
      XBB=RADIUS
      XCC=RADIUS
      YAA=-HL
      YBB=0.0
      YCC=HL
      ZAA=0.0
      ZBB=0.0
      ZCC=0.0
      Y11=YAA
      Y22=YBB
      Y33=YCC
```

```
C***VARIABLE POSITION FOR RING ARRAY ELEMENTS
      DELPHI=360./NEL
      IC=0
      DO 20 I=1,NEL
      IC=IC+1
      PHI=DELPHI*(I-1)*DCR
      XD=RADIUS*COS(PHI)
      ZD=RADIUS*SIN(PHI)
      X11=XD
      X22=XD
      X33=XD
      IF(I.EQ.1) X11=X11+ARAD
      IF(I.EQ.1) X22=X22+ARAD
      IF(I.EQ.1) X33=X33+ARAD
      Z11=ZD
      Z22=ZD
      Z33=ZD
CC    WRITE(6,87)IC
87    FORMAT(1X,'IC=',I5)
CC    WRITE(6,88)XAA,XBB,XCC,YAA,YBB,YCC,ZAA,ZBB,ZCC
88    FORMAT(1X,'XYZABC=',9E12.4)
CC    WRITE(6,89)X11,X22,X33,Y11,Y22,Y33,Z11,Z22,Z33
89    FORMAT(1X,'XYZ123=',9E12.4)
      ZMA=ZABG(XA',XBB,XCC,YAA,YBB,YCC,ZAA,ZBB,ZCC,X11,X22,X33,
     2Y11,Y22,Y33,Z11,Z22,Z33)
      Z(1,IC)=ZMA
      IF(IC.EQ.1) Z(1,1)=Z(1,1)+ZLOAD
20    CONTINUE
      DO 40 I=1,NEL
      IF(I.LT.9)WRITE(6,30)I,Z(1,I)
      WRITE(8,30)I,Z(1,I)
30    FORMAT(1X,'Z(1,',I4,')=',2E12.5)
40    CONTINUE
      RETURN
      END
*********file zabgenloss.f*********************
C***PROGRAM TO CALCULATE MUTUAL IMPEDANCE BETWEEN TWO DIPOLES            ZAB00010
C***WITH ARBITRARY LENGTH AND ORIENTATION.  A PIECEWISE-                  ZAB00020
C***SINUSOIDAL CURRENT DISTRIBUTION IS ASSUMED.                          ZAB00030
      COMPLEX FUNCTION ZABG(X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,XA,XB,XC,YA,YB,YCZAB00040
     2,ZA,ZB,ZC)                                                         ZAB00050
      COMPLEX P11,P12,P21,P22,Q11,Q12,Q21,Q22,R11,R12,R21,R22            ZAB00060
      COMPLEX S11,S12,S21,S22,JCOM,GAM,CGDS,SGDS,SGDT,ETA,EP3            ZAB00070
      COMPLEX EGDS,EGDT
      COMMON /F/ FHZ,ER3,SIG3,TD3
C***ALL DIMENSIONS IN METERS
      PI=3.141592654
      TPI=2.*PI
      B=TPI
      JCOM=(0.,1.)                                                       ZAB00090
      E0=8.854E-12
      U0=1.2566E 5
      OMEGA=TPI*FHZ                                                      WIR01670
      IF(SIG3.LT.0.)EP3=ER3*E0*CMPLX(1.,-TD3)                            WIR01720
      IF(TD3.LT.0.)EP3=CMPLX(ER3*E0,-SIG3/OMEGA)                         WIR01730
      ETA=CSQRT(U0/EP3)                                                  WIR01740
      GAM=OMEGA*CSQRT(-U0*EP3)                                           WIR01750
```

```
      AM=0.0001
      IF(CABS(GAM*AM).GT.0.06) WRITE(6,7923)AM
7923  FORMAT(1X,'CABS(GAM*AM) IS GREATER THAN 0.06, AM=',E14.5)
      INT=0                                                            ZAB00130
      XBA=XB-XA                                                        ZAB00140
      YBA=YB-YA                                                        ZAB00150
      ZBA=ZB-ZA                                                        ZAB00160
      X21=X2-X1                                                        ZAB00170
      Y21=Y2-Y1                                                        ZAB00180
      Z21=Z2-Z1                                                        ZAB00190
      DS=SQRT(XBA*XBA+YBA*YBA+ZBA*ZBA)                                 ZAB00200
      DT=SQRT(X21*X21+Y21*Y21+Z21*Z21)                                 ZAB00210
CC    DSK=B*DS                                                         ZAB00220
CC    DTK=B*DT                                                         ZAB00230
CC    CGDS=CMPLX(COS(DSK),0.0)                                         ZAB00240
CC    SGDS=CMPLX(0.0,SIN(DSK))                                         ZAB00250
CC    SGDT=CMPLX(0.0,SIN(DTK))                                         ZAB00260
C***FOR LOSSY MEDIUM THE NEXT LINES ARE APPROPRIATE
      EGDS=CEXP(GAM*DS)
      EGDT=CEXP(GAM*DT)
      CGDS=(EGDS+1./EGDS)/2.
      SGDS=(EGDS-1./EGDT)/2.
      SGDT=(EGDT-1./EGDT)/2.
CC    WRITE(6,1345)XA,XB,YA,YB,ZA,ZB
1345  FORMAT(1X,'XYZAB=',6E14.5)
CC    WRITE(6,1346)X1,X2,Y1,Y2,Z1,Z2
1346  FORMAT(1X,'XYZ12=',6E14.5)
      CALL GGS(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM,                 ZAB00270
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,P11,P12,P21,P22)                ZAB00280
      CALL GGS(XA,YA,ZA,XB,YB,ZB,X2,Y2,Z2,X3,Y3,Z3,AM,                 ZAB00290
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,Q11,Q12,Q21,Q22)                ZAB00300
      CALL GGS(XB,YB,ZB,XC,YC,ZC,X1,Y1,Z1,X2,Y2,Z2,AM,                 ZAB00310
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,R11,R12,R21,R22)                ZAB00320
      CALL GGS(XB,YB,ZB,XC,YC,ZC,X2,Y2,Z2,X3,Y3,Z3,AM,                 ZAB00330
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,S11,S12,S21,S22)                ZAB00340
      ZABG=P22+Q21+R12+S11                                             ZAB00350
CC    WRITE(6,7898)ZABG
7898  FORMAT(1X,'EXITING ZABG WITH ZABG=',2E14.5)
      RETURN                                                           ZAB00360
      END                                                              ZAB00370
**********file dzabgnloss.f*******
C****DOUBLE PRECISION VERSION
C***PROGRAM TO CALCULATE MUTUAL IMPEDANCE BETWEEN TWO DIPOLES          DZA00020
C***WITH ARBITRARY LENGTH AND ORIENTATION.  A PIECEWISE-               DZA00030
C***SINUSOIDAL CURRENT DISTRIBUTION IS ASSUMED.                        DZA00040
      SUBROUTINE DSZABG(SX1,SX2,SX3,SY1,SY2,SY3,SZ1,SZ2,SZ3,           DZA00050
     2SXA,SXB,SXC,SYA,SYB,SYC,SZA,SZB,SZC,DZABG)                       DZA00060
      IMPLICIT REAL*8 (A-H,O-Z)                                        DZA00070
      COMPLEX*16 P11,P12,P21,P22,Q11,Q12,Q21,Q22,R11,R12,R21,R22       DZA00080
      COMPLEX*16 S11,S12,S21,S22,JCOM,GAM,CGDS,SGDS,SGDT,ETA           DZA00090
      COMPLEX*16 DZABG,EP3,EGDS,EGDT
      REAL*4 SX1,SX2,SX3,SY1,SY2,SY3,SZ1,SZ2,SZ3                       DZA00110
      REAL*4 SXA,SXB,SXC,SYA,SYB,SYC,SZA,SZB,SZC                       DZA00120
      REAL*4 FHZ,ER3,SIG3,TD3
      COMMON /F/ FHZ,ER3,SIG3,TD3
      JCOM=(0.D0,1.D0)                                                 DZA00140
      PI=3.1415926535898D0
      TPI=2.0D0*PI
```

```
      B=TPI
      E0=8.854D-12
      U0=1.2566D-6
      OMEGA=TPI*FHZ                                                     WIR01670
CC    WRITE(6,2843)OMEGA                                                WIR01680
2843  FORMAT(1X,'OMEGA=',E12.5)                                         WIR01690
      IF(SIG3.LT.0.0D0)EP3=ER3*E0*DCMPLX(1.0D0,-TD3)                    WIR01720
      IF(TD3.LT.0.0D0)EP3=DCMPLX(ER3*E0,-SIG3/OMEGA)                    WIR01730
CC    WRITE(6,7755)ER3,E0,EP3
7755  FORMAT(1X,'ER3,E0,EP3=',4E12.5)
      ETA=CDSQRT(U0/EP3)                                                WIR01740
      GAM=OMEGA*CDSQRT(-U0*EP3)                                         WIR01750
C**COMPUTE GAMMA BY EQUATION IN HAYT PAGE 333
CC    GAM=JCOM*OMEGA*CDSQRT(U0*EP3)*CDSQRT(1.D0-JCOM*SIG3/(OMEGA*EP3))
CC    WRITE(6,8888)GAM
8888  FORMAT(1X,'HYAT GAM=',2E12.5)
      AM=0.0001D0
      INT=0                                                             DZA00180
      X1=SX1                                                            DZA00190
      X2=SX2                                                            DZA00200
      X3=SX3                                                            DZA00210
      Y1=SY1                                                            DZA00220
      Y2=SY2                                                            DZA00230
      Y3=SY3                                                            DZA00240
      Z1=SZ1                                                            DZA00250
      Z2=SZ2                                                            DZA00260
      Z3=SZ3                                                            DZA00270
      XA=SXA                                                            DZA00280
      XB=SXB                                                            DZA00290
      XC=SXC                                                            DZA00300
      YA=SYA                                                            DZA00310
      YB=SYB                                                            DZA00320
      YC=SYC                                                            DZA00330
      ZA=SZA                                                            DZA00340
      ZB=SZB                                                            DZA00350
      ZC=SZC                                                            DZA00360
      XBA=XB-XA                                                         DZA00370
      YBA=YB-YA                                                         DZA00380
      ZBA=ZB-ZA                                                         DZA00390
      X21=X2-X1                                                         DZA00400
      Y21=Y2-Y1                                                         DZA00410
      Z21=Z2-Z1                                                         DZA00420
      DS=DSQRT(XBA*XBA+YBA*YBA+ZBA*ZBA)                                 DZA00430
      DT=DSQRT(X21*X21+Y21*Y21+Z21*Z21)                                 DZA00440
CC    DSK=B*DS                                                          DZA00450
CC    DTK=B*DT                                                          DZA00460
CC    CGDS=DCMPLX(DCOS(DSK),0.0D0)                                      DZA00470
CC    SGDS=DCMPLX(0.0D0,DSIN(DSK))                                      DZA00480
CC    SGDT=DCMPLX(0.0D0,DSIN(DTK))                                      DZA00490
      EGDS=CDEXP(GAM*DS)
      EGDT=CDEXP(GAM*DT)
      CGDS=(EGDS+1.0D0/EGDS)/2.0D0
      SGDS=(EGDS-1.0D0/EGDS)/2.0D0
      SGDT=(EGDT-1.0D0/EGDT)/2.0D0
      CALL DGGS(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AM,                 DZA00500
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,P11,P12,P21,P22)                 DZA00510
      CALL DGGS(XA,YA,ZA,XB,YB,ZB,X2,Y2,Z2,X3,Y3,Z3,AM,                 DZA00520
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,Q11,Q12,Q21,Q22)                 DZA00530
```

```
      CALL DGGS(XB,YB,ZB,XC,YC,ZC,X1,Y1,Z1,X2,Y2,Z2,AM,                 DZA00540
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,R11,R12,R21,R22)                 DZA00550
      CALL DGGS(XB,YB,ZB,XC,YC,ZC,X2,Y2,Z2,X3,Y3,Z3,AM,                 DZA00560
     2DS,CGDS,SGDS,DT,SGDT,INT,ETA,GAM,S11,S12,S21,S22)                 DZA00570
      DZABG=P22+Q21+R12+S11                                             DZA00580
CC    WRITE(6,8899)DZABG
8899  FORMAT(1X,'EXITING DZABG, DZABG=',2E14.5)
      RETURN                                                            DZA00590
      END                                                               DZA00600
```

REFERENCES

1. C.A. Perez and L.W. Brady, *Principles and Practice of Radiation Oncology*, J.B. Lippincott Co., Philadelphia, 1987.

2. A.W. Guy, "History of biological effects and medical applications of microwave energy," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-32, No. 9, pp. 1182–1200, September 1984.

3. J. Overgaard, "The effect of local hyperthermia alone, and in combination with radiation on solid tumors," in *Cancer Therapy by Hyperthermia and Radiation, Proceedings of the 2nd International Symposium*. C. Streffer (Editor), 2–4 June 1977, pp. 49–61, Urban & Schwarzenberg, Inc., Baltimore, 1978.

4. J. Overgaard, "Clinical hyperthermia, an update," *Proceedings of the 8th International Congress of Radiation Research*, Vol. 2, pp. 942–947, July 1987.

5. "Special issue on phased arrays for hyperthermia treatment of cancer," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-34, No. 5, May 1986.

6. "Special issue on hyperthermia and cancer therapy," *IEEE Transactions on Biomedical Engineering*, Vol. BME-31, No. 1, January 1984.

7. V. Sathiaseelan, M.F. Iskander, G.C.W. Howard, and N.M. Bleehen, "Theoretical analysis and clinical demonstration of the effect of power control using the annular phased-array hyperthermia system," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-34, No. 5, pp. 514–519, May 1986.

8. V. Sathiaseelan, "Potential for patient-specific optimization of deep heating patterns through manipulation of amplitude and phase," *Strahlentherapie Onkologie*, Vol. 165, No. 10, pp. 743–745, 1989.

9. D. Sullivan, "Three-dimensional computer simulation in deep regional hyperthermia using the finite-difference time-domain method," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-38, No. 2, pp. 204–211, Feb. 1990.

10. B.S. Trembly, A.H. Wilson, M.J. Sullivan, A.D. Stein, T.Z. Wong, and J.W. Strohbehn, "Control of the SAR pattern within an interstitial microwave array through variation of antenna driving phase," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-34, No. 5, pp. 568–571, May 1986.

11. G. Sato, C. Shibata, S. Sekimukai, H. Wakabayashi, K. Mitsuka, and K. Giga, "Phase-controlled circular array heating equipment for deep-seated tumors: preliminary experiments," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-34, No. 5, pp. 520–525, May 1986.

REFERENCES
(Continued)

12. P.A. Cudd, A.P. Anderson, M.S. Hawley, and J. Conway, "Phased-array design considerations for deep hyperthermia through layered tissue," *IEEE Transactions on Microwave Theory and Techniques,* Vol. MTT-34, No. 5, pp. 526–531, May 1986.

13. N. Morita, T. Hamasaki, and N. Kumagai, "An optimal excitation method in multiapplicator systems for forming a hot zone inside the human body," *IEEE Transactions on Microwave Theory and Techniques,* Vol. MTT-34, No. 5, pp. 532–538, May 1986.

14. C. De Wagter, "Optimization of simulated two-dimensional temperature distributions induced by multiple electromagnetic applicators," *IEEE Transactions on Microwave Theory and Techniques,* Vol. MTT-34, No. 5, pp. 589–596, May 1986.

15. M. Knudsen and U. Hartmann, "Optimal temperature control with phased-array hyperthermia system," *IEEE Transactions on Microwave Theory and Techniques,* Vol. MTT-34, No. 5, pp. 597–603, May 1986.

16. C.F. Babbs, V.A. Vaguine, and J.T. Jones, "A predictive-adaptive, multipoint feedback controller for local heat therapy of solid tumors," *IEEE Transactions on Microwave Theory and Techniques,* Vol. MTT-34, No. 5, pp. 604–611, May 1986.

17. R.B. Roemer, "Physical and engineering aspects of hyperthermia," *Proceedings of the 8th International Congress of Radiation Research,* Vol. 2, pp. 948–953, July 1987.

18. R.B. Roemer, K. Hynynen, C. Johnson, R. Kress, "Feedback control and optimization of hyperthermia heating patterns: present status and future needs," *IEEE Eighth Annual Conference of the Engineering in Medicine and Biology Society,* pp. 1496–1499, November 1986.

19. A. Boag and Y. Leviatan, "Optimal excitation of multiapplicator systems for deep regional hyperthermia," *IEEE Transactions on Biomedical Engineering,* Vol. BME-37, No. 10, pp. 987–995, October 1990.

20. J.T. Loane III and S.W. Lee, "Gain optimization of a near-field focusing array for hyperthermia applications," *IEEE Transactions on Microwave Theory and Techniques,* Vol. 37, No. 10, pp. 1629–1635, October 1989.

21. P.F. Turner, T. Schaefermeyer, and T. Saxton, "Future trends in heating technology of deep-seated tumors," *Recent Results in Cancer Research,* Vol. 107, pp. 249–262, 1988.

22. P.F. Turner, A. Tumeh, T. Schaefermeyer, "BSD-2000 approach for deep local and regional hyperthermia: physics and technology," *Strahlentherapie Onkologie,* Vol. 165, No. 10, pp. 738–741, 1989.

23. R.T. Compton, Jr., *Adaptive Antennas, Concepts and Performance,* Prentice Hall, New Jersey, 1988.

REFERENCES
(Continued)

24. A. J. Fenn, "Theory and analysis of near-field adaptive nulling," *1986 IEEE Antennas and Propagation Society International Symposium Digest,* Vol. 2, IEEE, New York, pp. 579–582, 1986.

25. A. J. Fenn, "Theory and analysis of near-field adaptive nulling," *1986 Asilomar Conf. on Signals, Systems and Computers*, Computer Society Press of the IEEE, Washington, D.C., pp. 105–109, 1986.

26. A. J. Fenn, "Theoretical near-field clutter and interference cancellation for an adaptive phased-array antenna," *1987 IEEE Antennas and Propagation Society International Symposium Digest*, Vol. 1, IEEE, New York, pp. 46–49, 1987.

27. A.J. Fenn, "Moment-method analysis of near-field adaptive nulling," *IEEE Sixth International Conference on Antennas and Propagation, ICAP 89,* 4–7 April 1989, pp. 295–301.

28. A.J. Fenn, "Evaluation of adaptive phased-array antenna far-field nulling performance in the near-field region," *IEEE Transactions on Antennas and Propagation*, Vol. 38, No. 2, pp. 173–185, February 1990.

29. A.J. Fenn, "Near-field testing of adaptive radar systems," *Proc. 12th Annual Antenna Measurement Techniques Association Meeting and Symposium*. 8–11 October 1990, pp. 13 9–13 14.

30. A.J. Fenn, H.M. Aumann, F.G. Willwerth, and J.R. Johnson, "Focused near-field adaptive nulling: experimental investigation," *1990 IEEE Antennas and Propagation Society International Symposium Digest*, Vol. 1, 7–11 May 1990, pp. 186–189.

31. M.I. Skolnik, *Introduction to Radar Systems*, Second Edition, McGraw-Hill Book Company, 1980.

32. J.W. Strohbehn and R.B. Roemer, "A survey of computer simulations of hyperthermia treatments," *IEEE Transactions on Biomedical Engineering*, Vol. BME-31, No. 1, pp. 136–149, January 1984.

33. K.B. Ocheltree and L.A. Frizzell, "Determination of power deposition patterns for localized hyperthermia: a transient analysis," *International Journal of Hyperthermia,* Vol. 4, No. 3, pp. 281–296, 1988.

34. C.K. Chou, G.W. Chen, A.W. Guy, and K.H. Luk, "Formulas for preparing phantom muscle tissue at various radio frequencies," *Bioelectromagnetics*, Vol. 5, No. 4, pp. 435–441, 1984.

35. Y. Zhang, W.T. Joines, J.R. Oleson, "The calculated and measured temperature distribution of a phased interstitial antenna array," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 38, No. 1, pp. 69–77, January 1990.

REFERENCES
(Continued)

36. J.H. Richmond, "Radiation and scattering by thin-wire structures in the complex frequency domain," Ohio State University, ElectroScience Laboratory, Technical Report 2902-10, July 1973.

37. J.H. Richmond, "Computer program for thin-wire structures in a homogeneous conducting medium," Ohio State University, ElectroScience Laboratory, Technical Report 2902-12, August 1973.

38. J.H. Richmond, "Radiation and scattering by thin-wire structures in a homogeneous conducting medium (computer program description)," *IEEE Trans. Antennas Propagat.*, Vol. AP-22, no. 2, p. 365, March 1974.

39. A.J. Fenn, "Focused near-field nulling for adaptive electromagnetic hyperthermia applications," submitted to *1991 Progress in Electromagnetics Research Symposium,* 1–5 July 1991.

40. A.J. Fenn, "Adaptive hyperthermia for improved thermal dose distribution," submitted to *9th International Congress of Radiation Research,* 7–12 July 1991.

41. J.R. Johnson, A.J. Fenn, H.M. Aumann, and F.G. Willwerth, "An experimental adaptive nulling receiver utilizing the sample matrix inversion algorithm with channel equalization," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 39, No. 5, pp. 798–808 May 1991.

42. G. Strang, *Linear Algebra and Its Applications*, New York: Academic, 1976.

43. J.T. Mayhan, "Some techniques for evaluating the bandwidth characteristics of adaptive nulling systems," *IEEE Trans. Antennas Propagat.*, Vol. AP-27, No. 3, pp. 363–373, May, 1979.

44. A.J. Fenn, "Maximizing jammer effectiveness for evaluating the performance of adaptive nulling array antennas," *IEEE Trans. Antennas Propagat.*, Vol. AP-33, No. 10, pp. 1131–1142, Oct. 1985.

45. W.L. Stutzman and G.A. Thiele, *Antenna Theory and Design*, New York: Wiley, 1981.

46. I.J. Gupta and A.A. Ksienski, "Effect of mutual coupling on the performance of adaptive arrays," *IEEE Trans. Antennas Propagat.*, Vol. AP-31, No. 5, pp. 785–791

47. A.J. Fenn, "Moment-method analysis of near-field adaptive nulling," MIT Lincoln Laboratory, Technical Report 842, 7 April 1989.

48. J.D. Kraus, *Antennas,* 2nd Edition, McGraw-Hill Book Company, New York, 1988.

49. D.T. Paris and F.K. Hurd, *Basic Electromagnetic Theory*, Chapter 7, McGraw-Hill Book Company, New York, 1969.

REFERENCES
(Continued)

50. S.A. Schelkunoff and H.T. Friis, *Antennas: Theory and Practice*, Wiley, New York, 1952.

51. "Program manual for the simplified space payload thermal analyzer (Version 3.0/VAX)," Arthur D. Little, Inc., Cambridge, Massachusetts, Prepared for National Aeronautics and Space Administration, Contract NAS5-27606.

52. J.T. Bartoszek, B. Huckins, M. Coyle, "A simplified shuttle payload thermal analyzer (SSPTA) program," *AIAA 14th Thermophysics Conference*, June 4–6, 1979

53. "Thermal Network Modeling Handbook," TRW Systems Group, prepared for National Aeronautics and Space Administration, Contract NAS 9-10435, January 1972.

54. D. Sullivan. "Mathematical methods for treatment planning in deep regional hyperthermia, " *IEEE Transactions on Microwave Theory and Techniques*, Vol. 39, No. 5, pp. 864–872, May 1991.

55. R.C. Johnson and H. Jasik, *Antenna Engineering Handbook, Second Edition*, Chapter 22, McGraw-Hill Book Company, New York, 1984.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE 3 July 1991	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE

Application of Adaptive Nulling to Electromagnetic Hyperthermia for Improved Thermal Dose Distribution in Cancer Therapy

5. FUNDING NUMBERS

C — F19628-90-0002
PE — 63250F
PR — 227

6. AUTHOR(S)

Alan J. Fenn

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lincoln Laboratory, MIT
P.O. Box 73
Lexington, MA 02173-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-917

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Air Force
Space Systems Division
PO Box 92960
Los Angeles, CA 90009-2960

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ESD-TR-91-079

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Adaptive nulling is applied to the problem of generating a therapeutic thermal dose distribution in electromagnetic hyperthermia threatment of cancer. A system design concept for implementing adaptive hyperthermia is introduced. With the proposed design concept, it may be possible to maximize the applied electric field at a tumor position in the target body and simultaneously minimize or reduce the electric field at target positions where undesired high temperature regions (hot spots) occur. In a clinical situation, either a gradient search algorithm or sample matrix inversion algorithm would be used to rapidly form the adaptive null (or nulls) prior to any significant tissue heating.

Analysis of an annular phased-array antenna embedded in an infinite homogeneous medium shows the potential merit of combining adaptive nulling with conventional near-field focusing used in hyperthermia. The analysis is based on a well-known moment-method theory for conducting thin-wire antennas in a homogeneous conducting medium. The theory and software used to compute the moment-method received voltage at a short-dipole probe due to a transmitting dipole array are documented. Computer simulations show that adaptive nulling can prevent undesired high-temperature regions from occurring while simultaneously heating a deep-seated tumor site.

14. SUBJECT TERMS

electromagnetic hyperthermia	phased arrays
cancer therapy	adaptive nulling
adaptive arrays	method-of-moments

15. NUMBER OF PAGES

119

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR