5-31-1990

Cryogenic characterization of potential high temperature superconducter substrates at microwave frequencies

Sukhmani Kaur
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

Title of Thesis: Cryogenic characterization of potential high temperature superconductor substrates at microwave frequencies

Sukhmani Kaur, Master of Science, 1990

Thesis directed by: Dr. Edip Niver, Associate Professor

The lack of characterization data for dielectric substrate materials at microwave frequencies and cryogenic temperatures hinders the development of high T_c superconducting circuits. Low loss ($\tan\delta < 10^{-4}$) substrates with dielectric constants in the range of 5-100 at cryogenic temperatures are needed to realize the advantages of superconductor circuits. Measurement techniques previously reported are either time consuming or inaccurate. We report an adaptation of Courtney's method for accurate, simple evaluation of complex dielectric constants ($10 < \varepsilon_r < 100$) at low temperatures. The fixture was calibrated using known properties of polycrystalline $ZrSnTiO_4$. We measured ε_r and $\tan\delta$ on three potential substrate materials, Al_2O_3, MgO, and $LaAlO_3$ at 5-10 GHz in the temperature range 77-300K. No temperature hysteresis in either ε_r or $\tan\delta$ was observed. The effects of inhomogeneity in the samples were studied, especially effects on the loss tangent. The accuracy of the ε_r measurements are 5%; the $\tan\delta$ accuracy is $\pm 0.5 \times 10^{-5}$. Based on our results at 77K, alumina ($\tan\delta = 0.229\times10^{-4}$, $\varepsilon_r = 9.72$) and MgO ($\tan\delta = 0.414\times10^{-4}$, $\varepsilon_r = 9.613$) are suitable. Results on the loss tangent of $LaAlO_3$ were inconclusive because of inhomogeneities in the crystal.

CRYOGENIC CHARACTERIZATION OF POTENTIAL HIGH TEMPERATURE SUPERCONDUCTOR SUBSTRATES AT MICROWAVE FREQUENCIES

by

Sukhmani Kaur

Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institute of Technology in partial fulfillment of
the requirements for the degree of
Master of Science in Electrical Engineering
1990

APPROVAL SHEET

Title of Thesis: Cryogenic characterization of potential high temperature superconductor substrates at microwave frequencies

Name of Candidate: Sukhmani Kaur
Master of Science in Electrical Engineering, 1990

Thesis and Abstract Approved: _____ May 9, 1990
Dr. E. Niver — Date
Associate Professor
Electrical Engineering

_____ 5.3.90
Dr. A. Fathy — Date
Member, Technical Staff
David Sarnoff Research Center

_____ 5/9/90
Dr. J. Matey — Date
Member, Technical Staff
David Sarnoff Research Center

_____ 5-3-90
Dr. K. Sohn — Date
Professor
Electrical Engineering

VITA

Name: Sukhmani Kaur.

Permanent address:

Degree and date to be conferred: M.Sc., E.E., 1990

Date of birth:

Place of birth:

Secondary education: Guru Harkrishan Public School,
New Delhi, India, May 1983.

Collegiate institutions attended	Dates	Degree	Date of Degree
Essex County College	9/83-8/84		
New Jersey Institute of Technology	9/84-5/88	B.Sc.	May 1988
New Jersey Institute of Technology	9/88-5/90	M.Sc.	May 1990

Major: Electrical Engineering

Minor: Biomedical Engineering

DEDICATION

This thesis is dedicated to my parents who worked hard all their lives so I could be where I am today, and to my brother who has always supported me unfailingly.

ACKNOWLEDGEMENT SHEET

First of all, I would like to thank my advisor, Dr. E. Niver, for all his help. I also thank my supervisors, James Matey, Aly Fathy and Dave Kalokitis, at David Sarnoff Research Center for their enormous help in the past two years. Besides my supervisors I would like to thank some other employees of DSRC for their encouragement and help throughout my work there. They are: Bogdan Brycki, Erwin Belohoubek, Richard Brown, Yehuda Erie, Val Pendrick and many others. I also thank Dr. Kosonocky for helping me obtain the research assistantship at DSRC.

I would also like to thank my friends from whom I have learnt so much and who were always behind me no matter what. In particular, I would like to thank my friend, Norayda Cava, who has been with me through my ups and downs for the past five years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

APPENDICES FIGURES:

LIST OF GRAPHS

CHAPTER I

INTRODUCTION

The discovery of High Temperature Superconductivity by Bednorz and Muller [9] stimulated a global research effort. Superconductors are materials which have zero DC resistance below their transition or critical temperature, T_c. They also exclude magnetic fields below T_c; this exclusion is commonly called the Meissner Effect.

One area of research attempts to bring T_c as close to room temperature as possible. Another area attempts to utilize these materials to improve the performance of microwave (μW) circuits. The improvements would result from reduced conductor losses due to the low surface resistance of the superconductor at microwave frequencies.

To maximize the superconductors performance, substrate losses must be low. The dielectric constant at cryogenic temperatures be accurately known. Superconductors like $YBa_2Cu_3O_{7-x}$ (yittrium barium copper oxide, sometimes referred to as YBCO or 1-2-3) on different substrates such as strontium titanate are currently deposited. But these microwave substrates have not been well characterized down to liquid nitrogen temperature (77 K).

Potential microwave substrates should:

- be homogeneous and isotropic so that the circuit performance is not location or orientation related.
- be non-magnetic so that its magnetic field does not interfere with the fields excited in the superconductor.
- have low dielectric losses as they will have the dominant effect when conductor losses are eliminated using superconductor in microwave circuit applications.
- have good lattice constant match with the superconductor, with no phase transitions in the temperature range of interest.

- have good thermal expansion coefficient match with the superconductor, so that they are compatible over a large temperature range.

- have no chemical reaction with the deposited film to minimize degradation of the superconductor properties.

Materials currently being considered as good microwave substrates are single crystal strontium lanthanum aluminate ($SrLaAlO_4$), lanthanum orthogallate ($LaGaO_3$), lanthanum aluminate ($LaAlO_3$) and magnesium oxide (MgO).

This thesis characterizes selected dielectric materials which are potential candidates for microwave applications. This characterization includes microwave relative dielectric constant and loss tangent measurements from 77 K to room temperature.

1. Measurement Techniques

Many techniques are available to measure relative dielectric constants, ε_r, and loss tangents, tan δ, at microwave frequencies. These methods can be classified into four categories:

(1) Perturbation Techniques: compare the effect on the cavity resonance and the quality factor by the addition of dielectric samples [10], [13], [16], [17].

(2) Transmission Line Techniques: compare the change in the transmission coefficient and the velocity of propagation of a line with and without dielectric materials [15].

(3) Reflection Techniques: measure the reflection at the boundary of materials [18] to extract the parameters of interest.

(4) Resonance Techniques: measure the resonance frequency and quality factor of resonant dielectric waveguides.

2. Measurement Constraints

For our measurements, the experiment had to satisfy certain constraints:

(1) Measurement range: The technique had to be applicable to a large range of relative dielectric constant values (2 to 100) and also to loss tangent values (down to 10^{-5}).

(2) Measurement accuracy: The accuracy requirement for relative dielectric constant was within 5% and for the loss tangent was around 10%.

(3) Material requirement: The method had to measure the complex dielectric constant ($\varepsilon_r = \varepsilon_r' + j\varepsilon_r''$) for non-magnetic, homogeneous and isotropic materials.

(4) Temperature flexibility: The measurement technique had to be applicable to measurements from room temperature (300 K) to liquid nitrogen temperatures (77 K).

(5) Physical Size: Since the measurements were done at low temperatures, the limitations on the maximum size of the fixture were determined by the cryostat size. The available cryostat had an inner diameter of approximately 2-1/2 inches and a length of about 6 inches.

(6) Mechanical Movement: The setup should require no change in the settings or adjustments after the cryostat was evacuated and sitting in the liquid Helium dewar.

(7) Shape of the material: Since many dielectrics are easily available in rod form, it was preferable to find a method which would use rod shaped samples.

Due to the first two constraints, the perturbation, transmission and reflection techniques were eliminated from consideration. The chosen resonance technique, described by Courtney [2] was modified to satisfy the physical, mechanical and measurement range constraints described above.

Courtney's technique is described in Chapter II and the changes implemented in Chapter III.

CHAPTER II

COURTNEY'S MEASUREMENT METHOD

As mentioned in Chapter I, Courtney's method can be used to measure the relative dielectric constant, ε_r, and the loss tangent, $\tan\delta$, of a dielectric material. Figure 3 shows a fixture consisting of a right-cylindrical dielectric sample sandwiched between two conductor plates. In our experiment, the conductor plates are made of gold plated brass. Courtney's technique is suitable for measurements of the complex dielectric constant of non-magnetic dielectric materials with relative dielectric constant variations between 10 and 100, and with $\tan\delta$ between 10^{-2} and 10^{-5} in the microwave frequency range. The measurement accuracy of the dielectric constant is about 2-3% and that of the losses is around 10%. However, the measurement can also be extended to calculate the complex permeability of magnetic materials [2].

In this measurement, the sample acts as a dielectric resonator. Resonant modes are excited by the microwave energy provided by a probe at the end of a coaxial cable. In transmission measurements, a second probe located on the opposite side of the sample couples to the energy transmitted through the sample. Most of the energy provided to the sample is confined within, and only a very small amount is present outside. This is primarily due to the higher dielectric constant of the sample as compared to that of the air surrounding the sample.

We modified this method of measurement to be compatible for cryogenic measurements. Hence a number of changes were implemented in the fixture. These are described in the next chapter.

1. Resonance Modes

Various resonance modes can be excited in the cylindrical sample (see Fig. 1) according to the frequency and energy provided by the coupling loop. These discrete modes have different resonance frequencies. The

frequency of each resonance is dependent on the ε_r, the dimensions of the material and the properties of the mode. The equations to calculate these frequencies are shown in Appendix C.

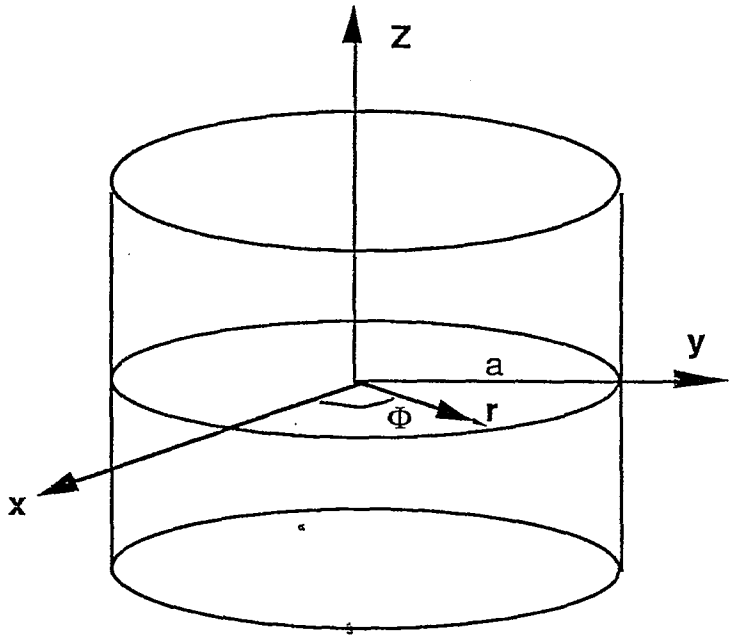


Fig.1 Cylindrical Coordinate System

In general, there are three electric field components and three magnetic field components associated with each mode in any curvilinear coordinate system. The notations TE and TM are acronyms for Transverse Electric and Transverse Magnetic modes. This notation denotes that the longitudinal component of the electric or the magnetic field (i.e. E_z or H_z) is not present in the TE or TM mode, respectively. The HE and EH modes are hybrid modes consisting of a combination of TE and TM modes, i.e. all six field components may be present. For the HE or quasi-TE mode, the TE component carries most of the power, whereas for the EH mode, it is the TM component which carries most of the power.

An example of the notation used to describe a mode is TE_{011} where the m, n and p subscripts for TE_{mnp} represent the configuration of the field lines associated with this particular mode. The first subscript, m, denotes the azimuthal or ϕ variation of the fields for that particular mode in the

cylindrical coordinate system. In the dielectric resonator case this is either $\cos(m\,\phi)$ or $\sin(m\,\phi)$ with m = 0, 1, 2, ... The second subscript n refers to the number of nodes of the field in the radial or r direction, that are present inside the dielectric material rod. Similarly, p, the last subscript, represents the number of nodes of the field in the axial or z direction, inside the resonator. The subscripts n and p take values 1, 2, 3, ... Typical field representation is shown in Fig. 2(a).

For our measurement setup, the modes seen in the frequency spectrum, are in the following ascending order:

TM_{210}, HE_{111}, TM_{020}, TE_{011}, HE_{211}, TM_{011}, ...

The expressions which are used to calculate the ε_r and the tan δ in this experiment belong to the TE_{011} mode. This mode can be easily identified as described in Section 3 of this chapter. Other modes such as TE_{012} can also be used to carry out the same measurement if they can be clearly identified and are highly trapped in the dielectric rod. The field expressions corresponding to each mode are given in Appendix C.

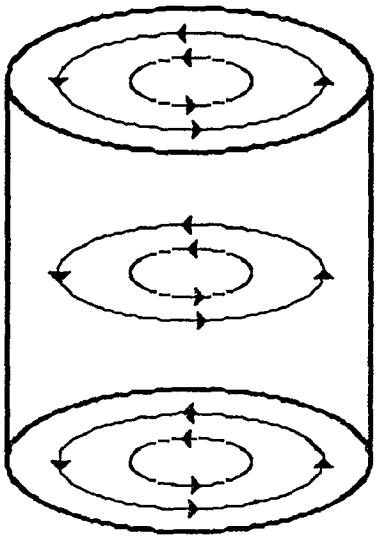


Fig. 2(a) Electric field lines for TE_{01} mode inside a dielectric resonator

2. Coupling to the TE_{011} Mode

In Courtney's article, energy is coupled to the electric field lines associated with the TE_{011} mode. The configuration of the electric field lines for this mode is shown in Fig. 2(a). When the sample is sandwiched between two conducting plates, as shown in Fig. 2(b), the electric field tends to vanish at the conductor plates.

Coupling to these field lines can be done by using two right angle E-field probes (see Fig. 3). This measurement method uses two probes, one probe excites the sample while the other probe senses the transmitted fields. This technique is called the Transmission Measurement technique.

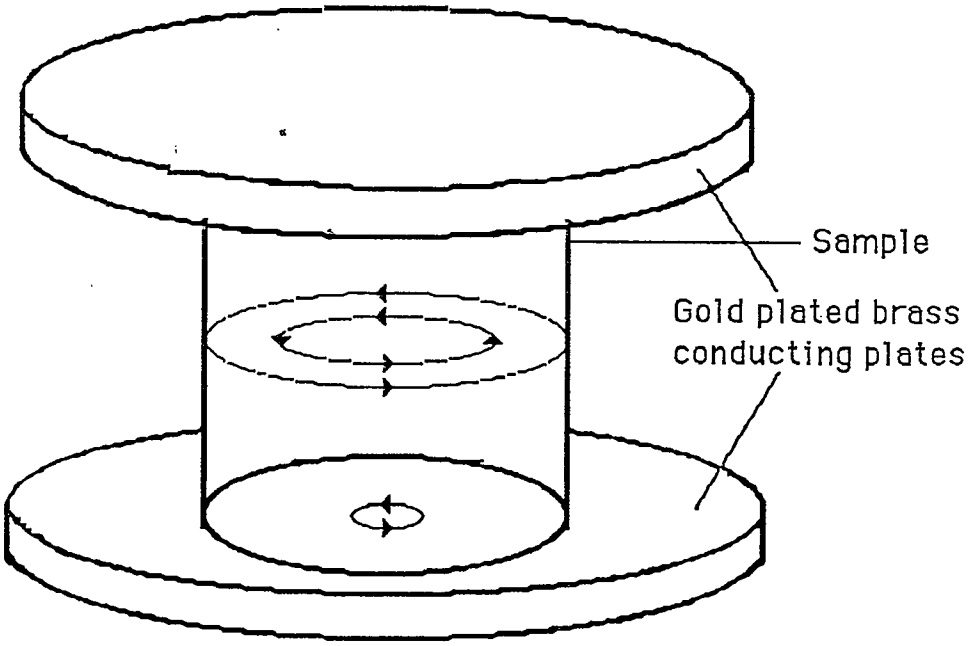


Fig. 2(b) Electric field lines for TE_{011} mode in a dielectric resonator sandwiched between two conducting plates

3. Identification of TE_{011} Mode

There are different techniques for identifying the TE_{011} mode once it has been excited. The most important characteristic is that this mode is the second mode observed on the discrete spectrum. This is because the TM_{mn0} modes generally tend to be very lossy [6] and hence are difficult to observe because of their relatively low Q (due to radiation).

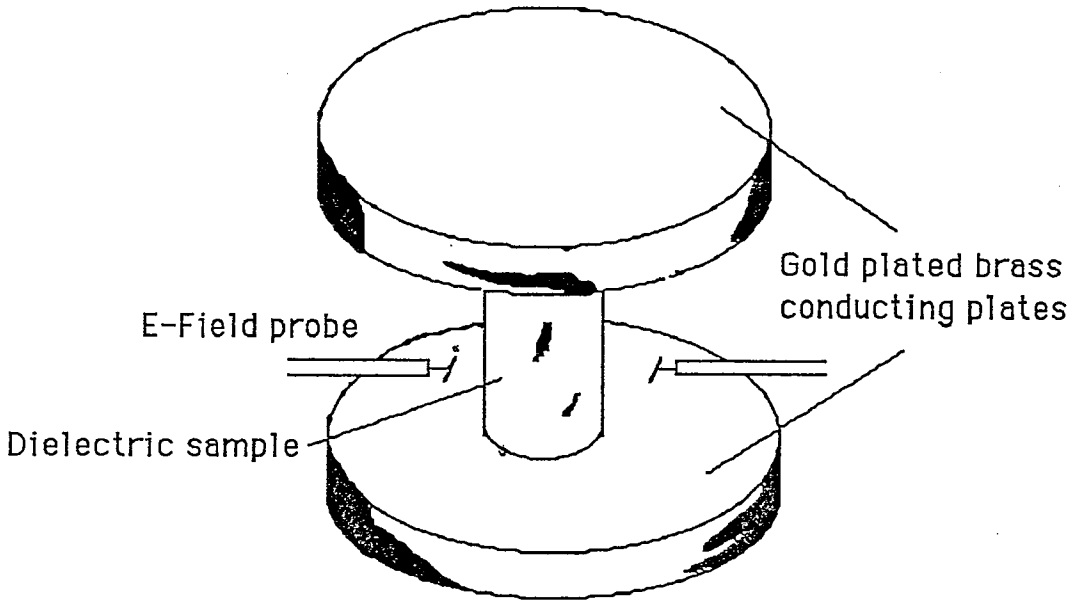


Fig. 3 Coupling to electric lines of force using E-field probes

The interpretation of the mode identification charts [6], i.e. calculating the frequencies of resonance of the various modes for a particular dielectric material of certain dimensions can ensure the previously identified modes. The appropriate equations will be shown in detail later on.

However, if the TM modes are visible on the network analyzer display, then the TE_{011} mode is not the second mode in the frequency spectrum (as noted in technique one for identification) and the TE and TM modes can be distinguished by slightly moving the top plate up and down. The TM modes move up in frequency as the top plate is moved up, while the

TE modes remain stagnant. A good measurement setup has relatively enough separation between these modes.

4. Quality Factor and Loss Tangent

The quality factor, Q is the ratio of the maximum energy stored to the energy dissipated, i.e.

$$Q = 2\pi \frac{\text{maximum energy stored during a cycle}}{\text{average energy dissipated per cycle}}$$

The loss tangent of a dielectric is inversely proportional to the dielectric Q, Q_d of the material. Q_d is related to the unloaded Q (i.e. the total Q due to the losses in the resonator setup) by the equation:

$$\frac{1}{Q_d} = \frac{1}{Q_0} - \frac{1}{Q_c} - \frac{1}{Q_r}$$

where, Q_0 is the unloaded Q

Q_c is the Q due to the conductivity of the plates

Q_r is the Q due to the radiation from the sample

The loss tangent, tanδ, is given by:

$$\tan\delta = \frac{1}{Q_d}$$

5. Formulas for Calculations

The formulas for relative dielectric constant and loss tangent calculations are given by the following two equations:

$$\varepsilon_r = 1 + \left(\frac{c}{\pi D f_0}\right)^2 (\alpha^2 + \beta^2) \qquad (1)$$

$$\tan\delta = \frac{A}{Q_0} - B \qquad (2)$$

where, $$\alpha = \frac{\pi D}{\lambda}\sqrt{\left[\varepsilon_r - \left(\frac{p\lambda}{2L}\right)^2\right]}$$

$$\beta = \frac{\pi D}{\lambda}\sqrt{\left[\left(\frac{p\lambda}{2L}\right)^2 - 1\right]}$$

$$A = 1 + \frac{F(\alpha)G(\beta)}{\varepsilon_r}$$

$$B = \frac{R_s[1 + F(\alpha)G(\beta)]}{2\pi f_0^3 \mu_0^2 L^3 \varepsilon_0 \varepsilon_r}$$

$$F(\alpha) = \frac{J_1^2(\alpha)}{[J_1^2(\alpha) - J_0(\alpha)J_2(\alpha)]}$$

$$G(\beta) = \frac{[K_0(\beta)K_2(\beta) - K_1^2(\beta)]}{K_1^2(\beta)}$$

J_m - Bessel function of the first kind of order m

K_m - Modified Bessel function of the first kind of order m

J_m' - Derivative of J_m with respect to its argument α

K_m' - Derivative of K_m with respect to its argument β

ε_0 - Permittivity of free space

ε_r - Relative dielectric constant of the material

μ_0 - Permeability of free space

D - Diameter of the sample

L - Height or thickness of the sample

f_0 - Frequency of resonance of TE_{011} mode

λ - Wavelength corresponding to f_0

Q_0 - Unloaded Q of the sample

p - Third subscript of a mode (p = 1 for TE_{011} mode)

R_s - Surface resistance of the conducting plates

c - Speed of light

The frequency of resonance (f_0) and the unloaded Q (Q_0) are measured in the experiment. The process of f_0 and Q_0 measurement is described in Appendix B [1] and the program 'Q0.FOR' is given in Appendix C.

6. Potential Measurement Problems

An air gap between the sample and the conductor plates can cause measurement problems if the relative dielectric constant of the sample is high [12] since the electric field lines see the sample as consisting of the dielectric material with air gaps on each end. This can result in underestimating the ε_r. Fortunately, this problem is not too serious in our measurement setup, since the electric field vanishes at the conductor plates (see Fig. 1(b)).

A second problem that can exist is the radiation problem of the sample for the TE_{011} mode. This results in overestimating the measured dielectric losses. This problem can be overcome by using samples with a height less than one-half the wavelength in free space corresponding to the frequency of TE_{011} mode [3].

After the setup was modified, two other issues: finite plate size due to restriction of the cryostat size and unknown surface resistance of gold plated brass, came up. These difficulties and the resolution will be described in the next chapter.

CHAPTER III

MODIFIED MEASUREMENT SETUP

1. Fixture Modifications

The measurement fixture utilized in current measurement setup is basically Courtney's fixture with six changes made to reduce it's complexity and to make it more compatible for low temperature measurements:

(1) Compensation for Dimensional Changes: To utilize the fixture for cryogenic measurements, it was necessary that the fixture be capable of compensating for the dimensional changes in the sample due to the temperature variation. Therefore, the bottom plate of the fixture was spring loaded so no air gaps are produced between the conductor plates and the sample if the sample or conductor dimensions change. The two conducting plates were then screwed into a grooved chassis so that various size samples could be fitted into the fixture by moving the plates closer or further apart [21].

(2) Magnetic Field Coupling: The second change was the use of a horizontal loop (Fig. 4) instead of the right angle E-field probe (Fig. 3) to couple the energy. The loop couples to the magnetic field lines set up in the fixture instead of the electric field lines. The magnetic field was found to be easier to couple to and was therefore used.

(3) Top Coupling: The formulas for ε_r and $\tan\delta$ have been derived with the assumption that the plates are infinite. But due to the practical limitations of infinite plates, a ratio of three or greater for the diameter of the conductor plates to that of the sample is considered satisfactory since the fields attenuate very rapidly as you move radially away from the sample. The maximum size of the plates is determined by the inner diameter of the cryostat can (approximately two and a half inches) since the fixture is to be enclosed inside it. Bringing the coaxial cable in between the conductor plate and the inner cryostat wall would further reduce the diameter of the plate (Fig. 4(a)). By

bringing the coaxial cable through the plate, the full inner diameter of the cryostat can be used. A right angle loop was soldered to the end of

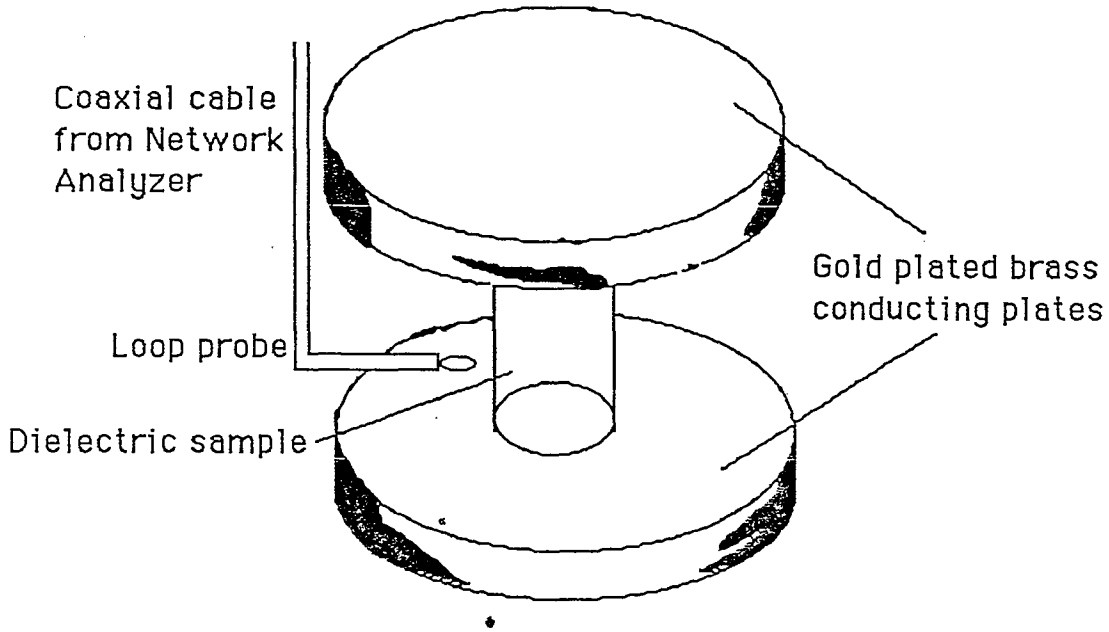


Fig. 4(a) Coupling to magnetic lines of force using side coupling

the coaxial cable. The two methods of measurement, i.e. the side and the top coupling, were used to measure the sensitivity of the measurement due to the hole in the top plate (data to prove this is provided in Appendix F). The results obtained showed almost no change in the data acquired. Accordingly, the second approach of top coupling was utilized, in place of side coupling that was used by Courtney.

(4) Vertical Movement: The next change was brought about due to the requirement of vertical movement capability of the loops, so that samples with varying thicknesses could be measured. The loop should be present at nearly one-half the height of the sample so that the coupling to the magnetic field lines is easy. Hence the coaxial cable, after passing out of the cryostat, was made to pass through a hollow support tube, on top of which a vacuum feed through was placed, so that the cable could be moved up or down relative to the fixture [22].

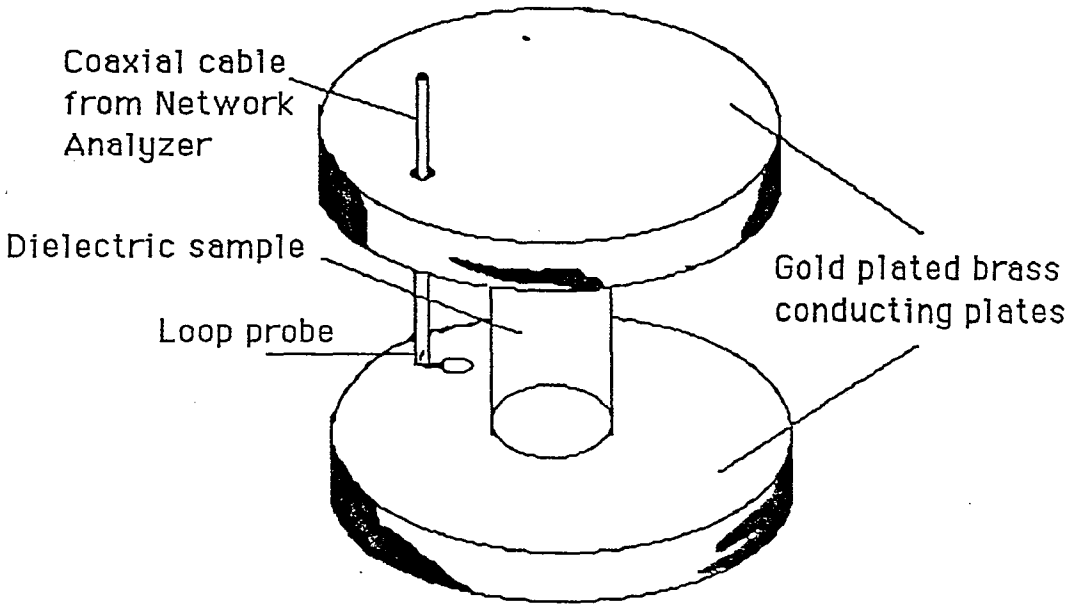


Fig. 4(b) Coupling to magnetic lines of force using top coupling

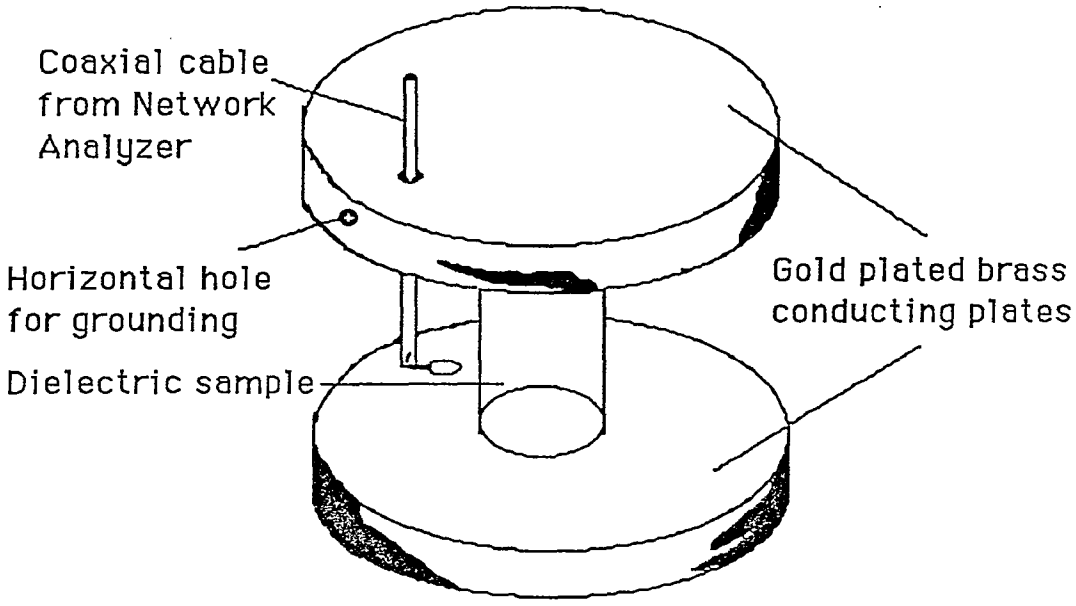


Fig. 4(c) Grounding of top plate using a screw

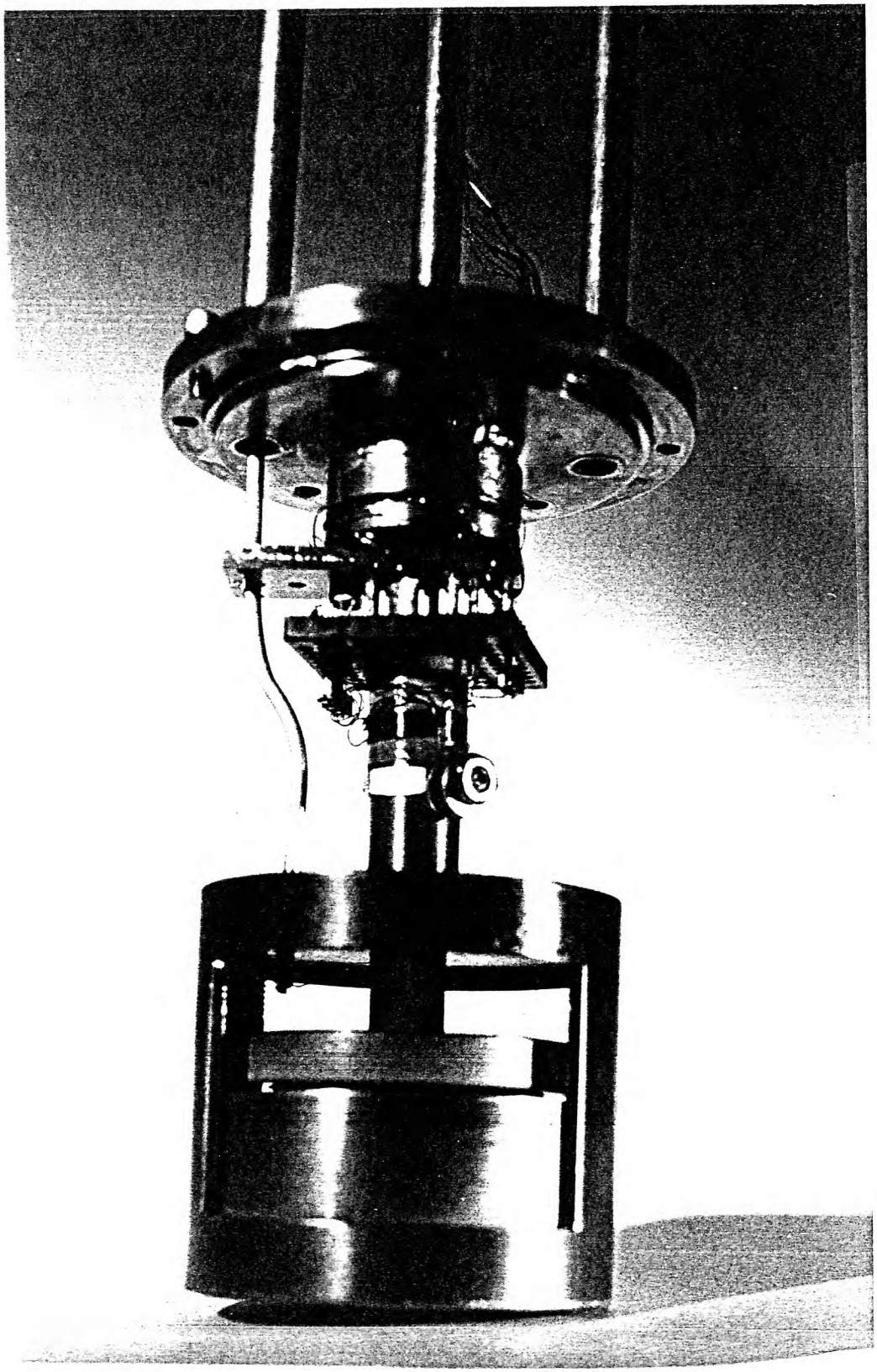


Fig.4(d) Photograph of experimental fixture

(5) Reflection Measurements: Reflection measurements were carried out in place of Transmission measurements. A single loop (Reflection Measurements) was used to reduce the complexity of the fixture in place of two loops for the transmission measurements, as used by Courtney.

(6) Grounding: The last change made in the fixture was due to the grounding requirement of the conducting plates. Since the outer conductor of the coaxial cable is at ground potential, the top plate was brought in contact with this conductor using a screw that passes through a horizontal hole in this plate (Fig. 4(c)). The screw presses the coax against the top plate, grounding it. The bottom plate, in contact with the top plate through the chassis that holds the two plates, is thus also grounded.

Figure 4(d) shows a photograph of the fixture used for the experiment.

2. Sample Requirements for Measurement

In summary, the samples that can be measured in the fixture have certain limitations on their dimensions and properties, as listed below. They should

(1) be right-cylindrical in shape.

(2) be non-magnetic ($\mu = \mu_0$), isotropic (ε and μ same in every direction), and homogeneous (ε and μ independent of position inside the dielectric).

(3) sample thickness should be less than one-half the wavelength of TE_{011} resonance of the dielectric sample being measured. This condition is required to eliminate the radiation losses of the sample.

(4) diameter to thickness ratio of the sample should be as large as possible for good mode separation. This results in non-degenerate modes, and

therefore there is less difficulty in identifying them. This also results in the TM_{nm0} modes (TM modes with subscript p = 0, i.e. no axial variation of the fields) being as far as possible from all other modes. This is essential since a TM_{nm0} mode overlapping with another mode would affect it detrimentally, since the TM_{nm0} modes are very lossy.

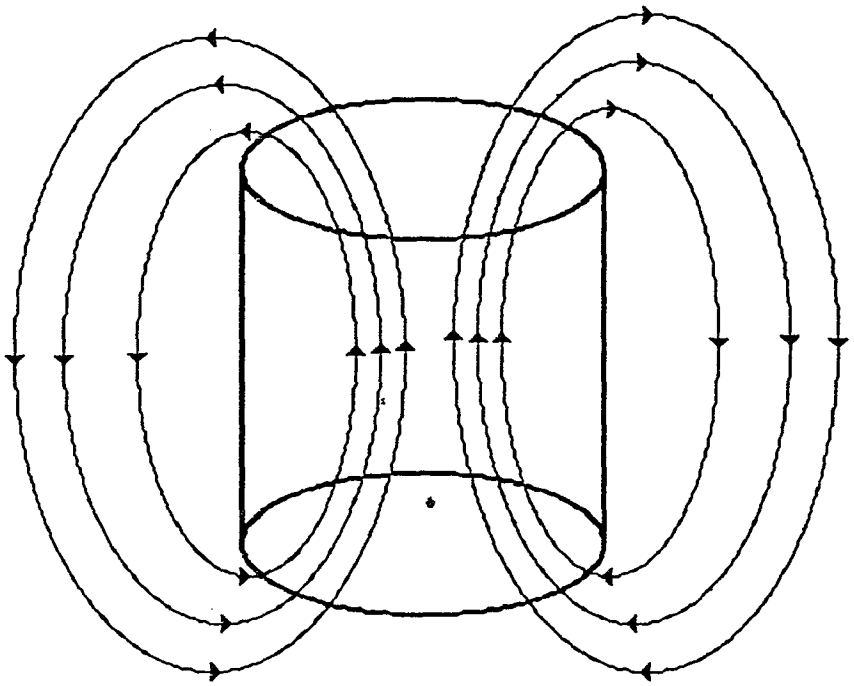


Fig. 5(a) Magnetic field lines for TE_{011} mode for a dielectric resonator

(5) The conducting plate diameter should be at least three times the diameter of the sample. This is necessary to satisfy the condition of infinite conducting plates, as mentioned in Section 1.

3. Coupling to the TE_{011} Mode

Chapter II describes the method of coupling to the electric field lines associated with the TE_{011} mode. The modified fixture utilized the magnetic field lines coupling for the measurement.

The magnetic field distributions of TE_{011} mode are shown in Fig. 5(a). Figure 5(b) shows the magnetic field configurations when the sample is sandwiched between two conducting plates. To achieve the magnetic field coupling to this mode, horizontal wire loops shown in Fig. 4(b) can be utilized [4].

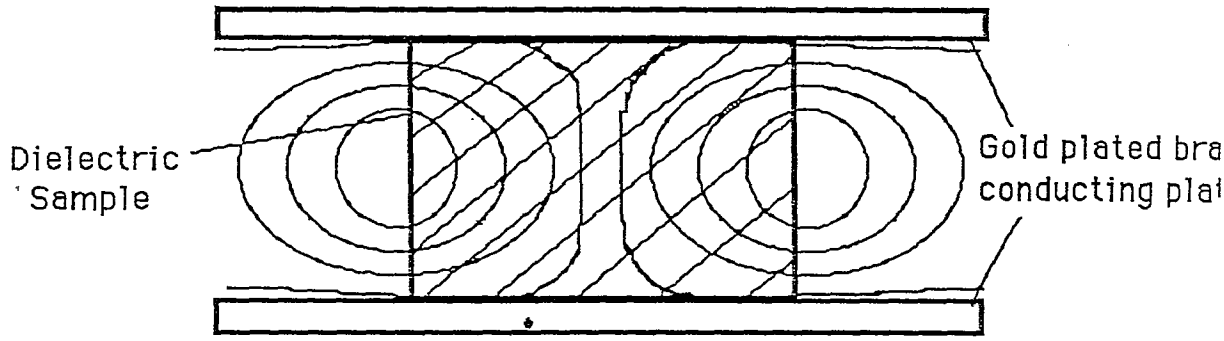


Fig. 5(b) Magnetic field lines for TE011 mode in a dielectric resonator sandwiched between two conducting plates

4. Measuring Equipment Setup

The coaxial cable from the experimental fixture is connected to the Hewlett Packard 8746B S-parameter Test Set (see Fig. 6), which is set to S_{11} mode for Reflection Measurements (see the following section). The resonance can be seen on the Network Analyzer's Polar Display (HP8410A Network Analyzer) and Magnitude Display (HP8418A Auxiliary Display Holder). The frequencies of interest for unloaded Q measurement can be measured with the EIP Model 578 Source Locking Microwave Frequency Counter.

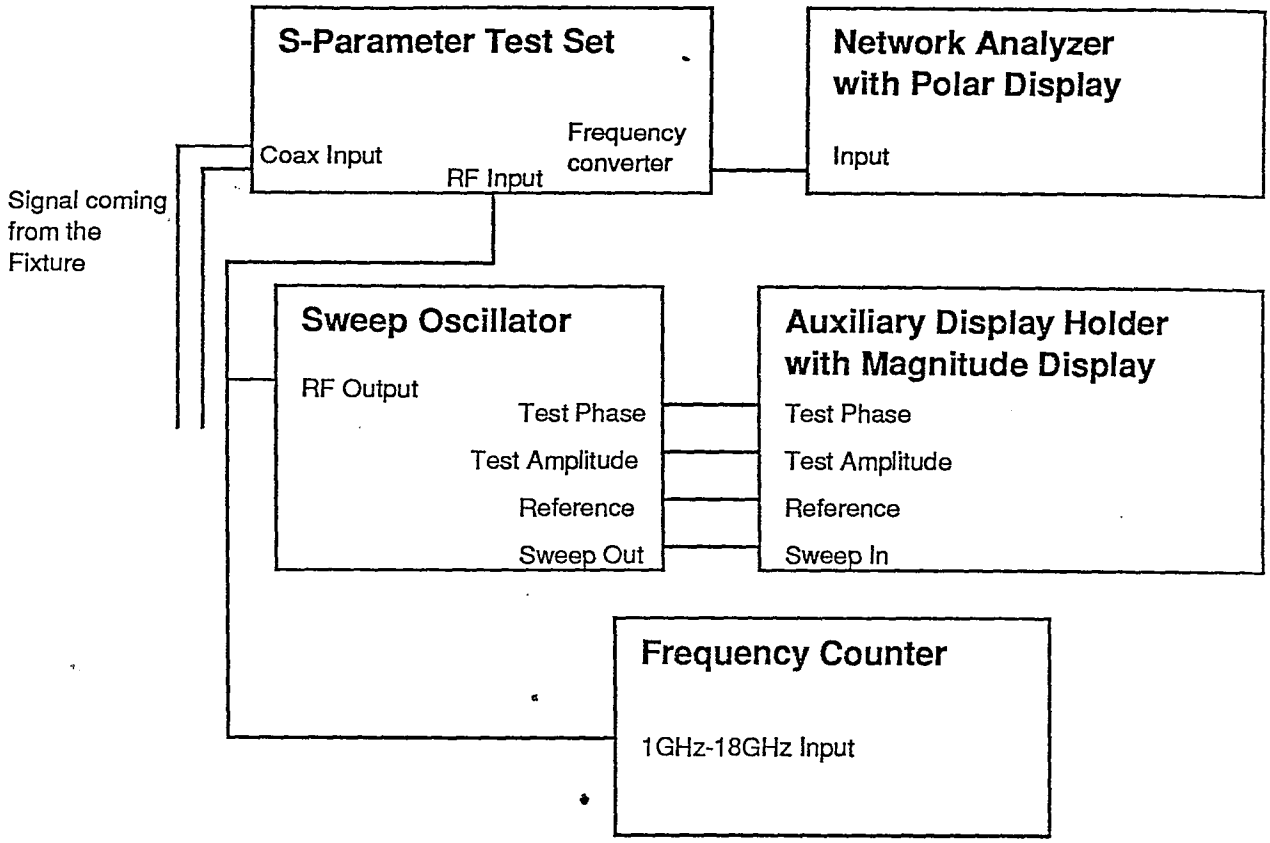


Fig. 6 Equipment setup for measurements

5. Measurement Techniques

There are essentially two ways of measuring the properties of the sample. These are:

(1) Transmission Measurements

(2) Reflection Measurements

The difference between the two techniques is the use of two loops (Transmission Measurements) versus a single loop (Reflection Measurements). As mentioned in Section 1, the latter method was used for the modified fixture.

6. Dielectric Constant and Loss Tangent Formulas

The formulas for the calculation of the relative dielectric constant and the loss tangent are the same as listed in section 4 of Chapter II for Courtney's fixture.

7. Vacuum Pumping System

The cryostat can is initially at atmospheric pressure. It is evacuated with a Veeco VR300 Mobile Model Pumping Station, consisting of two pumps, a Mechanical or Roughing Pump and an Oil-Diffusion or High Vacuum Pump (see Fig. 7). There are three pressure detecting thermocouple gauges that are attached to the system. The first gauge is next to the foreline valve. The second one is close to the roughing valve, and the last one is connected to the line going to the cryostat can.

First, with the diffusion pump isolated, the mechanical pump creates a vacuum of 100 milliTorr (100 mTorr). Then the diffusion pump, with the High-Vacuum valve, is used to further decrease the pressure. For the setup at David Sarnoff Research Center, the lowest vacuum that can be achieved inside the cryostat can (measured at gauge 3) is approximately 10 mTorr. This ultimate pressure is limited by the length and the conductance of the connecting tubing, the pumping speed and the outgassing characteristics of the system.

8. Cryogenic Measurement Procedure

A block diagram of the entire setup is shown in Fig. 8(a) and a photograph of the same in Fig. 8(b). First the sample measurements are performed at room temperature (Appendix B gives the details on how to measure the frequency of resonance and unloaded Q for relative dielectric constant and loss tangent calculations). Next, the thermometer and the heater connections are made to the temperature controller (the heater and thermometer details are given in Appendix D). Then the fixture is enclosed in the cryostat can and the can is screwed close. The pumping system evacuates the cryostat can by following the procedure outlined in Section 7. The measurement of relative dielectric constant and the unloaded Q is once again repeated at room temperature to ensure that no change has occured.

Next the cryostat can is dunked in a dewar containing liquid nitrogen and the temperature of the sample is allowed to decrease. Once the sample reaches a temperature of interest, the temperature is maintained with the controller. The sample is allowed another fifteen minutes to half an hour to stabilize and then measurements are carried out at this new temperature. This process is repeated for all the temperatures of interest. The important thing to note is that the heater is capable of providing only a certain amount of heat. This limits the temperature range in which it can be used. The temperature range for the tested fixture was up to 50K above the liquid nitrogen temperature of 77K. For this reason, the data acquired is limited to measurements between 77K and 127K besides measurements at room temperature.

To speed up the cooling down process, a second procedure can be followed. The process is the same up to the point of evacuating the cryostat can at room temperature. Then the can is cut off from the pumping station and backfilled with gaseous nitrogen. It is then dunked in liquid nitrogen. The presence of a heat transferring gas inside the can, reduces the temperature more quickly. The sample is allowed to reach 77K and measurements at this temperature are carried out. The pumping station is then used to reevacuate the can. After evacuation, the can is removed from the dewar and allowed to warm up to the next temperature of interest. Once it approaches this temperature, it is again dunked in liquid nitrogen and the temperature controller is used to hold the temperature so that the measurements can be repeated at the new temperature. This process is repeated for all the other temperatures of interest.

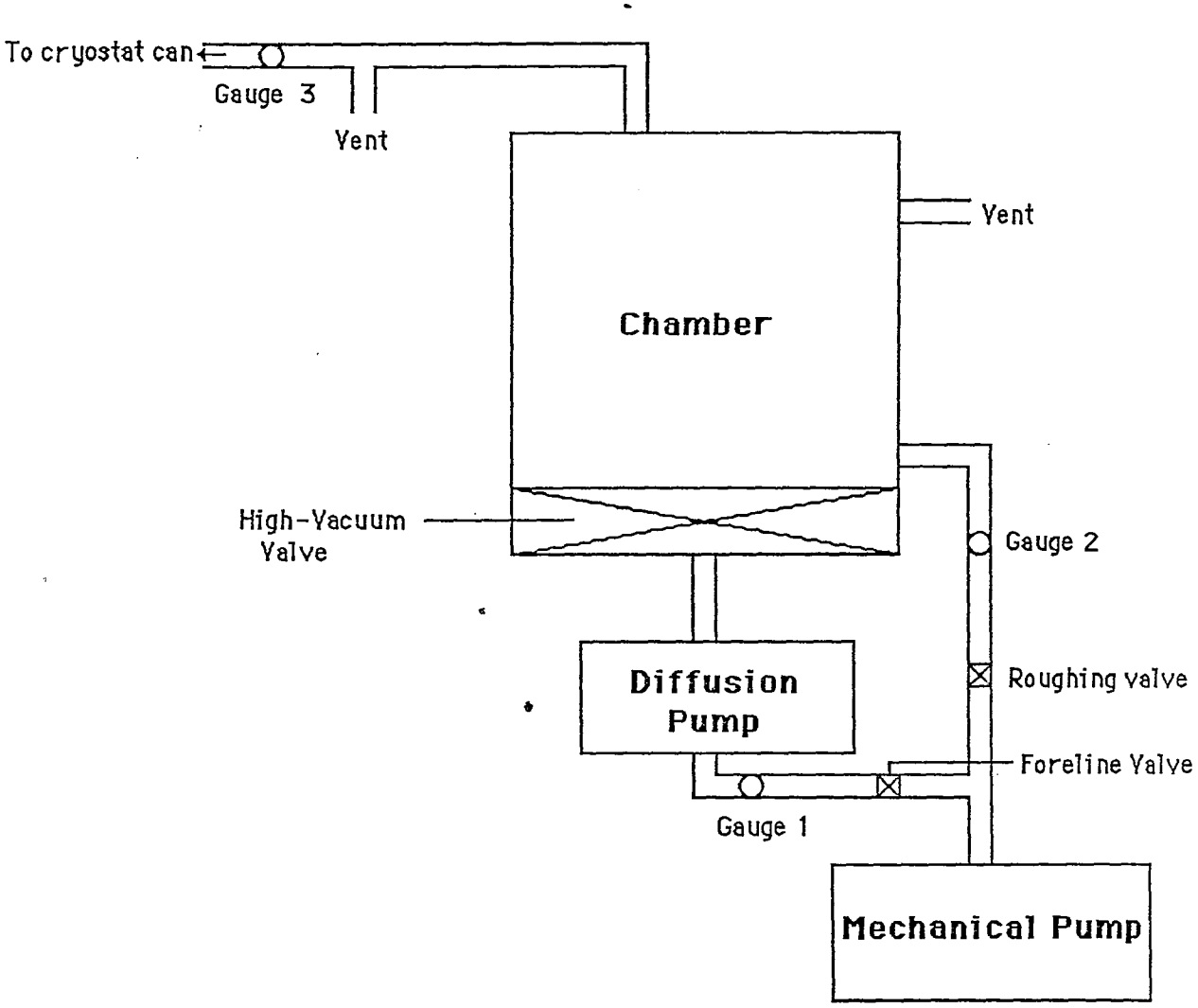


Fig. 7 Vacuum pumping system

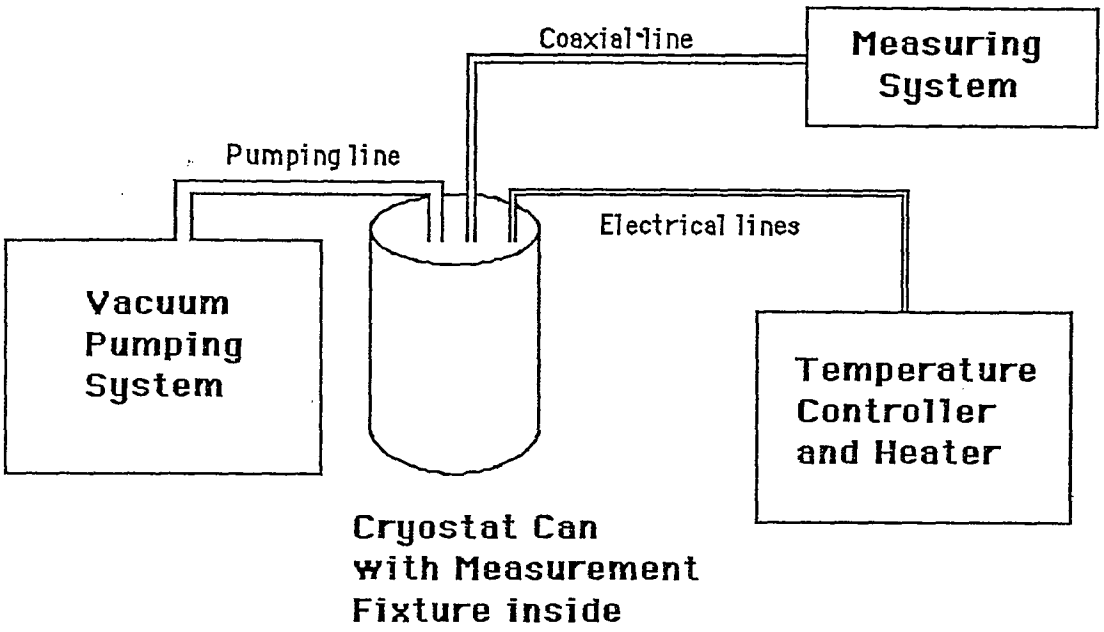


Fig. 8(a) Experimental apparatus

9. Measurement Limitations

As mentioned in Chapter I, the fixture is capable of measuring only a limited range of dielectric properties. These limits are set either by the size of the fixture or by the limit of the measuring technique.

The limit on the relative dielectric constant is brought about by the maximum and the minimum size of the samples that can be measured in the fixture. The maximum size of the sample is limited by the size of the plates since the ratio of the diameter of the plates to that of the sample is required to be at least three times (see section 1). Hence the sample size is limited to 0.65 inches. The smallest sample which can be measured is limited by the coaxial cable diameter. Therefore samples with diameters less than approximately 0.2 inches are not recommended for this fixture.

The limitations on the lower limit of tan δ (i.e. 10^{-5}) is set by the fact that if the substrate losses are below 10^{-5}, the plate conductor losses become dominant and therefore the dielectric losses are difficult to measure. The

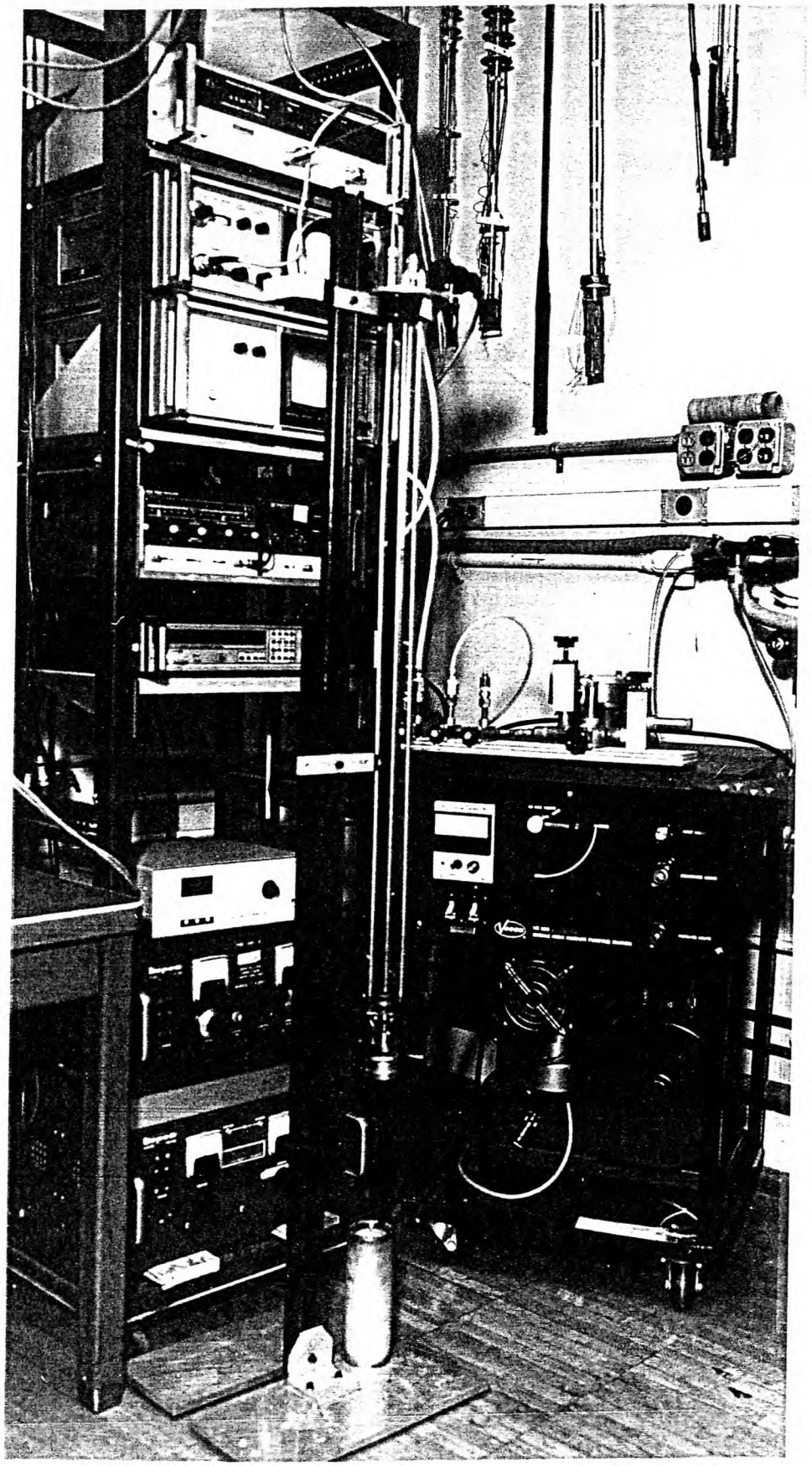


Fig. 8(b) Photograph of experimental setup

upper limit of 10^{-2} is set by the practicality of the measurement method. In any case, we are not interested in materials with high losses.

10. Extension of Technique to Measure the Conductor Losses

The conducting plates are made out of brass and then electroplated with gold. The conductivity of these plates is in between that of plated gold and brass. In general, electroplated gold has lower conductivity than bulk gold. It is important to know the conductivity so that the conductor losses can be accurately calculated. The Courtney Technique can be modified to measure the Conductor losses of the plates so that the tan δ can be accurately measured. To do this, the same experiment has to be repeated

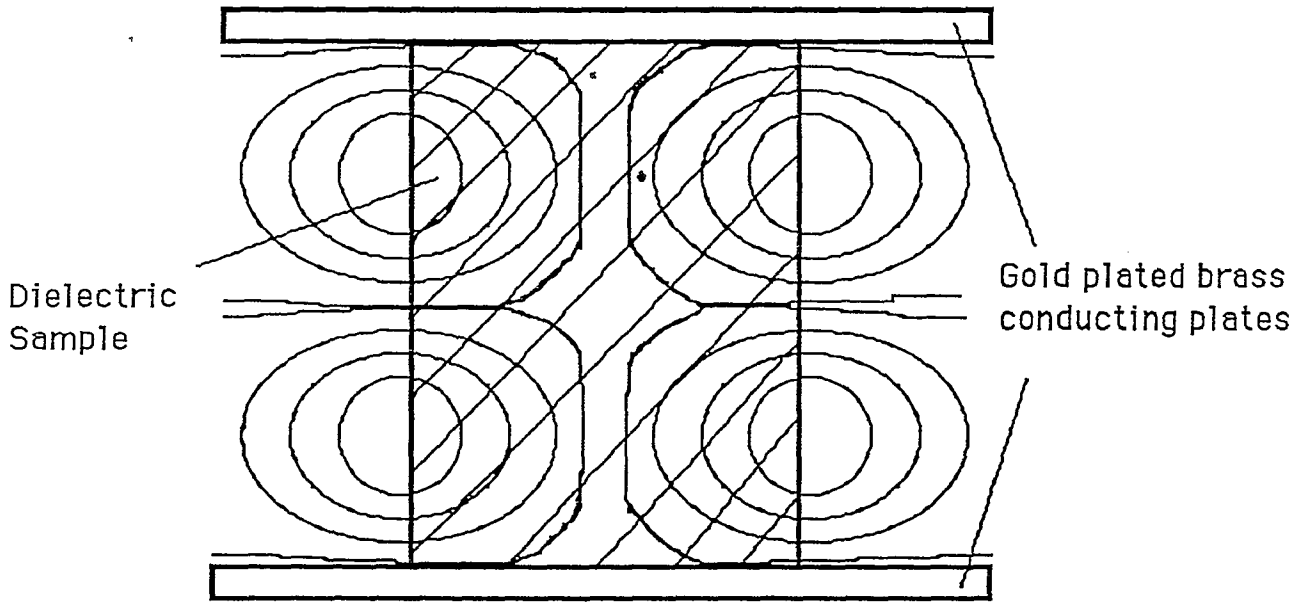


Fig. 9 Magnetic field lines for TE_{012} mode in a twice the thickness resonator

twice, once with a sample designed for TE_{011} resonance measurements, and then with a sample with the same diameter but twice the height (see Fig. 9 for the magnetic field lines configuration). The use of twice the height sample results in the TE_{012} mode to resonate at the frequency of the TE_{011} mode for the original sample.

By using the results of the above listed experiments the conductor losses can be calculated using the formulas derived in the book 'Dielectric Resonators' [5]. The equations used for the calculation of the conductivity of the gold plated brass plates are:

$$\tan\delta = \frac{1}{p-1}\left[1 + \frac{F(\alpha)G(\beta)}{\varepsilon_r}\right]\left[\frac{p}{Q_{0p}} - \frac{1}{Q_{01}}\right] \qquad (3)$$

where, $p = 2$ for TE_{012} mode

Q_{0p} is the unloaded Q measured for the TE_{01p} mode

Q_{01} is the unloaded Q for the TE_{011} mode

where, $$\alpha = \frac{\pi D}{\lambda}\sqrt{\left[\varepsilon_r - \left(\frac{p\lambda}{2L}\right)^2\right]}$$

$$\beta = \frac{\pi D}{\lambda}\sqrt{\left[\left(\frac{p\lambda}{2L}\right)^2 - 1\right]}$$

$$F(\alpha) = \frac{J_1^2(\alpha)}{[J_1^2(\alpha) - J_0(\alpha)J_2(\alpha)]}$$

$$G(\beta) = \frac{[K_0(\beta)K_2(\beta) - K_1^2(\beta)]}{K_1^2(\beta)}$$

Then the surface resistance can be calculated by using the formula:

$$R_s = \frac{B\,[2\pi f_0^3 \mu_0^2 L^3 \varepsilon_0 \varepsilon_r]}{1 + F(\alpha_1)G(\beta_1)} \qquad (4)$$

where $$B = \frac{A}{Q_{01}} - \tan\delta \qquad (\tan\delta \text{ calculated from equation (3)})$$

$$A = 1 + \frac{F(\alpha)G(\beta)}{\varepsilon_r}$$

J_m - Bessel function of the first kind of order m

K_m - Modified Bessel function of the first kind of

order m

J_m' - Derivative of J_m with respect to its argument α

K_m' - Derivative of K_m with respect to its argument β

ε_0 - Dielectric constant of free space

ε_r - Relative dielectric constant of the material

μ_0 - Permeability of free space

D - Diameter of the sample

L - Height or thickness of the sample

f_0 - Frequency of resonance of TE_{011} mode

λ - Wavelength corresponding to f_0

Q_0 - Unloaded Q of the sample

This process of calculation of the conductivity of the plates was tried using two samples of the same dimensions, stacked one on top of the other. But the results obtained were not satisfactory. The reason for this could be the small air gap between the two samples.

Hence, the conductivity was calculated using the published data for a $ZrSnTiO_4$ sample. This procedure is explained in the next chapter.

CHAPTER IV

MEASUREMENTS AND RESULTS

This chapter discusses the sample dimensions calculation process and later discusses the calibration process of the fixture using a polycrystalline $ZrSnTiO_4$ sample. Lastly, it provides and discusses the results obtained for 99.5% Al_2O_3, single crystal MgO and single crystal $LaAlO_3$.

1. Step by Step Sample Dimensions Calculation

In order to design a test sample, the sample dimensions, an approximate value of its relative dielectric constant and the frequency of measurement are required. This frequency corresponds to the frequency of resonance of the TE_{011} mode.

The first step is to pick various acceptable diameters for the sample which are compatible with the fixture. The diameter should be less than one-third the diameter of the conducting plates, i.e. less than 0.65 inches, and larger than 0.2 inches. The upper limit on the diameter is set because of the assumption in the calculations that the conducting plates are infinite, but due to practical limitations, a 1:3 ratio for the diameter of the sample to that of the plates is considered satisfactory. The lower limit on the diameter (of 0.2 inches) is set by the coaxial cable diameter. If the sample is too small, the coax would be unable to couple tightly to the field lines. In general, diameters of 350, 400, 450, 500 and 550 mils (1000 mils = 1 inch) are picked. Then the relative dielectric constant value, the frequency of resonance value, and one of the diameter dimension value, say 350 mils, is input to the Fortran program 'DESIGN.FOR'. The output of this program is the corresponding height of the sample required to have a TE_{011} resonance at the desired frequency. For more details on this program see Appendix C. This process of height calculation is repeated for the other diameters. All these samples are candidates for the sample dimensions to be picked.

After obtaining the above data, the value of one-half the wavelength corresponding to the TE_{011} frequency is calculated. If any of the candidate samples has a height close to or greater than the value of one-half the wavelength, it is eliminated from consideration for the sample. This condition on the height is set to eliminate radiation losses (Chapter II, Section 6). For the remaining samples, the frequency of resonance of modes close to the TE_{011} mode is calculated from the graph in an article by Kobayashi and Tanaka [6] (shown in Fig. 10). The frequencies of HE_{1np}, TE_{0np} and TM_{0np} modes can be verified with programs given in Appendix C. After this process is repeated for all the samples, the frequencies of the various modes is marked on a frequency axis graph. From this graph, the best sample with the TM modes as far away as possible from any main modes, especially the TE_{011} mode, is picked. If none of the graphs are satisfactory, then other diameter values are picked and the process is repeated.

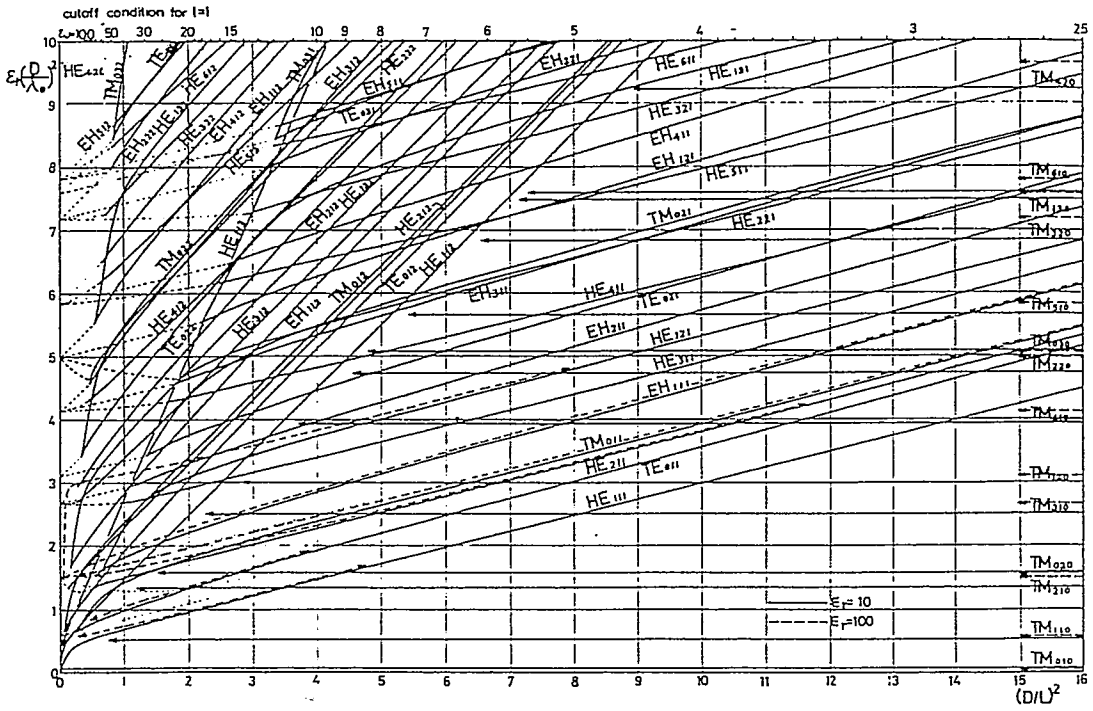


Fig. 10 Mode chart for sample from Kobayashi and Tanaka article [6]

2. Numerical Example

The above discussed calculation process is demonstrated with an example. Let the material have an approximate relative dielectric constant of 26 and let the frequency at which the measurement is to be carried out be 6 GHz (or $6x10^9$ Hz). Picking diameter values of 350, 375, 400, 425, 450, 475, 500 and 525 mils, the corresponding heights for the diameters are calculated using the program 'DESIGN.FOR'. The values for the heights shown in the table are rounded off to the nearest integer divisible by five since usually when cutting the sample, the dimensions are within +/- 5 mils. The data obtained from the program is shown in Table I. Note that 1000 mils = 1 inch = 2.54 cm.

TABLE I

Dimensions Needed to Obtain $f(TE_{011}) = 6GHz$ For Material With $\varepsilon_r = 26$

$\varepsilon_r = 26$

$f(TE_{011}) = 6GHz$

Diameter (D) (mils)	Height (T) (mils)	$(D/T)^2$
350	510	0.471
375	420	0.797
400	370	1.169
425	340	1.563
450	320	1.977
475	300	2.507
500	280	3.189
525	270	3.781

The value of one-half the wavelength corresponding to the frequency of 6 GHz is 984.25 mils. Since the heights of the samples shown in the above table are less than the value of one-half the wavelength, all of the above sample dimensions are good candidates for the measurement.

The next step in choosing the right sample dimensions is the calculation of the frequencies of resonance of various modes for the above samples. This is done by using the graph shown in Fig. 10. The first sample with D = 350 mils and T = 510 mils, has a $(D/T)^2$ ratio of 0.471. This value is noted on the x-axis of the graph and the corresponding values of $\varepsilon_r(D/\lambda)^2$ for the various modes is read off from the y-axis. Since the value of ε_r and D is known, the value of the frequency of resonance can be calculated. Table II show the results of this process. This calculation is repeated for all the other samples in Table I and the results are shown in Tables III through IX.

TABLE II

Frequencies of Resonances of Various Modes for Sample I of Table I

$\varepsilon_r = 26$

D = 350 mils

T = 510 mils

$(D/T)^2 = 0.471$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{011}	0.55	4.908
HE_{111}	0.60	5.126
TE_{011}	0.85	6.102
HE_{112}	1.03	6.717
TM_{011}	1.30	7.546
TE_{012}	1.40	7.831
TM_{210}	1.45	7.969
HE_{211}	1.60	8.371
TM_{020}	1.65	8.501

TABLE III

Frequencies of Resonances of Various Modes for Sample II of Table I

$\varepsilon_r = 26$

D = 375 mils

T = 420 mils

$(D/T)^2 = 0.797$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{110}	0.55	4.581
HE_{111}	0.70	5.168
TE_{011}	0.95	6.020
HE_{112}	1.40	7.308
TM_{210}	1.45	7.438
TM_{011}	1.50	7.565
HE_{211}	1.55	7.690
TM_{020}	1.65	7.934

TABLE IV

Frequencies of Resonances of Various Modes for Sample III of Table I

$\varepsilon_r = 26$

D = 400 mils

T = 370 mils

$(D/T)^2 = 1.169$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{110}	0.55	4.295
HE_{111}	0.70	4.845
TE_{011}	1.15	6.210
TM_{210}	1.35	6.728
TM_{011}	1.45	6.973
HE_{211}	1.50	7.092
TM_{020}	1.65	7.438

TABLE V

Frequencies of Resonances of Various Modes for Sample IV of Table I

$\varepsilon_r = 26$

D = 425 mils

T = 340 mils

$(D/T)^2 = 1.563$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{110}	0.55	4.042
HE_{111}	0.80	4.875
TE_{011}	1.22	6.020
TM_{210}	1.35	6.332
TM_{020}	1.65	7.001
HE_{211}	1.70	7.106
TM_{011}	1.73	7.169
HE_{112}	2.03	7.765
EH_{111}	2.07	7.841

TABLE VI

Frequencies of Resonances of Various Modes for Sample V of Table I

$\varepsilon_r = 26$

D = 450 mils

T = 320 mils

$(D/T)^2 = 1.977$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{110}	0.55	3.817
HE_{111}	0.95	5.017
TE_{01}	1.35	5.981
TM_{210}	1.38	6.047
TM_{020}	1.65	6.612

TABLE VII

Frequencies of Resonances of Various Modes for Sample VI of Table I

$\varepsilon_r = 26$

D = 475 mils

T = 300 mils

$(D/T)^2 = 2.507$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{110}	0.55	3.616
HE_{111}	1.15	5.229
TM_{210}	1.38	5.728
TE_{011}	1.58	6.130
TM_{020}	1.65	6.264
HE_{211}	1.90	6.722
TM_{011}	2.03	6.948

TABLE VIII

Frequencies of Resonances of Various Modes for Sample VII of Table I

$\varepsilon_r = 26$

D = 500 mils

T = 280 mils

$(D/T)^2 = 3.189$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
HE_{111}	1.30	5.282
TM_{210}	1.38	5.442
TM_{020}	1.65	5.951
TE_{011}	1.72	6.076
HE_{211}	2.08	6.681
TM_{011}	2.20	6.871

TABLE IX

Frequencies of Resonances of Various Modes for Sample VIII of Table I

$\varepsilon_r = 26$

D = 525 mils

T = 270 mils

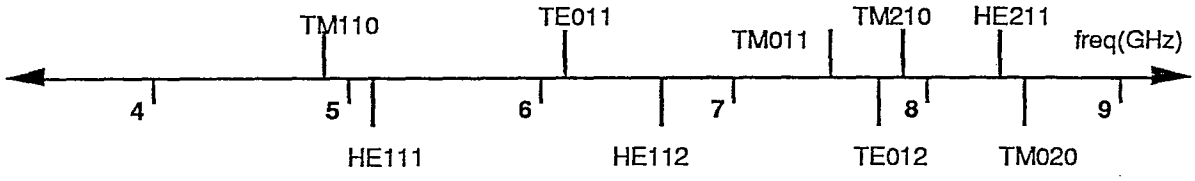
$(D/T)^2 = 3.781$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{210}	1.38	5.183
HE_{111}	1.42	5.257
TM_{020}	1.65	5.667
TE_{011}	1.85	6.001
HE_{211}	2.25	6.618
TM_{011}	2.35	6.763

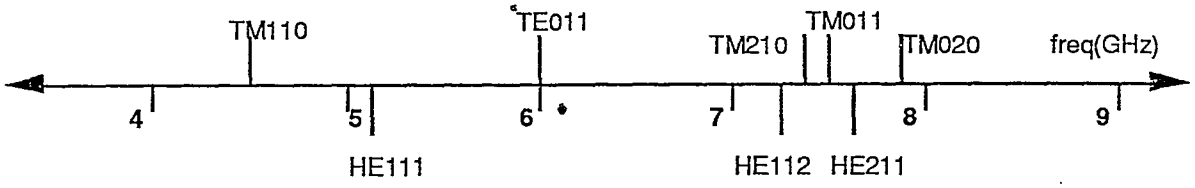
The next step involves the drawing of frequency axis graphs so that the mode separation can be easily seen. The graph is shown on the following two pages. We want to pick a sample which has good mode separation, especially the TM modes as far as possible from TE_{011} modes. By looking at the graph, samples III through VIII can be eliminated from consideration due to the nearness of TM_{0np} modes to the TE_{011} mode. Samples I and II are both good candidates for the measurement. Sample II can be picked over sample I due to its smaller height which leads to lesser radiation losses and also due to the better separation between the TE_{011} mode and the other modes surrounding it.

MODE CHART FOR MATERIAL WITH $\varepsilon_r = 26$ AND FREQUENCY OF RESONANCE OF TE_{011} MODE APPROXIMATELY 6 GHz

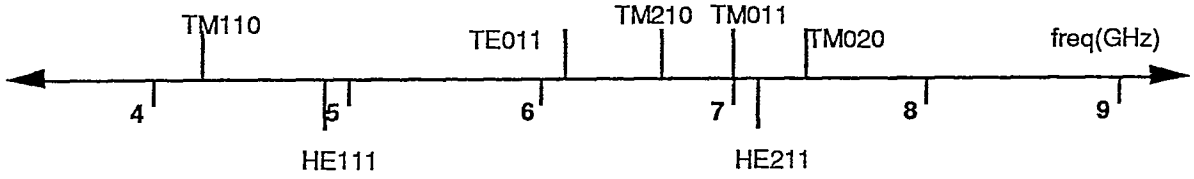
Sample I: D = 350 mils; T = 510 mils; $(D/T)^2 = 0.471$



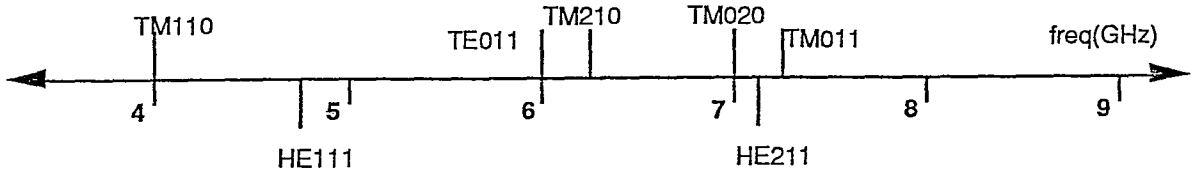
Sample II: D = 375 mils; T = 420 mils; $(D/T)^2 = 0.797$



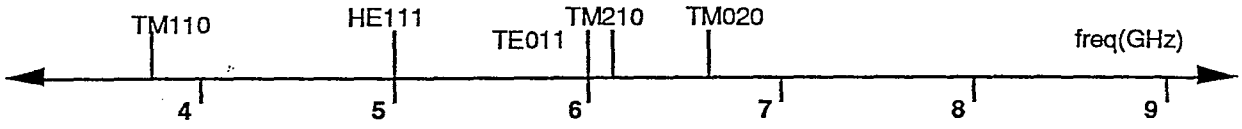
Sample III: D = 400 mils; T = 370 mils; $(D/T)^2 = 1.169$



Sample IV: D = 425 mils; T = 340 mils; $(D/T)^2 = 1.563$

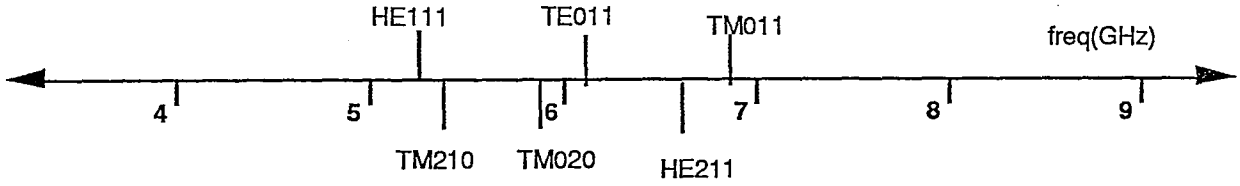


Sample V: D = 450 mils; T = 320 mils; $(D/T)^2 = 1.977$

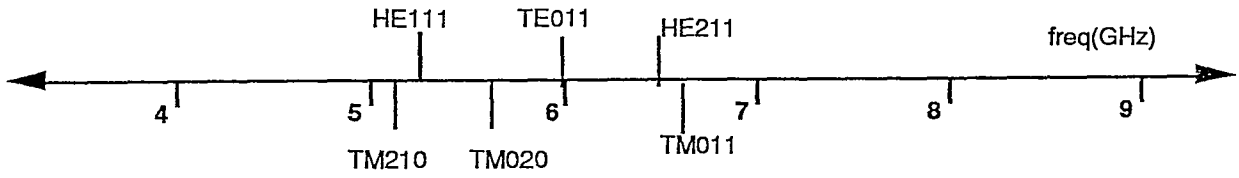


Sample VI: D = 475 mils; T = 300 mils; $(D/T)^2 = 2.507$

Sample VII: D = 500 mils; T = 280 mils; $(D/T)^2 = 3.189$



Sample VIII: D = 525 mils; T = 270 mils; $(D/T)^2 = 3.781$



3. Calibration Process and Calculation of Conductivity of the Plates

As explained in Chapter III, the double height experiment could not be used to calculate the conductivity of the gold plated brass plates. Therefore, in order to calibrate the measurement setup a sample whose properties are well defined with temperature was used. The results obtained from the experiment were used to calculate the conductivity of the gold plated brass plates.

A sample of polycrystalline Zirconium Tin Titanate ($ZrSnTiO_4$) was used for calibration. The cylindrical sample had a diameter of 500 mils (1.27cm) and a height of 272.5 mils (0.692cm). This material was obtained from Murata Erie North America, Inc. The sample is a U series disc resonator. Its relative dielectric constant is expected to remain constant at 38, since it is a temperature compensated material. Table X shows the frequencies of resonance of the various modes associated with this sample. The expected frequency values are obtained from Fig. 10 and the observed frequencies were read off from the network analyzer display in the experiment. The published data and the measured results obtained are given in Table XI and the graph of the relative dielectric constant versus temperature is shown in Fig. 11(a). The raw data is given in Appendix E. The frequency of resonance of TE_{011} mode and the associated Q_0 were calculated using the program 'Q0.FOR'. The ε_r was calculated using the program 'ER.FOR'. For more details on the programs see Appendix C. Figure 10 shows the ε_r increasing slightly as the temperature is lowered, but remaining approximately around thirty eight, as expected.

TABLE X

Mode Frequencies for Polycrystalline $ZrSnTiO_4$ Sample

D = 500 mils

T = 272.5 mils

$(D/T)^2 = 3.367$

$\lambda_0/2 = 1159$ mils

Mode	$\varepsilon_r(D/\lambda)^2$	Expected Frequency (GHz)	Observed Frequency (GHz)
HE_{111}	1.35	4.474	4.48
TM_{210}	1.37	4.507	•
TM_{020}	1.60	4.871	•
TE_{011}	1.78	5.138	5.09
HE_{211}	2.18	5.686	5.78
TM_{310}	2.55	6.150	•
TM_{011}	2.60	6.210	•
EH_{111}	2.70	6.328	6.30
TM_{120}	3.05	6.726	•

For the calculation of the loss tangent, the value of the conductivity of the plates is required. As mentioned before, since the plates are made out of gold plated brass, the conductivity is expected to be in between that of plated gold and brass. The conductivity of the plates at room temperature and at 77K was first calculated using the published data provided by the

company from which the sample was obtained. The expected loss tangent values of polycrystalline $ZrSnTiO_4$ are given in an article by Tamura et al. [19]. The conductivity was calculated using the program 'COND.FOR' (see Appendix C). The values obtained were:

Temperature (K)	Conductivity (mhos/m)
290	$4x10^7$
77	$15x10^7$

The conductivity equation obtained from this data is:

$$\sigma = \frac{0.877\sigma_0}{1 + 0.00339(T - 293)}$$

where $\sigma_0 = 4.562x10^7$ mhos/m is the theoretical conductivity of gold at room temperature [7]. The 0.877 factor is calculated from the ratio of 4.562 to 4.0 and the constant 0.00339 is calculated by using the conductivity at 77K. Table XII shows the conductivity of the plates versus temperature obtained using the above equation along with the theoretical conductivity for bulk gold [7]. Figure 12 shows the graphs of conductivity of gold plated brass and the theoretical conductivity of gold.

Using this conductivity equation, the loss tangent of $ZrSnTiO_4$ was then obtained using the program 'TANDLT.FOR'. A factor of four times improvement in the losses was observed, as expected (see Fig. 11(b)).

Table XI

Polycrystalline $ZrSnTiO_4$ Results

Diameter = 500 mils

Height = 272.5 mils

Expected Relative Dielectric Constant = 38 (Commercially available Temperature Compensated Material)

Temp. (K)	Frequency of TE_{011} (GHz)	Q_0	ε_r	tanδ ($\times 10^{-4}$)	Published tanδ ($\times 10^{-4}$)
291	5.09156	3535.7	37.952	1.0640	1.05
200	•	•	•	•	0.85
120	5.08250	5522.7	38.088	0.6803	•
110	5.08167	5776.2	38.100	0.6441	•
100	5.08129	6051.2	38.106	0.6100	0.60
90	5.07927	6348.0	38.136	0.5785	•
78.5	5.07869	6681.8	38.145	0.5547	0.55

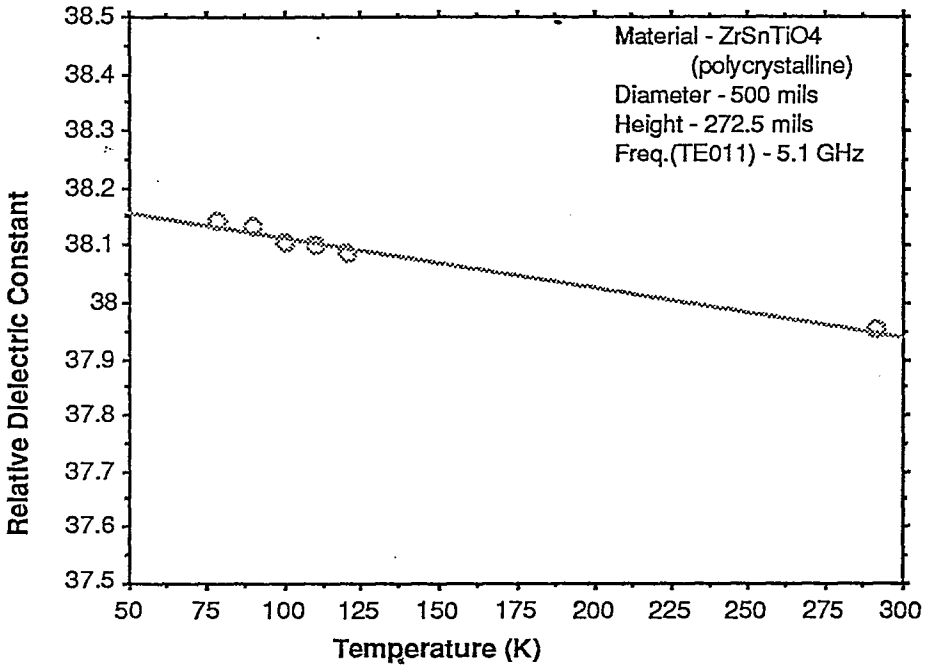


Fig. 11(a) Relative dielectric constant versus temperature for $ZrSnTiO_4$

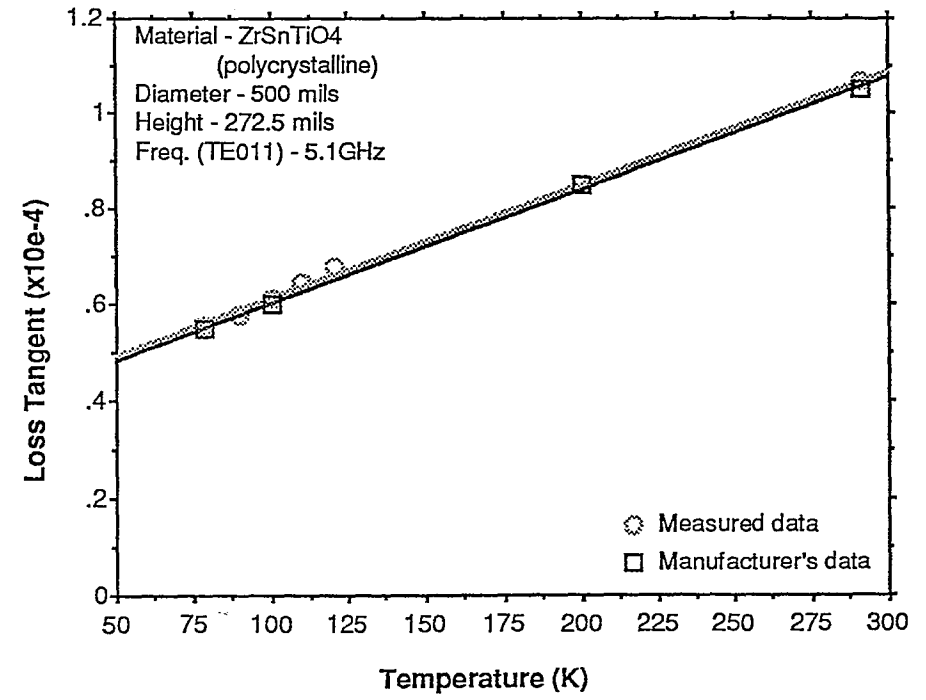


Fig. 11(b) Loss tangent versus temperature for $ZrSnTiO_4$

Table XII

Conductivity of Gold Plated Brass Versus Temperature

Temp. (K)	Calculated conductivity of gold plated brass (mhos/m)	Theoretical conductivity of gold (mhos/m)
293	4.000×10^7	4.562×10^7
120	9.675×10^7	•
110	10.539×10^7	•
100	11.572×10^7	•
90	12.830×10^7	•
77	14.942×10^7	23.084×10^7

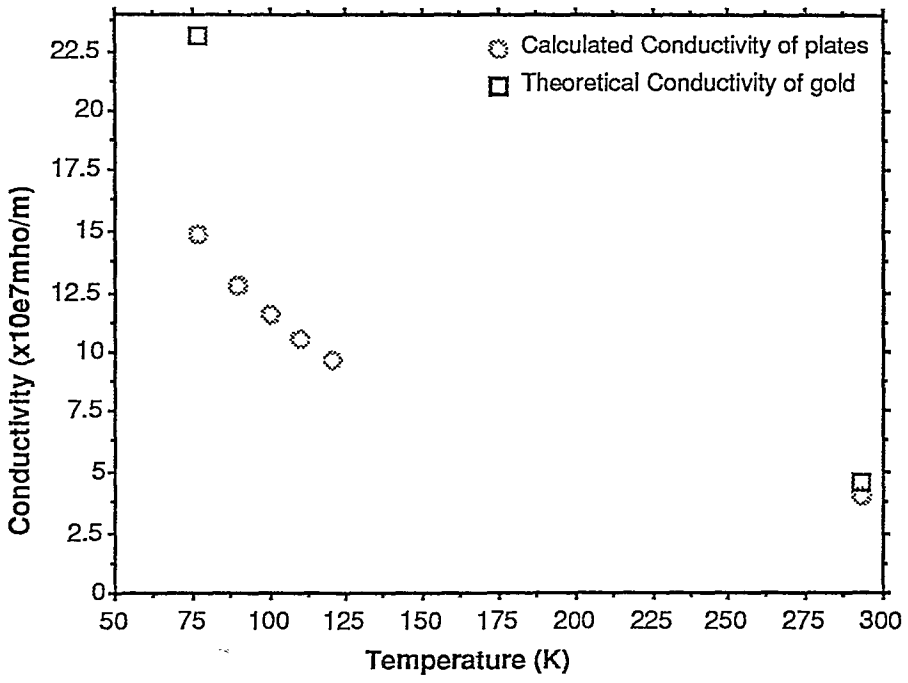


Fig. 12 Conductivity of gold plated brass versus temperature

4. Alumina (99.5%) Results

The next material that was tested in the fixture was Alumina (99.5% Al_2O_3). This material was obtained from McDanel Refractory Co., Beaver Falls, Pennsylvania. The sample diameter was 705.5 mils (1.792 cm) and the height was 353.5 mils (0.898 cm). These dimensions were calculated (using the program 'DESIGN.FOR') to make the resonator have a TE_{011} resonance of 7.4 GHz as can be seen in Table XIII. Table XIII shows the observed frequencies as seen on the Amplitude Display of the Network Analyzer along with the expected frequencies calculated using Fig. 10.

Table XIV shows the results obtained for the relative dielectric constant and the loss tangent as a function of temperature (using programs 'QO.FOR', 'ER.FOR' and 'TANDLT.FOR'). It can be seen that the ε_r does not vary too much over the temperature range whereas the loss tangent improves by a factor of four. The graphs for this data are shown in Figs. 13(a) and 13(b). The raw data is given in Appendix E. The published value of ε_r as being 9.8 at room temperature was obtained from and article by Tamura et al.[20] and the value for the loss tangent from a report by Rockwell International at 7GHz [24].

TABLE XIII

Mode Frequencies for 99.5% Al_2O_3 Sample

D = 705.5 mils

T = 353.5 mils

$(D/T)^2 = 3.983$

Mode	$\varepsilon_r(D/\lambda)^2$	Expected Frequency (GHz)	Observed Frequency (GHz)
TM_{210}	1.35	6.250	•
HE_{111}	1.45	6.477	6.475
TM_{020}	1.59	6.783	•
TE_{011}	1.93	7.473	7.389
HE_{211}	2.28	8.122	8.066
TM_{011}	2.40	8.333	•
TM_{310}	2.53	8.556	•
EH_{111}	2.85	9.081	8.978
TM_{120}	3.00	9.317	•
EH_{311}	3.40	9.919	9.870

Table XIV

99.5% Alumina Results

Diameter = 705.5 mils

Height = 353.5 mils

Temp. (K)	Frequency of TE_{011}(GHz)	Q_0	ε_r	tanδ ($x10^{-4}$)
293	7.38893	4533.6	9.847	1.0021
120	7.43393	8847.9	9.726	0.3557
110	7.43483	9070.4	9.724	0.3571
100	7.43585	9531.8	9.721	0.3333
90	7.43657	10048.7	9.719	0.3098
77	7.43655	11442.1	9.719	0.2291

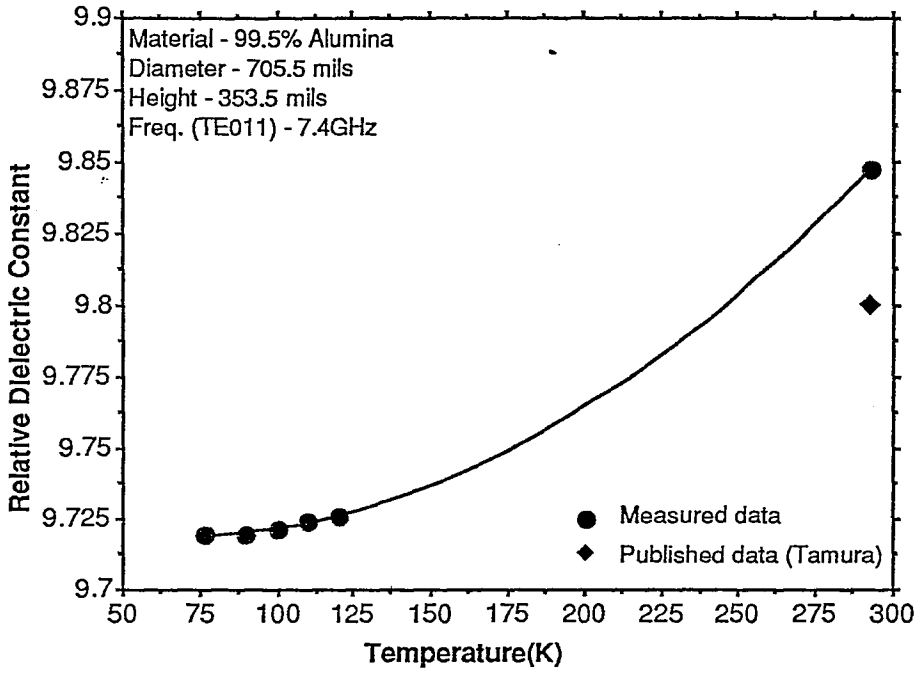


Fig. 13(a) Relative dielectric constant versus temperature for Al_2O_3

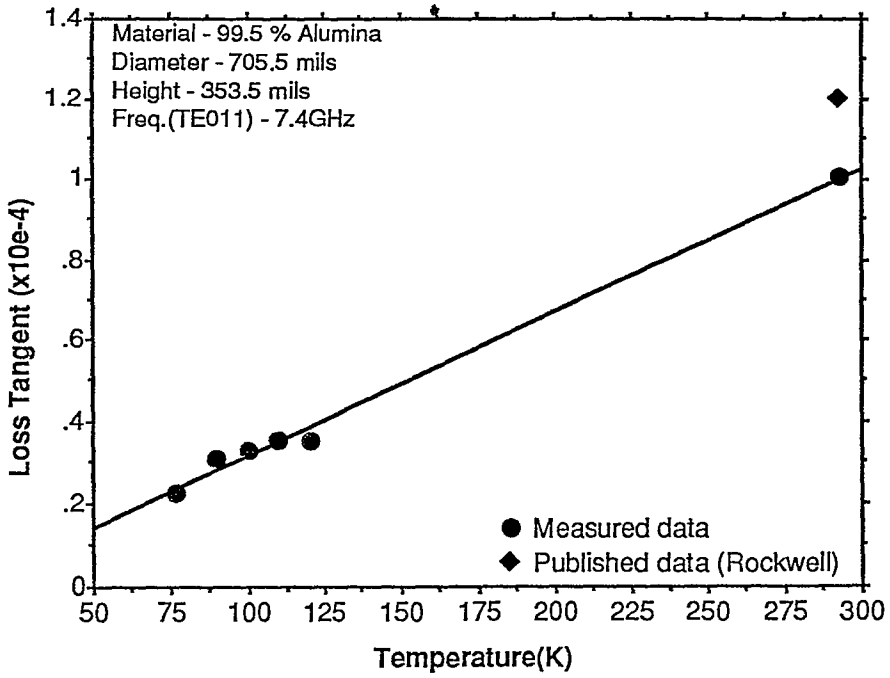


Fig. 13(b) Loss tangent versus temperature for Al_2O_3

5. Magnesium Oxide Results

The Magnesium Oxide sample measured is single crystal with a diameter of 388 mils (0.985 cm) and a height of 362 mils (0.919 cm). The sample was obtained from GFI Advanced Technologies, Forest Hills, NY. These dimensions resulted in a TE_{011} resonance of approximately 10 GHz (see Table XV). This table shows the frequencies calculated from Fig. 10 and the various mode resonances observed on the display of the Network Analyzer.

The sample was inhomogeneous as its Q_0 varied with rotation of the sample in relation to the coupling loop. The variation of Q_0 was within the range of 7000 - 9000. Hence the experiment was repeated at two different orientations to obtain an average value (see Appendix E for details on the averaging process and the raw data). Table XVI shows the results obtained for the relative dielectric constant and the loss tangent as a function of temperature using the programs 'Q0.FOR', 'ER.FOR' and 'TANDLT.FOR'. It can be seen that the ε_r decreases slightly as the temperature is lowered and the loss tangent improves only slightly too. This table also shows the published results for the relative dielectric constant at 1-10KHz frequency. The published results were obtained from a 1973 article by Bartels and Smith [8]. A single point ($\varepsilon_r = 9.83$ at 1KHz) is also shown on the relative dielectric constant graph. This point was obtained from an article by Fontanella et al. [14] . The graphs for the relative dielectric constant and the loss tangent versus temperature are shown in Figs. 14(a) and 14(b).

Besides the inhomogeneity problem, another problem encountered with the MgO sample was its deterioration property with passage of time. Hence the sample had to be stored in vacuum or in a gaseous nitrogen environment. When the sample was first received, it had an approximate Q_0 of 9000 at room temperature. When the same sample was measured two and a half months later, its Q_0 had depreciated to about 5000. The results quoted in the following tables are those which were obtained from a second sample which had been stored in vacuum.

TABLE XV

Mode Frequencies for Single Crystal MgO Sample

D = 388 mils

T = 362 mils

$(D/T)^2 = 1.149$

Mode	$\varepsilon_r(D/\lambda)^2$	Expected Frequency (GHz)	Observed Frequency (GHz)
TM_{110}	0.55	7.248	•
HE_{111}	0.75	8.464	8.468
TE_{011}	1.05	10.015	9.941
TM_{210}	1.35	11.356	•
TM_{011}	1.45	11.770	•
HE_{211}	1.50	11.970	12.46

Table XVI

MgO (Single Crystal) Results

Diameter = 388 mils

Height = 362 mils

Temp. (K)	Frequency of TE_{011}(GHz)	Q_0	ε_r	Published ε_r	tanδ ($x10^{-4}$)
300	1-10KHz	•	•	9.956	•
290	9.941030	8647.5	9.802		0.5074
210	1-10KHz	•	•	9.876	•
150	1-10KHz	•	•	9.823	•
120	10.02422	11563.3	9.632	•	0.4514
110	10.02719	12334.3	9.626	9.797	0.4112
100	10.02970	12227.3	9.621	•	0.4359
90	10.03193	12164.4	9.616	•	0.4580
77	10.03323	13200.3	9.613	•	0.4141
70	1-10KHz	•	•	9.783	•

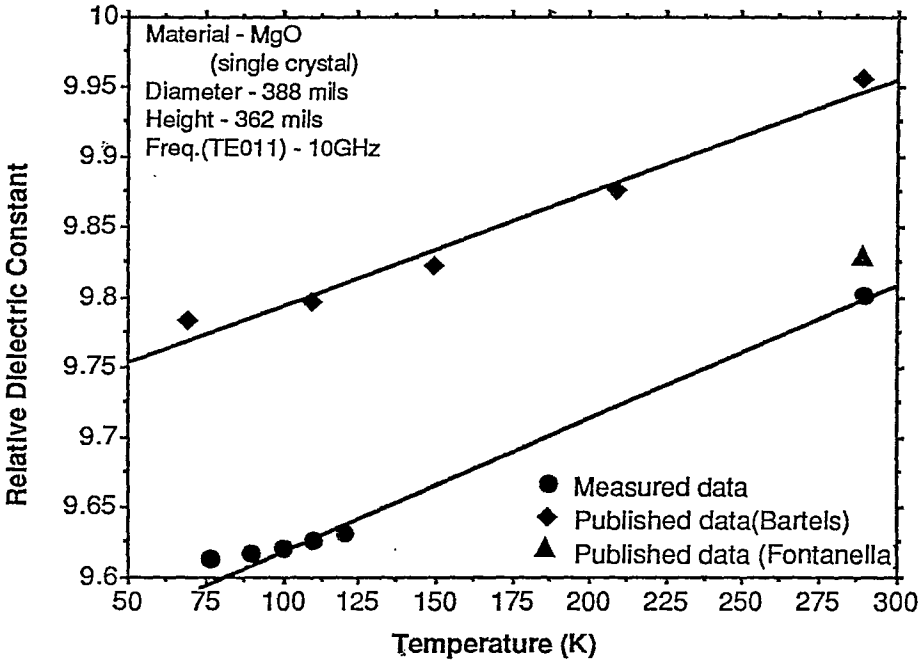


Fig. 14(a) Relative dielectric constant versus temperature for MgO

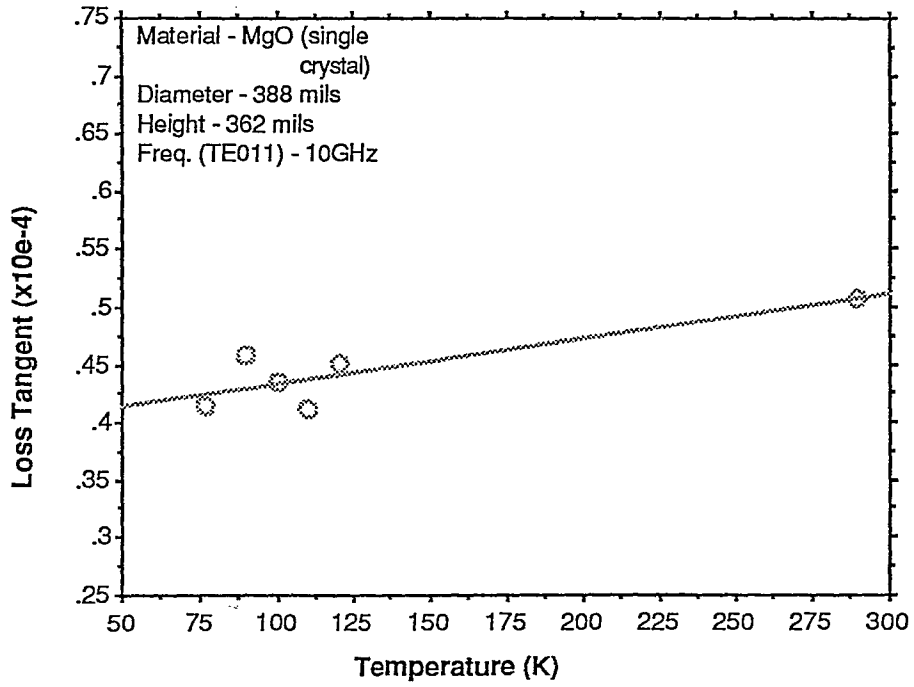


Fig. 14(b) Loss tangent versus temperature for MgO

6. Lanthanum Aluminate (Single Crystal) Results

Lanthanum Aluminate ($LaAlO_3$) is considered to be another potential material for superconductor substrates. A single crystal sample of this material was obtained from North Carolina production facility of Airtron-Litton. The sample obtained was from the end of a boule. It had an approximate diameter of 2 inches and a height of an inch. From the physical appearance of the sample it could be seen that it had a four fold symmetry. A photograph of the samples four fold symmetry as seen by cross polarization is shown in Fig. 15.

From the original sample a cylindrical sample was cut from one of its quarters. This sample had a diameter of 375 mils (0.952 cm) and a height of 418.5 mils (1.063 cm). When this sample was put in the fixture and rotated in its orientation with respect to the coupling loop, a change in the sample Q_0 was observed (like the MgO sample), even though the frequency of resonance remained constant. This proved that the sample was inhomogeneous. Figure 16 shows the change in the measured Q_0 with rotation. As can be seen from the figure, the Q_0 varies over a much wider range of values, unlike the MgO sample.

Table XVII shows the frequencies of resonances of the modes of interest along with the observed frequencies.

A sample position, for which the measured Q_0 was an average value of 8900, was chosen. At this position, the measurements were carried out to obtain the results shown in Figs. 17(a) and 17(b). Table XVIII shows the results obtained for the sample. The ε_r decreases slightly over the measured temperature range (see Fig. 17). Figure 17 also shows a published result for the ε_r (equal to 23.5) at room temperature and 10GHz frequency [11]. The loss tangent obtained does not show much variation as the temperature is decreased.

Due to the extreme inhomogeneity of the sample, the loss tangent results cannot be stated with certainty. If another orientation of the sample were to be chosen, the results would be different though with the same trend.

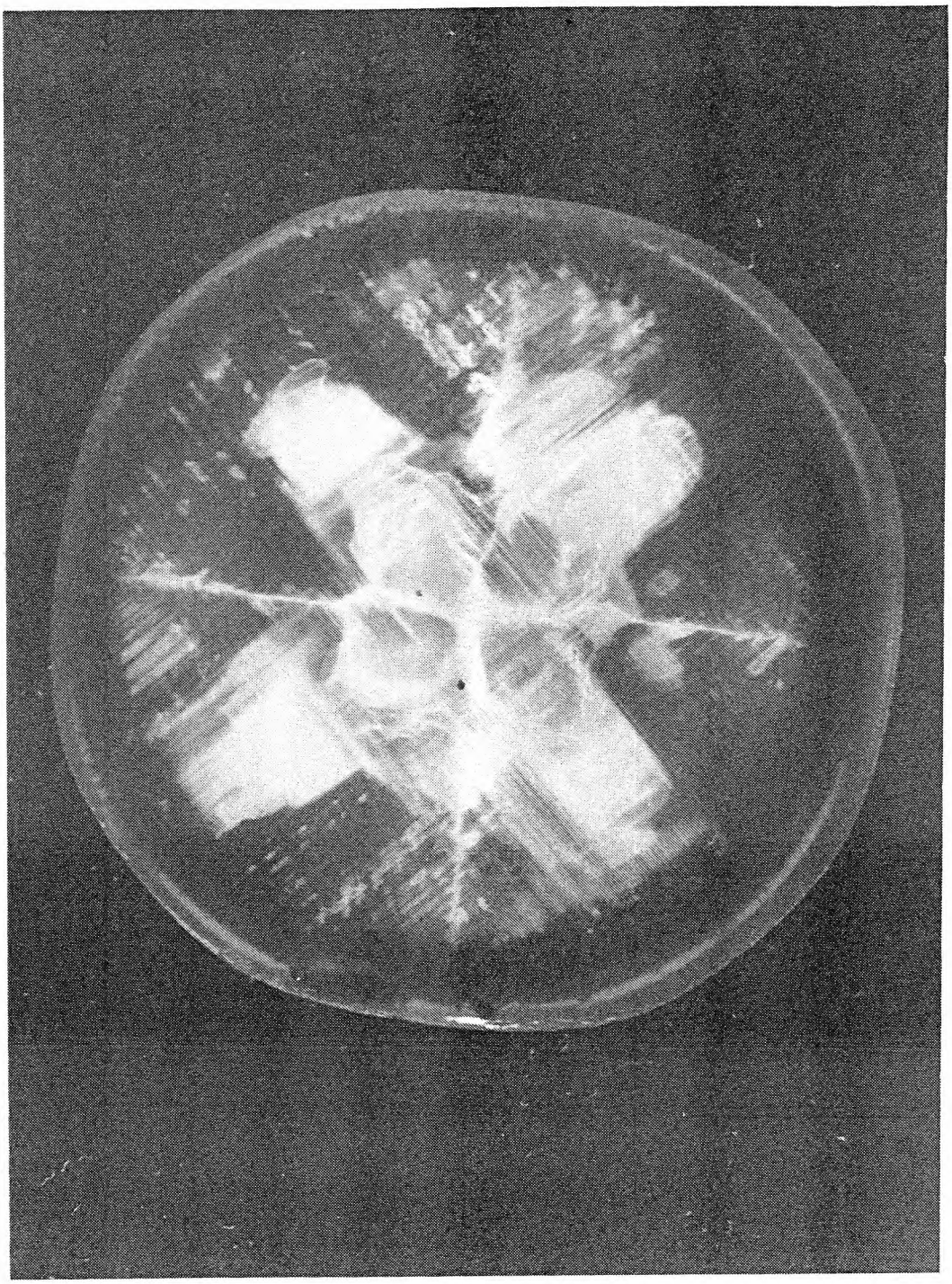


Fig. 15 Photograph of four fold symmetry of $LaAlO_3$ sample

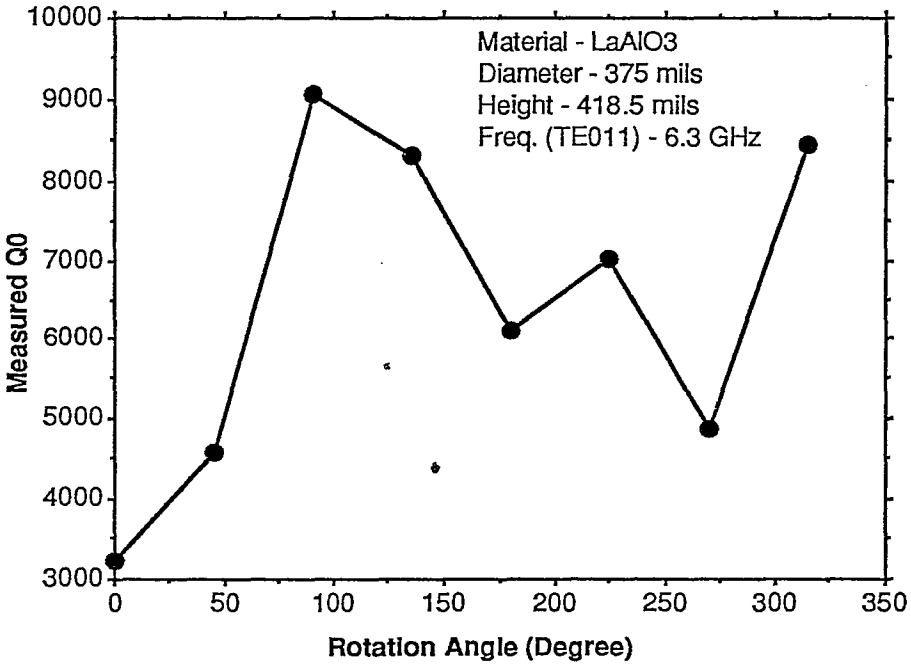


Fig. 16 Unloaded Q versus angle of rotation for $LaAlO_3$

TABLE XVII

Mode Frequencies for Single Crystal $LaAlO_3$ Sample

D = 375 mils

T = 418.5 mils

$(D/T)^2 = 0.803$

Mode	$\varepsilon_r(D/\lambda)^2$	Frequency of Resonance (GHz)
TM_{110}	0.55	4.581
HE_{111}	0.70	5.168
TE_{011}	0.95	6.020
HE_{112}	1.40	7.308
TM_{210}	1.45	7.438
TM_{011}	1.50	7.565
HE_{211}	1.55	7.690
TM_{020}	1.65	7.934

Table XVIII

$LaAlO_3$ (Single Crystal) Results

Diameter = 375 mils

Height = 418.5 mils

Temp. (K)	Frequency of TE_{011}(GHz)	Q_0	ε_r	tanδ (x10^{-4})
293	6.23676	8909.7	24.207	0.5349
120.0	6.28598	10832.1	23.821	0.5533
100.0	6.28967	10663.3	23.792	0.598
90.0	6.29165	11433.7	23.777	0.5493
77.0	6.29295	11868.1	23.767	0.5378

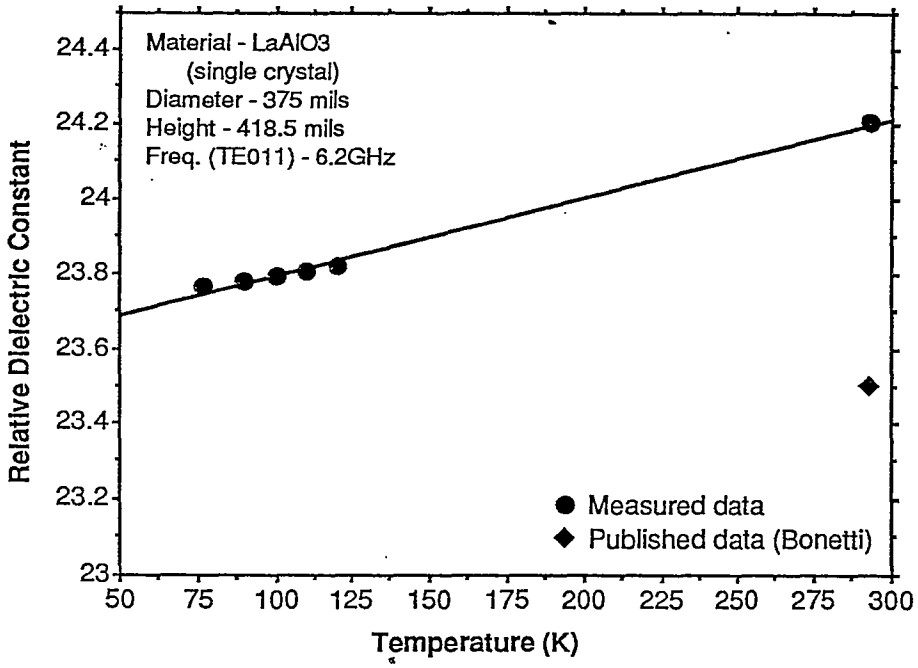


Fig. 17(a) Relative dielectric constant versus temperature for $LaAlO_3$

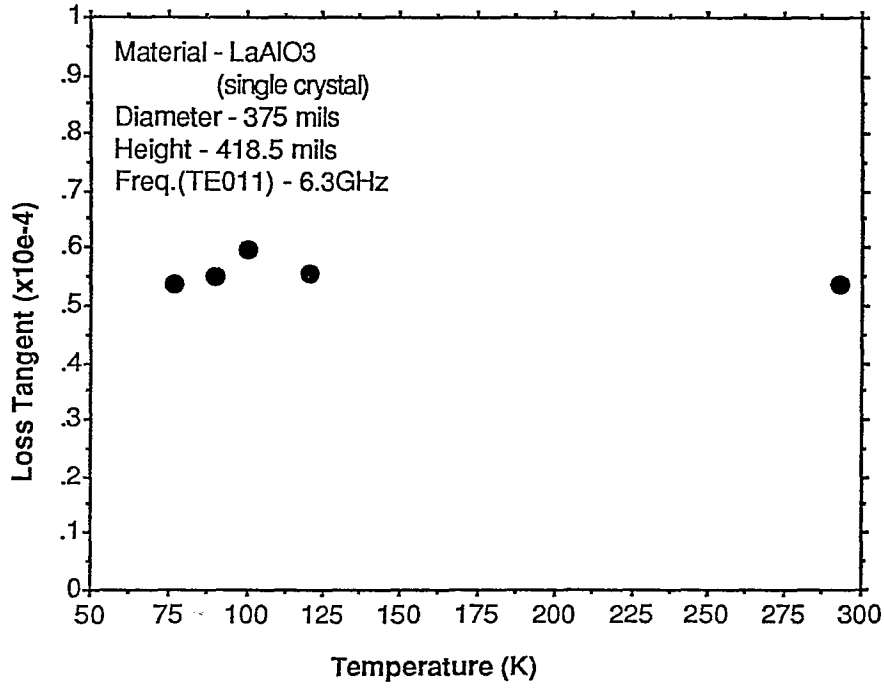


Fig. 17(b) Loss tangent versus temperature for $LaAlO_3$

CHAPTER V

CONCLUSION

Courtney's experimental fixture was modified for cryogenic measurements. The fixture was successfully used to measure the complex dielectric constant of homogeneous, isotropic,non-magnetic, cylindrical samples at microwave frequencies and versus cryogenic temperatures.

The polycrystalline ZrSnTiO4 sample was used to calibrate the conductivity of the gold plated brass conducting plates and then the fixture was used to measure the properties of 99.5% Al2O3, single crystal MgO and single crystal LaAlO3.

A summary of the results obtained at 300K and 77K is given below:

Material	ε_r (300K)	ε_r (77K)	tanδ(x10^{-4})(300K)	tanδ(x10^{-4})(77K)
Al_2O_3 (99.5%)	9.847	9.719	1.0021	0.2291
MgO (single crystal)	9.802	9.613	0.50741	0.41414
$LaAlO_3$ (single crystal)	24.207	23.767	•	•

It can be seen that MgO seems to exhibit lower dielectric losses at higher temperatures as compared to Alumina. Hence MgO appears to be better suited for high temperature superconductor deposition. But MgO also degrades very rapidly if it is kept in open atmosphere. Therefore, depending on the superconductor and its transition temperature, it might be advisable to use the 99.5% Alumina.

The $LaAlO_3$ sample was found to be extremely inhomogeneous and hence its tanδ could not be evaluated accurately. Even though the dielectric losses for this material cannot be stated for certain, it appears that $LaAlO_3$ might be usable as a substrate material.

APPENDIX A

DERIVATION OF THE TRANSCENDENTAL EQUATION

The transcendental equation for the experimental fixture can be obtained using the four basic Maxwell's equations for a linear, homogeneous and isotropic medium. The Maxwell's equations are as follows:

$$\nabla \times \mathbf{E} = -j\omega\mu\,\mathbf{H} \tag{1}$$

$$\nabla \times \mathbf{H} = j\omega\varepsilon\,\mathbf{E} + \mathbf{J} \tag{2}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{3}$$

and $$\nabla \cdot \varepsilon\,\mathbf{E} = q \tag{4}$$

Here, $$\mathbf{J} = \sigma\,\mathbf{E} \tag{5}$$

$$\mathbf{B} = \mu\,\mathbf{H} \tag{6}$$

and $$\varepsilon = \varepsilon_0\,(\varepsilon_r - j\,(\sigma / \omega\,\varepsilon_0)) \tag{7}$$

where **E** - Electric field vector

H - Magnetic field vector

J - current density vector

q - charge density vector

B - magnetic charge density

σ - conductivity

μ - permeability

ε - permittivity

ε_0 - permittivity of free space = 8.854×10^{-12} F/m

ε_r - relative permittivity or relative dielectric constant of the material

ω - frequency in radians / cycles ($\omega = 2\pi f$)

In a source free region, (i.e. $\mathbf{J} = 0$) and for a loss free case (i.e. $\sigma = 0$), the above equations become:

$$\nabla \times \mathbf{E} = -j\,\omega\,\mu\,\mathbf{H} \tag{8}$$

$$\nabla \times \mathbf{H} = j\,\omega\,\varepsilon\,\mathbf{E} \tag{9}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{10}$$

and $$\nabla \cdot \mathbf{E} = 0 \tag{11}$$

To obtain independent Electric and Magnetic field equations, take the curl of both sides of equation (8) to obtain equation (12).

$$\nabla \times (\nabla \times \mathbf{E}) = \nabla \times (-j\,\omega\,\mu\,\mathbf{H})$$

$$= -j\,\omega\,\mu\,(\nabla \times \mathbf{H}) \tag{12}$$

(due to the assumption that the material is isotropic)

Substituting for $\nabla \times \mathbf{H}$ from equation (9) into equation (12)

$$\nabla \times (\nabla \times \mathbf{E}) = -j\,\omega\,\mu\,(j\,\omega\,\mathbf{E})$$

$$= \omega^2\,\mu\,\varepsilon\,\mathbf{E}$$

$$= k^2\,\mathbf{E} \tag{13}$$

where k is called the wave number of the media and is defined as:

$$k = \omega\sqrt{\mu\,\varepsilon} = \frac{2\pi}{\lambda} \tag{14}$$

Similarly, $$\nabla \times (\nabla \times \mathbf{H}) = k^2\,\mathbf{H} \tag{15}$$

Equations (13) and (15) are called the Vector Helmholtz Equations.

For a general vector **A**, the following identity holds:

$$\nabla \times (\nabla \times \mathbf{A}) = \nabla (\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}$$

Substituting this identity for **E** in equation (13),

$$\nabla (\nabla \cdot \mathbf{E}) - \nabla^2 \mathbf{E} = k^2 \mathbf{E} \qquad (16)$$

But, $\nabla \cdot \mathbf{E} = 0$ from equation (11).

Therefore, equation (16) simplifies to

$$\nabla^2 \mathbf{E} + k^2 \mathbf{E} = 0$$

or, $$[\nabla^2 + k^2] \mathbf{E} = 0 \qquad (17)$$

Similarly, for the Magnetic field vector

$$[\nabla^2 + k^2] \mathbf{H} = 0 \qquad (18)$$

Since we are dealing with a cylindrical dielectric resonator, the cylindrical coordinate system is utilized.

Equations (17) and (18) will be utilized to solve for the **z** components of the electric and magnetic fields.

Therefore, $$[\nabla^2 + k^2] \mathbf{E}_z = 0 \qquad (19)$$

$$[\nabla^2 + k^2] \mathbf{H}_z = 0 \qquad (20)$$

These equations are called the wave equations or the scalar Helmholtz equations. The other components of the electric and magnetic fields, i.e. $\mathbf{E}_\phi$, $\mathbf{E}_\rho$, $\mathbf{H}_\phi$ and $\mathbf{H}_\rho$,can be calculated once the **z** components are known.

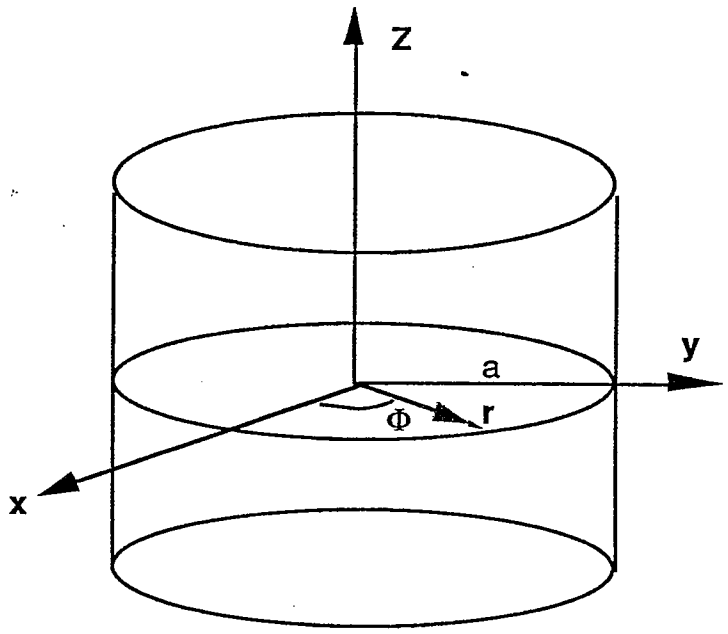


Fig.1A Cylindrical coordinate system

Expanding equation (8) in cylindrical coordinates:

$$\frac{1}{\rho}\frac{\partial}{\partial\phi}(E_z) - \frac{\partial}{\partial z}(E_\phi) = -j\,\omega\,\mu\,H_\rho \tag{21a}$$

$$\frac{\partial}{\partial z}(E_\rho) - \frac{\partial}{\partial \rho}(E_z) = -j\,\omega\,\mu\,H_\phi \tag{21b}$$

$$\frac{1}{\rho}\left[\frac{\partial}{\partial \rho}(\rho E_\phi) - \frac{\partial}{\partial \phi}(E_\rho)\right] = -j\,\omega\,\mu\,H_z \tag{21c}$$

Similarly, expanding equation (9) in cylindrical coordinates:

$$\frac{1}{\rho}\frac{\partial}{\partial\phi}(H_z) - \frac{\partial}{\partial z}(H_\phi) = j\,\omega\,\varepsilon\,E_\rho \tag{22a}$$

$$\frac{\partial}{\partial z}(H_\rho) - \frac{\partial}{\partial \rho}(H_z) = j\,\omega\,\varepsilon\,E_\phi \tag{22b}$$

$$\frac{1}{\rho}\left[\frac{\partial}{\partial \rho}(\rho H_\phi) - \frac{\partial}{\partial \phi}(H_\rho)\right] = j\,\omega\,\varepsilon\,E_z \qquad (22c)$$

Substituting for E_ϕ from equation (22b) into equation (21a):

$$\frac{1}{\rho}\frac{\partial}{\partial \phi}(E_z) - \frac{\partial}{\partial z}\left[\frac{1}{j\omega\varepsilon}\frac{\partial}{\partial z}(H_\rho) - \frac{1}{j\omega\varepsilon}\frac{\partial}{\partial \rho}(H_z)\right] = -j\,\omega\,\mu\,H_\rho$$

or,
$$\frac{1}{\rho}\frac{\partial}{\partial \phi}(E_z) - \frac{1}{j\omega\varepsilon}\frac{\partial^2}{\partial z^2}(H_\rho) + \frac{1}{j\omega\varepsilon}\frac{\partial^2}{\partial \rho \partial z}(H_z) = -j\,\omega\,\mu\,H_\rho$$

or,
$$\left[\frac{1}{j\omega\varepsilon}\frac{\partial^2}{\partial z^2} - j\omega\mu\right] H_\rho = \frac{1}{\rho}\frac{\partial}{\partial \phi}(E_z) + \frac{1}{j\omega\varepsilon}\frac{\partial^2}{\partial \rho \partial z}(H_z)$$

or,
$$\left[\frac{\partial^2}{\partial z^2} + k^2\right] H_\rho = \frac{j\omega\varepsilon}{\rho}\frac{\partial}{\partial \phi}(E_z) + \frac{\partial^2}{\partial \rho \partial z}(H_z) \qquad (23)$$

(since $k^2 = \omega^2\mu\varepsilon$)

Similarly, substitute for H_ρ from equation (21a) into equation (22b):

$$\left[\frac{\partial^2}{\partial z^2} + k^2\right] E_\phi = j\omega\mu\frac{\partial}{\partial \rho}(H_z) + \frac{1}{\rho}\frac{\partial^2}{\partial z \partial \phi}(E_z) \qquad (24)$$

Also, substitute for E_ρ from equation (22a) into equation (21b) to obtain the following equation:

$$\left[\frac{\partial^2}{\partial z^2} + k^2\right] H_\phi = -j\omega\varepsilon\frac{\partial}{\partial \rho}(E_z) + \frac{1}{\rho}\frac{\partial^2}{\partial z \partial \phi}(H_z) \qquad (25)$$

Finally, substitute for H_ϕ from equation (21b) into equation (22a):

$$\left[\frac{\partial^2}{\partial z^2} + k^2\right] E_\rho = \frac{j\omega\mu}{\rho}\frac{\partial}{\partial \phi}(H_z) + \frac{\partial^2}{\partial z \partial \rho}(E_z) \qquad (26)$$

From equations (23) through (26) it can be seen that the ρ and ϕ components of the electric and magnetic fields can be obtained from the $\mathbf{E}_z$ and $\mathbf{H}_z$ components.

To solve for $\mathbf{E}_z$ and $\mathbf{H}_z$, we have to go back and solve equations (19) and (20). This is done by using the method of separation of variables, which is described in the book 'Dielectric Resonators' [3] and can also be found in any mathematics book. The results obtained are:

$$\mathbf{E}_{z1} = \sum_m A_m J_m(k_{\rho 1}\rho) \cos(m\phi)\, e^{-jk_z z} e^{j\omega t} \qquad (\rho<a) \qquad (27a)$$

$$\mathbf{H}_{z1} = \sum_m B_m J_m(k_{\rho 1}\rho) \sin(m\phi)\, e^{-jk_z z} e^{j\omega t} \qquad (\rho<a) \qquad (27b)$$

$$\mathbf{E}_{z2} = \sum_m C_m K_m(k_{\rho 2}\rho) \cos(m\phi)\, e^{-jk_z z} e^{j\omega t} \qquad (\rho<a) \qquad (27c)$$

$$\mathbf{H}_{z2} = \sum_m D_m K_m(k_{\rho 2}\rho) \sin(m\phi)\, e^{-jk_z z} e^{j\omega t} \qquad (\rho<a) \qquad (27d)$$

where $\mathbf{E}_{z1}$ - Electric field vector inside the dielectric

$\mathbf{E}_{z2}$ - Electric field vector outside the dielectric

$\mathbf{H}_{z1}$ - Magnetic field vector inside the dielectric

$\mathbf{H}_{z2}$ - Magnetic field vector outside the dielectric

Note that the fields contain the same coefficient for the z propagation, k_z the waveguide propagation constant, since they must have the same phase along the axis.

The choice of the Bessel function for the various components is due to the Boundary Conditions of the setup. Since the **E** and **H** fields are finite at the center of the sample (at r = 0), the J_m Bessel functions are chosen. For the $\mathbf{E}_{z2}$ and $\mathbf{H}_{z2}$ components, the fields are supposed to be finite at r = a, and then should attenuate exponentially, i.e. no radial propagation of fields.

Hence the K_m Bessel functions are chosen for the fields outside the dielectric sample.

In the above equations, the variables A, B, C and D are constants and

$$k_{\rho 1}^2 = k^2 - k_z^2 \tag{28}$$

and

$$k_{\rho 2}^2 = k_z^2 - k_0^2 \tag{29}$$

where k_0 - free space wave number

$$k = k_0 \sqrt{\varepsilon_r}$$

In equation (32), k_z and k_0 switch places so that the argument of K_m in equations (29) and (32) is real.

Now substituting for $\mathbf{E}_z$ and $\mathbf{H}_z$ from equations (27a) and (27b) into equation (23) to solve for $\mathbf{E}_\phi$:

$$[\frac{\partial}{\partial z^2} + k^2]\, \mathbf{E}_{\phi 1} = j\omega\mu_0 \sum_m B_m \frac{\partial}{\partial \rho} \{\dot{J}_m(k_{\rho 1}\rho)\} \sin(m\phi)\, e^{-jk_z z} e^{j\omega t} + \frac{1}{\rho} \sum_m A_m \{J_m(k_{\rho 1}\rho)\} (-m) \sin(m\phi) (-jk_z)\, e^{-jk_z z} e^{j\omega t}$$

$$[\frac{\partial}{\partial z^2} + k^2]\, \mathbf{E}_{\phi 1} = \sum_m \left[A_m \frac{jk_z m}{\rho} \{J_m(k_{\rho 1}\rho)\} + B_m\, j\omega\mu_0\, (k_{\rho 1}) \frac{\partial}{\partial (k_{\rho 1}\rho)} \{J_m(k_{\rho 1}\rho)\} \right] \sin(m\phi) e^{-jk_z z} e^{j\omega t}$$

$$[\frac{\partial}{\partial z^2} + k^2]\, \mathbf{E}_{\phi 1} = \sum_m \left[A_m \frac{jk_z m}{\rho} \{J_m(k_{\rho 1}\rho)\} + B_m\, j\omega\mu_0 k_{\rho 1} \{J_m'(k_{\rho 1}\rho)\} \right] \sin(m\phi) e^{-jk_z z} e^{j\omega t}$$

$$\mathbf{E}_{\phi 1} = \frac{1}{k^2 - k_z^2} \sum_m \left[A_m \frac{jk_z m}{\rho} \{J_m(k_{\rho 1}\rho)\} + B_m\, j\omega\mu_0 k_{\rho 1} \{J_m'(k_{\rho 1}\rho)\} \right] \sin(m\phi) e^{-jk_z z} e^{j\omega t}$$

$$\mathbf{E}_{\phi 1} = \frac{1}{(k_{\rho 1}^2 + k_z^2) - k_z^2} \sum_m \left[A_m \frac{jk_z m}{\rho} \{J_m(k_{\rho 1}\rho)\} + B_m\, j\omega\mu_0 k_{\rho 1} \{J_m'(k_{\rho 1}\rho)\} \right] \sin(m\phi) e^{-jk_z z} e^{j\omega t}$$

$$\mathbf{E}_{\phi 1} = \frac{1}{k_{\rho 1}^2}\sum_m \left[A_m \frac{jk_z m}{\rho} \{J_m(k_{\rho 1}\rho)\} + B_m j\omega\mu_0 k_{\rho 1} \{J_m'(k_{\rho 1}\rho)\} \right] \sin(m\phi)e^{-jk_z z}e^{j\omega t} \quad (30a)$$

Similarly substituting for $\mathbf{E}_z$ and $\mathbf{H}_z$ from equations (27a) and (27b) into equations (24), (25) and (26) to solve for $\mathbf{H}_\phi$, $\mathbf{E}_\rho$ and $\mathbf{H}_\rho$ for inside the dielectric resonator:

$$\mathbf{H}_{\phi 1} = \frac{-1}{k_{\rho 1}^2}\sum_m \left[jA_m\omega\varepsilon_0\varepsilon_r k_{\rho 1} \{J_m'(k_{\rho 1}\rho)\} + jB_m \frac{mk_z}{\rho} \{J_m(k_{\rho 1}\rho)\} \right] \cos(m\phi)e^{-jk_z z}e^{j\omega t} \quad (30b)$$

$$\mathbf{E}_{\rho 1} = \frac{-1}{k_{\rho 1}^2}\sum_m \left[jA_m k_z k_{\rho 1} \{J_m'(k_{\rho 1}\rho)\} + jB_m \frac{\omega\mu_0 m}{\rho} \{J_m(k_{\rho 1}\rho)\} \right] \cos(m\phi)e^{-jk_z z}e^{j\omega t} \quad (30c)$$

$$\mathbf{H}_{\rho 1} = \frac{1}{k_{\rho 1}^2}\sum_m \left[-jA_m \frac{\omega\varepsilon_0\varepsilon_r m}{\rho} \{J_m(k_{\rho 1}\rho)\} - jB_m k_{\rho 1} k_z \{J_m'(k_{\rho 1}\rho)\} \right] \sin(m\phi)e^{-jk_z z}e^{j\omega t} \quad (30d)$$

Similarly, for outside the dielectric resonator,

$$\mathbf{E}_{\phi 2} = \frac{-1}{k_{\rho 2}^2}\sum_m \left[C_m \frac{jk_z m}{\rho} \{K_m(k_{\rho 2}\rho)\} + D_m j\omega\mu_0 k_{\rho 2} \{K_m'(k_{\rho 2}\rho)\} \right] \sin(m\phi)e^{-jk_z z}e^{j\omega t} \quad (31a)$$

$$\mathbf{H}_{\phi 2} = \frac{1}{k_{\rho 2}^2}\sum_m \left[C_m j\omega\varepsilon_0 k_{\rho 2} \{K_m'(k_{\rho 2}\rho)\} + D_m \frac{jmk_z}{\rho} \{K_m(k_{\rho 2}\rho)\} \right] \cos(m\phi)e^{-jk_z z}e^{j\omega t} \quad (31b)$$

$$\mathbf{E}_{\rho 2} = \frac{1}{k_{\rho 2}^2}\sum_m \left[jC_m k_z k_{\rho 2} \{K_m'(k_{\rho 2}\rho)\} + jD_m \frac{\omega\mu_0 m}{\rho} \{K_m(k_{\rho 2}\rho)\} \right] \cos(m\phi)e^{-jk_z z}e^{j\omega t} \quad (31c)$$

$$\mathbf{H}_{\rho 2} = \frac{1}{k_{\rho 2}^2}\sum_m \left[jC_m \frac{\omega\varepsilon_0 m}{\rho} \{K_m(k_{\rho 2}\rho)\} + jB_m k_{\rho 2} k_z \{K_m'(k_{\rho 2}\rho)\} \right] \sin(m\phi)e^{-jk_z z}e^{j\omega t} \quad (31d)$$

Now, considering the boundary conditions which come about due to the continuity of the fields tangential to the boundary between the dielectric resonator and the surrounding air:

$$\mathbf{E}_{z1} = \mathbf{E}_{z2} \qquad (\rho = a) \quad (32a)$$

$$\mathbf{H}_{z1} = \mathbf{H}_{z2} \qquad (\rho = a) \quad (32b)$$

$$\mathbf{E}_{\phi 1} = \mathbf{E}_{\phi 2} \qquad (\rho = a) \quad (32c)$$

and $$\mathbf{H}_{\phi 1} = \mathbf{H}_{\phi 2} \qquad (\rho = a) \tag{32d}$$

Substituting equations (27a) through (27d) in equations (32a) and (32b), we get:

$$A_m J_m(k_{\rho 1}a) = C_m K_m(k_{\rho 2}a) \tag{33a}$$

$$B_m J_m(k_{\rho 1}a) = D_m K_m(k_{\rho 2}a) \tag{33b}$$

Now, substituting equations (30a), (30b), (31a) and (31b) into equations (32c) and (32d) with $\rho = a$:

$$\frac{k_z m}{k_{\rho 1}^2 a} A_m J_m(k_{\rho 1}a) + \frac{\omega\mu_0}{k_{\rho 1}} B_m J_m'(k_{\rho 1}a) = \frac{-k_z m}{k_{\rho 2}^2 a} C_m K_m(k_{\rho 2}a) - \frac{\omega\mu_0}{k_{\rho 2}} D_m K_m'(k_{\rho 2}a) \tag{33c}$$

$$\frac{\omega\varepsilon_0\varepsilon_r}{k_{\rho 1}} A_m J_m'(k_{\rho 1}a) + \frac{k_z m}{k_{\rho 1}^2 a} B_m J_m(k_{\rho 1}a) = \frac{-\omega\varepsilon_0}{k_{\rho 2}} C_m K_m'(k_{\rho 2}a) - \frac{k_z m}{k_{\rho 2}^2 a} D_m K_m(k_{\rho 2}a) \tag{33d}$$

The unknown k_z can be written in terms of variables k_0 and k from equations (28) and (29).

Now let, $$\alpha = k_{\rho 1}a \tag{34a}$$

$$\beta = k_{\rho 2}a$$

or, $$\beta = a\sqrt{k_z^2 - k_0^2} \qquad \text{(from equation (29))}$$

or, $$\beta = a\sqrt{(k^2 - k_{\rho 1}^2) - k_0^2}$$

or, $$\beta = a\sqrt{k_0^2(\varepsilon_r - 1) - k_{\rho 1}^2}$$

or, $$\beta = \sqrt{(k_0 a)^2(\varepsilon_r - 1) - \alpha^2} \tag{34b}$$

Also, $$k_z a = a\sqrt{k_0^2\varepsilon_r - k_{\rho 1}^2}$$

or, $$k_z a = \sqrt{(k_0 a)^2\varepsilon_r - \alpha^2} \tag{34c}$$

Substituting equations (34a) through (34c) into equations (33a) through (33d), we get:

$$A_m J_m(\alpha) - C_m K_m(\beta) = 0 \tag{35a}$$

$$B_m J_m(\alpha) - D_m K_m(\beta) = 0 \tag{35b}$$

$$\frac{k_z ma}{\alpha^2} A_m J_m(\alpha) + \frac{\omega\mu_0 a}{\alpha} B_m J_m'(\alpha) + \frac{k_z ma}{\beta^2} C_m K_m(\beta) + \frac{\omega\mu_0 a}{\beta} D_m K_m'(\beta) = 0 \tag{35c}$$

$$\frac{\omega\varepsilon_0\varepsilon_r a}{\alpha} A_m J_m'(\alpha) + \frac{k_z ma}{\alpha^2} B_m J_m(\alpha) + \frac{\omega\varepsilon_0 a}{\beta} C_m K_m'(\beta) + \frac{k_z ma}{\beta^2} D_m K_m(\beta) = 0 \tag{35d}$$

Writing equation (35a) through equation (35d) in determinant form to solve:

$$\begin{vmatrix} J_m(\alpha) & 0 & -K_m(\beta) & 0 \\ 0 & J_m(\alpha) & 0 & -K_m(\beta) \\ \frac{k_z ma}{\alpha^2} J_m(\alpha) & \frac{\omega\mu_0 a}{\alpha} J_m'(\alpha) & \frac{k_z ma}{\beta^2} K_m(\beta) & \frac{\omega\mu_0 a}{\beta} K_m'(\beta) \\ \frac{\omega\varepsilon_0\varepsilon_r a}{\alpha} J_m'(\alpha) & \frac{k_z ma}{\alpha^2} J_m(\alpha) & \frac{\omega\varepsilon_0 a}{\beta} K_m'(\beta) & \frac{k_z ma}{\beta^2} K_m(\beta) \end{vmatrix} = 0$$

Solving the above determinant,

$$\left(k_z ma J_m(\alpha) K_m(\beta)\right)^2 \left(\frac{1}{\alpha^4} + \frac{1}{\beta^4}\right) + 2\left(\frac{J_m(\alpha) K_m(\beta) k_z ma}{\alpha\beta}\right)^2 - \omega^2 a^2 \varepsilon_0\mu_0 \left(\frac{J_m^2(\alpha) K_m'^2(\beta)}{\beta^2} + \frac{J_m'^2(\alpha) K_m^2(\beta)\varepsilon_r}{\alpha^2}\right)$$

$$- \frac{\omega^2 a^2 \mu_0 \varepsilon_0 J_m(\alpha) K_m(\beta) J_m'(\alpha) K_m'(\beta)}{\alpha\beta}\left(1 + \varepsilon_r\right) = 0$$

Dividing through by $-K_m^2(\beta)\omega^2\mu_0\varepsilon a^2$ and rearranging, we get

$$F_1(\alpha)F_2(\alpha) - F_3^2(\alpha) = 0 \tag{36}$$

where
$$F_1(\alpha) = \frac{J_m'(\alpha)}{\alpha} + \frac{J_m(\alpha)K_m'(\beta)}{\varepsilon_r \beta K_m(\beta)} \tag{37a}$$

$$F_2(\alpha) = \frac{J_m'(\alpha)}{\alpha} + \frac{J_m(\alpha)K_m'(\beta)}{\beta K_m(\beta)} \tag{37b}$$

and
$$F_3(\alpha) = \frac{k_z m J_m(\alpha)}{k_0 \sqrt{\varepsilon_r}} \left(\frac{1}{\alpha^2} + \frac{1}{\beta^2}\right) \tag{37c}$$

Multiplying by $(\varepsilon_r / J_m^2(\alpha))$, we obtain the transcendental equation of Courtney's article:

$$\left[\frac{\varepsilon_r J_m'(\alpha)}{\alpha J_m(\alpha)} + \frac{K_m'(\beta)}{\beta K_m(\beta)}\right]\left[\frac{J_m'(\alpha)}{\alpha J_m(\alpha)} + \frac{K_m'(\beta)}{\beta K_m(\beta)}\right] = m^2\left[\frac{\varepsilon_r}{\alpha^2} + \frac{1}{\beta^2}\right]\left[\frac{1}{\alpha^2} + \frac{1}{\beta^2}\right]$$

APPENDIX B

MEASUREMENT OF FREQUENCY OF RESONANCE AND UNLOADED Q

On the Network Analyzer Amplitude display, the frequency spectrum is seen as a straight line. A notch is seen on the amplitude display at the resonance frequency of the various modes. On the phase display (equivalent to a Smith Chart), the frequency spectrum is seen as a full circle. For each resonance, a smaller circle enclosed inside the large one is seen on this display.

For the experiment, the amplitude display was utilized to calculate the resonance frequency (f_0) and the unloaded Q (Q_0), since it was found to be more reliable and easier to read as compared to the phase display.

The process of measurement of the f_0 and the Q_0 is derived from a 1976 article by Aitken [1]. The top of the notch is used as the reference level. The bottom of the notch is called the minimum level D_m (a negative number in dB). The frequency value at this lowest point is the f_0 of the mode of measurement.

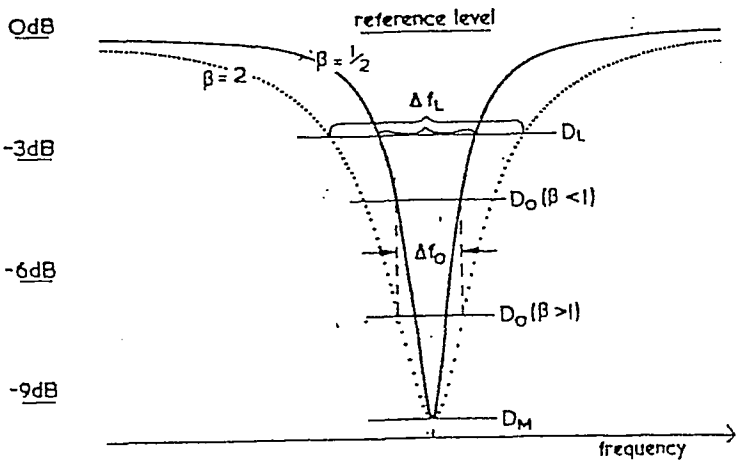


Fig. 2A Resonance as seen on amplitude display of network analyzer

The first step in the calculation of the Q_0 is determining whether the coupling loops are over or undercoupled to the resonator. This can be done by looking at the phase display and measuring the point of intersection of the resonance circle with the horizontal axis. The number on the horizontal axis at the point of intersection is equivalent to the coupling coefficient, β. If β is greater than unity, then the resonator is overcoupled and vice versa. The exact value of β can be determined by using the formulas,

$$\beta = \frac{1 + \text{antilog}_{10}\frac{D_m}{20}}{1 - \text{antilog}_{10}\frac{D_m}{20}} \qquad (\beta > 1) \qquad (1)$$

$$\beta = \frac{1 - \text{antilog}_{10}\frac{D_m}{20}}{1 + \text{antilog}_{10}\frac{D_m}{20}} \qquad (\beta < 1) \qquad (2)$$

The next step involves the calculation of the level D_0 whose intersection with the resonance notch gives the value of the two 3dB frequency points which are later used to calculate the unloaded Q, Q_0. The value of the D_0 level is obtained from equation (3):

$$D_0 = 20 \log_{10} \left\{ \sqrt{\frac{(2 - 2\beta + \beta^2)}{(2 + 2\beta + \beta^2)}} \right\} \qquad (3)$$

The value of the two 3dB points at this level gives the two frequency points, f_{lo} and f_{hi}. The center frequency is calculated from the average of f_{lo} and f_{hi}:

$$f_0 = \frac{f_{lo} + f_{hi}}{2} \qquad (4)$$

Q_0 is calculated by:

$$Q_0 = \frac{f_0}{f_{hi} - f_{lo}} \qquad (5)$$

The above calculations are incorporated in the Fortran program 'Q0.FOR'. The first input required by the program is whether β is less than

or greater than unity. This data is read off the polar display of the Network Analyzer. Next, the program asks for the value of D_m (in dB as a positive number). The program then calculates the exact value of β and the value of D_0. From the values of f_{lo} and f_{hi}, f_0 and Q_0 are calculated.

APPENDIX C

FORTRAN PROGRAMS

The main variables used in the programs are:

er Relative dielectric constant

f Frequency of resonance of mode of interest (GHz)

freq Frequency of resonance of mode of interest (Hz)

d Diameter of the sample (mils)

t Height of the sample (mils)

pi Constant π

$a = \pi^2 d^2 er$

$b = \pi^2 d^2 p^2 / 4t^2$

$c = \pi^2 d^2$

$$\text{alpha} = \alpha = \sqrt{\left[\frac{a}{\lambda^2} - b\right]}$$

$$\text{beta} = \beta = \sqrt{\left[b - \frac{c}{\lambda^2}\right]}$$

1. Program 'DESIGN.FOR'

This program is used to calculate the value of the height of the sample necessary for a particular diameter and a particular frequency of resonance for TE_{011} mode. An approximate value of the ε_r of the material also needs to be given.

```
      Program DESIGN
c
*************************************************************************
c ***                    File Name: DESIGN.FOR                          ***
c ***  This program is used to design the samples.  The input data      ***
c ***  needed is the frequency of resonance of TE011 mode, an           ***
c ***  approximate value of the er, and the diameter value.  This       ***
c ***  program finds a range for the height of the material for         ***
c ***  which the function is valid for TE011 resonance and then         ***
c ***  calculates the function in that range with the height            ***
c ***  changing by ten mils every time.  The optimum height for a       ***
c ***  particular diameter is found by calculating the height at        ***
c ***  which the function is zero (with the use of the BISECTOR         ***
c ***  subroutine to satisfy the condition for TE011 resonance.         ***
*************************************************************************
c ***  Declare variables
       Integer          t_min_i,t_max_i
       Real             wlr, func, f, T1, T2, v
       Real             t_low, t_hi, t_min, t_max
       Real             pre_al, pre_b, alpha, beta, func1, func2
       Real             t, d, er, pi, a, b, c, ratj1jz, ratk1kz
       Character        answer
c ***  Open data file, 'DESIGN.DAT', that will store the data as it is
c ***  produced so it can be accessed later on.
       open (unit=20, file='DESIGN.DAT', status='unknown')
c ***  Get input data
       write (6,10)
 10    format(' ENTER RELATIVE DIELECTRIC CONSTANT (REAL #)')
       write(20,10)
       read(5,*) er
       write(20,*) er
       write(6,20)
 20    format(' ENTER FREQUENCY OF RESONANCE OF TE011 MODE (GHz)')
       write(20,20)
       read(5,*) f
       write(20,*) f
 30    write(6,35)
 35    format(' ENTER DIAMETER OF SAMPLE IN MILS (REAL #)')
       write(20,35)
       read(5,*) d
       write(20,*) d
c ***  Define constant pi
       pi = 4.0 * atan(1.0)
c ***  Calculate the wavelength in mils
       wlr = (30000.0 / 2.54) / f
c ***  Calculate the limits for the height of the samples.  The lower
c ***  limit on the height is set by the point where the variable
c ***  alpha is zero. The upper limit is set by the radiation condition.
       t_min = wlr / (2 * sqrt(er))
c ***  Convert variable 't_min' to integer value.
       t_min_i = t_min + 10
       t_max = wlr / 2
c ***  Convert variable 't_max' to integer value.
       t_max_i = t_max
c ***  The value of the variable 'v' is used to determine if it is the
c ***  first run through the DO loop.  If so, the subroutine BISECTOR
c ***  cannot be called since at least two runs through the loop are
c ***  required.
```

```
      v = 0
c ***  Begin DO loop
      Do 80 t = t_min_i, t_max_i, 10
            v = v + 1
            a = (pi ** 2) * (d ** 2) * er
            b = ((pi * d / t) ** 2) / 4
            c = (pi ** 2) * (d ** 2)
c ********** Calculate variables 'alpha' and 'beta'
            pre_al = (a / (wlr ** 2)) - b
            pre_b = b - (c / (wlr ** 2))
            alpha = sqrt(pre_al)
            beta = sqrt(pre_b)
c ********** Call the subroutine J1JZ to calculate the ratio of the
c ********** Bessel functions J0 / J1 for the above calculated alpha.
            call J1JZ(alpha, ratjzj1)
c ********** Call the subroutine K1KZ to calculate the ratio of the
c ********** Bessel functions K0 / K1 for the above calculated beta.
            call K1KZ(beta, ratkzk1)
c ********** Calculate the function, variable 'func'.
            func = (alpha * ratjzj1) + (beta * ratkzk1)
c ********** Output the results.
            write(6,40) t
 40         format(' HEIGHT OF THE SAMPLE = ', f7.2)
            write(20,40) t
            write(6,50) func
 50         format(' FUNCTION = ', f7.2)
            write(20,50) func
            write(6,*) '                                '
            write(20,*) '                               '
            if (v .eq. 1)  go to 77
c ********** If v is greater than one, then save the last two values
c ********** of the function to calculate if a zero crossing is
c ********** between these two values.  If so, then call subroutine
c ********** BISECTOR to calculate the point of zero crossing.
c ********** Also save the corresponding heights.  Note that 't_low'
c ********** corresponds to 'func1' and 't_hi' to 'func2'.
            func1 = func2
            func2 = func
            t_low = t_hi
            t_hi = t
            if (func2 .gt. 0.0) go to 72
            if (func1 .gt. 0.0) go to 75
            go to 77
 72         if (func1 .lt. 0.0) go to 75
            go to 77
 75         call BISECTOR(t_low, t_hi, func1, func2, wlr, d, er)
 77         write(6,*) '                               '
            write(20,*) '                              '
 80         continue
c ********** End DO loop
      write(6,*) ' RESULTS ARE STORED IN FILE DESIGN.DAT'
c ***  Close data file 'DESIGN.DAT'
      close(20)
      read(5,90) answer
      write(6,*)'                                                    '
      write(6,*)'DO YOU WANT TO RUN THE PROGRAM AGAIN FOR A DIFFERENT'
      write(6,*)'DIAMETER WITH THE SAME FREQUENCY OF RESONANCE AND '
      write(6,*)'RELATIVE DIELECTRIC CONSTANT? (Y/N)'
```

```
      write(20,*)'                                                    '
      write(20,*)'DO YOU WANT TO RUN THE PROGRAM AGAIN FOR A DIFFERENT'
      write(20,*)'DIAMETER WITH THE SAME FREQUENCY OF RESONANCE AND '
      write(20,*)'RELATIVE DIELECTRIC CONSTANT? (Y/N)'
 90   format(A1)
      if (answer .eq. 'y') goto 30
      if (answer .eq. 'Y') goto 30
      stop
      end


c *-------**------**-------**-------**-------**-------**-------**---
c *-------**------**-------**-------**-------**-------**-------**---

      Subroutine J1JZ(arg, ratjzj1)
c ***************************************************************************
c ***  This subroutine takes in the input, arg, which in this case   ***
c ***  is the value of the variable alpha which is calculated in the ***
c ***  main program.  The output of the subroutine is the value of   ***
c ***  the ratio of the Bessel functions J0 to J1.                   ***
c ***************************************************************************
c ***  Declare variables.
      Real             t(10), rjz, rj1, ratjzj1, sq_arg
      Integer          n
c ***  Start calculations
      if (arg .gt. 3.0) goto 200
      t(1) = (arg / 3.0) ** 2
c ***  Begin do loop
      do 190 n = 1, 5
 190     t(n+1) = t(n) * t(1)
c ***  End do loop
      rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &      t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &      t(6)
      rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &      * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &      0.00001109 * t(6)
      rj1 = rj1 * arg
      go to 220
 200  continue
      t(1) = 3.0 / arg
c ***  Begin do loop
      do 210 n = 1, 5
 210     t(n+1) = t(n) * t(1)
c ***  End do loop
      fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &     0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &     + 0.00014476 * t(6)
      tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &     0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &     + 0.00013558 * t(6)
      sq_arg = fz / sqrt(arg)
      rjz = sq_arg * cos(tz)
      f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &     0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &     - 0.00020033 * t(6)
      t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &     - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
```

```
     &      - 0.00029166 * t(6)
        sq_arg = f1 / sqrt(arg)
        rj1 = sq_arg * cos(t1)
 220    continue
        ratjzj1 = rjz / rj1
        return
        end

c *-------**-------**-------**-------**-------**-------**-------**-------**---
c *-------**-------**-------**-------**-------**-------**-------**-------**---

        Subroutine K1KZ(arg, ratkzk1)
c ***************************************************************************
c *** This subroutine takes in the input, arg, which in this case is ***
c *** the value of the variable beta which is calculated in the main ***
c *** program.  The output of the subroutine is the value of the     ***
c *** ratio of the Bessel functions K0 to K1.                        ***
c ***************************************************************************
c ***  Declare variables
        Integer          n
        Real             (10), rk1, rkz, ratkzk1
c ***  Start calculations
        if (arg .gt. 2.0) go to 250
        t(1) = (arg / 3.75) ** 2
c ***  Begin do loop
        do 230 n = 1, 5
 230       t(n+1) = t(n) * t(1)
c ***  End do loop
        riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &       t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &       * t(6)
        ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &       t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &       * t(6)
        ri1 = ri1 * arg
        t(1) = (arg / 2.0) ** 2
c ***  Begin do loop
        do 240 n = 1, 5
 240       t(n+1) = t(n) * t(1)
c ***  End do loop
        argh = arg / 2.0
        rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &       0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &       + 0.00010750 * t(5) + 0.00000740 * t(6)
        rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &       0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &       -  0.00110404 * t(5) - 0.00004686 * t(6)
        rk1 = (rk1 / arg)
        go to 270
 250    continue
        t(1) = 2.0 / arg
c ***  Begin do loop
        do 260 n = 1, 5
 260       t(n+1) = t(n) * t(1)
c ***  End do loop
        rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
     &       - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &       t(5) + 0.00053208 * t(6)
```

```
      rkz = rkz / (((arg) ** 0.5) * exp(arg))
      rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &       + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
     &       t(5) - 0.00068245 * t(6)
      rk1 = rk1 / (((arg) ** 0.5) * exp(arg))
 270  continue
      ratkzk1 = rkz / rk1
      return
      end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

      Subroutine BISECTOR(xa, xb, funca, funcb, wlr, d, er)
c ***********************************************************************
c ***  This subroutine calculates the value of the zero crossing      ***
c ***  between the two points provided by the main program when it    ***
c ***  calls this subroutine.                                         ***
c ***********************************************************************
c ***  Declare variables.
      Integer          iterations
      Real             xa, xb, xc, funca, funcb, funcc
      Real             root, t, d, er, f, wlr
c ***  Calculate xc, the midpoint between xa and xb, which will
c ***  become one of the new limits.
      iterations = 0
 280  xc = (xa + xb) / 2.0
      iterations = iterations + 1
c ***  The difference in the sign of xa and xb can be when the function
c ***  goes to plus or minus infinity.  Hence the zero cannot be found.
c ***  If the loop is repeated more than 100 times, it is very likely
c ***  that the case is not that of zero crossing.  Therefore, get out
c ***  of the loop.
      if (iterations .gt. 100) go to 335
c ***  Calculate the value of the function for xc, i.e. funcc.
      call FUNC(xc, d, er, wlr, funcc)
c ***  If the value of the function at both xa and xc are either
c ***  positive or negative, then let the new value of xa be equal to
c ***  xc.  Now calculate the new value of funca (the value of the
c ***  variable func at the point xa).  If on the other hand, the
c ***  value of the function at both xb and xc are of the same sign,
c ***  then let the new value of xb be equal to xc and now calculate
c ***  the new value of funcb(the value of the variable func at point
c ***  xb).  Repeat this till the value of function is approximately
c ***  equal to zero.  Then let the root be equal to xc and output the
c ***  results.
      if (funcc .lt. 0.0000) go to 300
      if (funca .lt. 0.0000) go to 310
 290  xa = xc
      call FUNC(xa, d, er, wlr, funca)
      if (abs(funca) .lt. 0.0001) go to 320
      go to 280
 300  if (funca .lt. 0.0000) go to 290
 310  xb = xc
      call FUNC(xb, d, er, wlr, funcb)
      if (abs(funcb) .lt. 0.0001) go to 320
      go to 280
 320  root = xc
```

```
c ***  Output the results.
       write(6,*)' SAMPLE HEIGHT AT WHICH THE FUNCTION IS '
       write(6,*)' EQUAL TO ZERO = ', root, ' MILS'
       write(6,*)'                                '
       write(20,*)' SAMPLE HEIGHT AT WHICH THE FUNCTION IS '
       write(20,*)' EQUAL TO ZERO = ', root, ' MILS'
       write(20,*)'                               '
 335   return
       end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

       Subroutine FUNC(t, d, er, wlr, function)
c ***********************************************************************
c *** This subroutine calculates the value of the variable function  ***
c *** for the particular height of sample that is provided as an     ***
c *** argument by the subroutine BISECTOR.  It calls upon the        ***
c *** subroutines J1JZ and K1KZ to provide the values of the ratios  ***
c *** of the Bessel functions J0 / J1 and the ratio K0 / K1.         ***
c ***********************************************************************
c ***  Declare variables.
       Real             wlr, t, d, er, pi, a, b, c
       Real             pre_al, pre_b, alpha, beta
       Real             ratj1jz, ratk1kz
c ***  Define constant pi
       pi = 4.0 * atan(1.0)
       a = (pi ** 2) * (d ** 2) * er
       b = ((pi * d / t) ** 2) / 4.0
       c = (pi ** 2) * (d ** 2)
c ***  Calculate variables alpha and beta
       pre_al = (a / (wlr ** 2)) - b
       pre_b = b - (c / (wlr ** 2))
       alpha = sqrt(pre_al)
       beta = sqrt(pre_b)
c ***  Call subroutine J1JZ to calculate the value of the Bessel
c ***  functions ratio J0 /J1.
       call J1JZ(alpha, ratjzj1)
c ***  Call subroutine K1KZ to calculate the value of the Bessel
c ***  functions ratio K0 / K1.
       call K1KZ(beta, ratkzk1)
c ***  Calculate the variable 'function'.
       function = (alpha * ratjzj1) + (beta * ratkzk1)
c ***  Return to subroutine BISECTOR the value of the variable
c ***  'function'.
       return
       end

c ***********************************************************************
c                              END OF PROGRAM
c ***********************************************************************
```

2. Programs 'TE.FOR', 'TM.FOR' and 'HE.FOR'

The transcendental equation for TE_{0np} resonance is:

$$\frac{\alpha J_0(\alpha)}{J_1(\alpha)} + \frac{\beta K_0(\beta)}{K_1(\beta)} = 0$$

The transcendental equation for TM_{0np} resonance is:

$$\frac{\alpha J_0(\alpha)}{\varepsilon_r J_1(\alpha)} + \frac{\beta K_0(\beta)}{K_1(\beta)} = 0$$

The transcendental equation for HE_{1np} resonance is:

$$\left[\frac{\varepsilon_r J_1'(\alpha)}{\alpha J_1(\alpha)} + \frac{K_1'(\beta)}{\beta K_1(\beta)}\right]\left[\frac{J_1'(\alpha)}{\alpha J_1(\alpha)} + \frac{K_1'(\beta)}{\beta K_1(\beta)}\right] - \left[\frac{\varepsilon_r}{\alpha^2} + \frac{1}{\beta^2}\right]\left[\frac{1}{\alpha^2} + \frac{1}{\beta^2}\right] = 0$$

These equations are incorporated in the programs 'TE.FOR', 'TM.FOR' and 'HE.FOR'.

The input data required by the programs is the value of p (third subscript of the mode), the diameter of the sample in mils, the height of the sample in mils, and the relative dielectric constant of the material.

```
      Program TE
c ***************************************************************************
c ***                    File Name: TE.FOR                                  ***
c *** This program finds the values of the function, func, for a           ***
c *** whole range of wavelengths, provided the diameter, height of         ***
c *** the sample, the dielectric constant and the value of p are           ***
c *** provided (for TE0np modes).  It then calculates the                  ***
c *** frequencies at which the various modes (various n's in TE0np)        ***
c *** are resonating.  These are calculated by calculating where           ***
c *** the variable function, func, goes to zero.                           ***
c ***************************************************************************
c **** Declare variables
      Integer          wl, wllo, wlhi, delta
      Real             wlr, wlloa, wlhia, func, f, mode_p, wl1, wl2
      Real             wl_low, wl_hi, v
      Real             pre_al, pre_b, alpha, beta, func1, func2
      Real             t, d, er, pi, a, b, c, ratj1jz, ratk1kz
```

```
c **** Open data file, 'TE.DAT', that will store the data as it is
c **** produced so it can be accessed later on.
      open (unit=20, file='TE.DAT', status='unknown')
c **** Get input data
      write (6,10)
 10   format(' ENTER VALUE OF p FOR TE0np MODE (REAL #)')
      write (20,10)
      read(5,*) mode_p
      write(20,*) mode_p
      write(6,20)
 20   format(' ENTER DIAMETER OF SAMPLE IN MILS (REAL #)')
      write(20,20)
      read(5,*) d
      write(20,*) d
      write(6,30)
 30   format(' ENTER HEIGHT OF SAMPLE IN MILS (REAL #)')
      write(20,30)
      read(5,*) t
      write(20,*) t
      write(6,40)
 40   format(' ENTER RELATIVE DIELECTRIC CONSTANT (REAL #)')
      write(20,40)
      read(5,*) er
      write(20,*) er
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4
      c = (pi ** 2) * (d ** 2)
c **** Define limits, wllo and wlhi, for the range of wavelength to be
c **** analyzed.
      wlloa = sqrt(c / b)
      wllo = wlloa + 1
      wlhia = sqrt(a / b)
      wlhi = wlhia - 1
c **** Define interval, delta, for wavelength at which the calculations
c **** are to be carried out. There will be approximately 150 points
c **** for calculations.
      delta = (wllo + wlhi) / 150
c **** The value of the variable 'v' is used to determine if it is the
c **** first run through the DO loop.  If so, the subroutine BISECTOR
c **** cannot be called since at least two runs through the loop are
c **** required.
      v = 0
c **** Begin DO loop
      Do 80 wl = wllo, wlhi, delta
         v = v + 1
c ******* Convert the wavelength to a real number
         wlr = wl * 1.0
         wl1 = wl2
         wl2 = wlr
c ******* Calculate the frequency in GHz
         f = (30000.0 / 2.54) / wlr
c ******* Calculate variables alpha and beta
         pre_al = (a / (wlr ** 2)) - b
         pre_b = b - (c / (wlr ** 2))
         alpha = sqrt(pre_al)
```

```
          beta = sqrt(pre_b)
c ******* Call the subroutine J1JZ to calculate the ratio of the
c ******* Bessel functions J0 / J1 for the above calculated alpha.
          call J1JZ(alpha, ratjzj1)
c ******* Call the subroutine K1KZ to calculate the ratio of the
c ******* Bessel functions K0 / K1 for the above calculated beta.
          call K1KZ(beta, ratkzk1)
c ******* Calculate the function, variable func.
          func = (alpha * ratjzj1) + (beta * ratkzk1)
c ******* Output the results.
          write(6,50) wlr
 50       format(' WAVELENGTH = ', i4, ' MILS')
          write(20,50) wlr
          write(6,60) f
 60       format(' FREQUENCY = ', f9.5, ' GHz')
          write(20,60) f
          write(6,70) func
 70       format(' FUNCTION = ',f7.2)
          write(20,70) func
          write(6,*) '                              '
          if (v .eq. 1)  go to 77
c ******* If v is greater than one, then save the last two values
c ******* of the function to calculate if a zero crossing is
c ******* between these two values.  If so, then call subroutine
c ******* BISECTOR to calculate the point of zero crossing.
c ******* Also save the corresponding wavelengths.  Note that wl_low
c ******* corresponds to func1 and wl_hi to func2.
          func1 = func2
          func2 = func
          if (func2 .gt. 0.0) go to 72
          if (func1 .gt. 0.0) go to 75
          go to 77
 72       if (func1 .lt. 0.0) go to 75
          go to 77
 75       wl_low = wl1
          wl_hi = wl2
          call BISECTOR(wl_low, wl_hi, func1, func2, t, d, er, mode_p,
     &                  f)
 77       write(20,*) '                              '
 80       continue
c ****    End do loop
       write(6,*) ' RESULTS ARE STORED IN FILE TE.DAT'
c **** Close data file 'TE.DAT'
       close(20)
       stop
       end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---
```

```
      Subroutine J1JZ(arg, ratjzj1)
c ***************************************************************
c **** This subroutine takes in the input, arg, which in this case   ***
c **** is the value of the variable alpha which is calculated in the ***
c **** main program.  The output of the subroutine is the value of   ***
c **** the ratio of the Bessel functions J0 to J1.                   ***
c ***************************************************************
c **** Declare variables.
      Real             t(10), rjz, rj1, ratjzj1, sq_arg
      Integer          n
c **** Start calculations
      if (arg .gt. 3.0) goto 200
      t(1) = (arg / 3.0) ** 2
c **** Begin do loop
      do 190 n = 1, 5
 190     t(n+1) = t(n) * t(1)
c **** End do loop
      rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &       t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &       t(6)
      rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &       * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &       0.00001109 * t(6)
      rj1 = rj1 * arg
      go to 220
 200  continue
      t(1) = 3.0 / arg
c **** Begin do loop
      do 210 n = 1, 5
 210     t(n+1) = t(n) * t(1)
c **** End do loop
      fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &      0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &      + 0.00014476 * t(6)
      tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &      0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &      + 0.00013558 * t(6)
      sq_arg = fz / sqrt(arg)
      rjz = sq_arg * cos(tz)
      f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &      0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &      - 0.00020033 * t(6)
      t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &      - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
     &      - 0.00029166 * t(6)
      sq_arg = f1 / sqrt(arg)
      rj1 = sq_arg * cos(t1)
 220  continue
      ratjzj1 = rjz / rj1
      return
      end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---
```

```
      Subroutine K1KZ(arg, ratkzk1)
c ***************************************************************************
c *** This subroutine takes in the input, arg, which in this case is ***
c *** the value of the variable beta which is calculated in the main ***
c *** program.  The output of the subroutine is the value of the     ***
c *** ratio of the Bessel functions K0 to K1.                        ***
c ***************************************************************************
c **** Declare variables
      Integer          n
      Real             t(10), rk1, rkz, ratkzk1
c **** Start calculations
      if (arg .gt. 2.0) go to 250
      t(1) = (arg / 3.75) ** 2
c **** Begin do loop
      do 230 n = 1, 5
 230     t(n+1) = t(n) * t(1)
c **** End do loop
      riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &      t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &      * t(6)
      ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &      t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &      * t(6)
      ri1 = ri1 * arg
      t(1) = (arg / 2.0) ** 2
c **** Begin do loop
      do 240 n = 1, 5
 240     t(n+1) = t(n) * t(1)
c **** End do loop
      argh = arg / 2.0
      rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &      0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &      + 0.00010750 * t(5) + 0.00000740 * t(6)
      rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &      0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &      -  0.00110404 * t(5) - 0.00004686 * t(6)
      rk1 = (rk1 / arg)
      go to 270
 250  continue
      t(1) = 2.0 / arg
c **** Begin do loop
      do 260 n = 1, 5
 260     t(n+1) = t(n) * t(1)
c **** End do loop
      rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
     &      - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &      t(5) + 0.00053208 * t(6)
      rkz = rkz / (((arg) ** 0.5) * exp(arg))
      rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &      + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
     &      t(5) - 0.00068245 * t(6)
      rk1 = rk1 / (((arg) ** 0.5) * exp(arg))
 270  continue
      ratkzk1 = rkz / rk1
      return
      end
```

```
c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

      Subroutine BISECTOR(xa, xb, funca, funcb, t, d, er, mode_p, f)
c ***********************************************************************
c **** This subroutine calculates the value of the zero crossing     ***
c **** between the two points provided by the main program when it   ***
c **** calls this subroutine.                                        ***
c ***********************************************************************
c **** Declare variables.
      Integer          iterations
      Real             xa, xb, xc, funca, funcb, funcc
      Real             root, t, d, er, f, mode_p
c **** Calculate xc, the midpoint between xa and xb, which will
c **** become one of the new limits.
      iterations = 0
 280  xc = (xa + xb) / 2.0
      iterations = iterations + 1
c **** The difference in the sign of xa and xb can be when the function
c **** goes to plus or minus infinity.  Hence the zero cannot be found.
c **** If the loop is repeated more than 100 times, it is very likely
c **** that the case is not that of zero crossing.  Therefore, get out
c **** of the loop.
      if (iterations .gt. 100) go to 335
c **** Calculate the value of the function for xc, i.e. funcc.
      call FUNC(xc, t, d, er, mode_p, funcc)
c **** If the value of the function at both xa and xc are either
c **** positive or negative, then let the new value of xa be equal to
c **** xc.  Now calculate the new value of funca (the value of the
c **** variable func at the point xa).  If on the other hand, the
c **** value of the function at both xb and xc are of the same sign,
c **** then let the new value of xb be equal to xc and now calculate
c **** the new value of funcb(the value of the variable func at point
c **** xb).  Repeat this till the value of function is approximately
c **** equal to zero.  Then let the root be equal to xc and output the
c **** results.
      if (funcc .lt. 0.0000) go to 300
      if (funca .lt. 0.0000) go to 310
 290  xa = xc
      call FUNC(xa, t, d, er, mode_p, funca)
      if (abs(funca) .lt. 0.0001) go to 320
      go to 280
 300  if (funca .lt. 0.0000) go to 290
 310  xb = xc
      call FUNC(xb, t, d, er, mode_p, funcb)
      if (abs(funcb) .lt. 0.0001) go to 320
      go to 280
 320  root = xc
c **** Output the results.
      write(6,*)' WAVELENGTH AT WHICH THE FUNCTION IS EQUAL TO '
      write(6,*)' ZERO = ', root, ' MILS CORRESPONDING TO '
      write(6,*)' FREQUENCY = ', f, ' GHz'
      write(6,*)'                                '
```

```
      write(20,*)' WAVELENGTH AT WHICH THE FUNCTION IS EQUAL TO '
      write(20,*)' ZERO = ', root, ' MILS CORRESPONDING TO '
      write(20,*)' FREQUENCY = ', f, 'GHz'
      write(20,*)'                                '
 335  return
      end

c *-------**-------**-------**-------**-------**-------**-------**-------**---
c *-------**-------**-------**-------**-------**-------**-------**-------**---

      Subroutine FUNC(wlr, t, d, er, mode_p, function)
c ***************************************************************************
c *** This subroutine calculates the value of the variable function  ***
c *** for the particular wavelength that is provided as an argument  ***
c *** by the subroutine BISECTOR.  It calls upon the subroutines     ***
c *** J1JZ and K1KZ to provide the values of the ratios of the       ***
c *** Bessel functions J0 / J1 and the ratio K0 / K1.                ***
c ***************************************************************************
c **** Declare variables.
      Real             wlr, t, d, er, pi, a, b, c
      Real             pre_a1, pre_b, alpha, beta
      Real             ratj1jz, ratk1kz, mode_p
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4.0
      c = (pi ** 2) * (d ** 2)
c **** Calculate variables alpha and beta
      pre_a1 = (a / (wlr ** 2)) - b
      pre_b = b - (c / (wlr ** 2))
      alpha = sqrt(pre_a1)
      beta = sqrt(pre_b)
c **** Call subroutine J1JZ to calculate the value of the Bessel
c **** functions ratio J0 /J1.
      call J1JZ(alpha, ratjzj1)
c **** Call subroutine K1KZ to calculate the value of the Bessel
c **** functions ratio K0 / K1.
      call K1KZ(beta, ratkzk1)
c **** Calculate the variable 'function'.
      function = (alpha * ratjzj1) + (beta * ratkzk1)
c **** Return to subroutine BISECTOR the value of the variable
c **** 'function'.
      return
      end

c ***************************************************************************
c                              END OF PROGRAM
c ***************************************************************************
```

```
      Program TM
c ***************************************************************************
c ****                    File Name: TM.FOR                                  ***
c *** This program finds the values of the function, func, for a            ***
c *** whole range of wavelengths, provided the diameter, height of          ***
c *** the sample, the dielectric constant and the value of p are            ***
c *** provided (for TM0np modes).  It then calculates the                   ***
c *** frequencies at which the various modes (various n's in TM0np)         ***
c *** are resonating.  These are calculated by calculating where            ***
c *** the variable function, func, goes to zero.  This program does         ***
c *** not work for p = 0, since that makes the variable 'beta'              ***
c *** imaginary and this program is not capable of handling complex         ***
c *** numbers.                                                              ***
c ***************************************************************************
c **** Declare variables.
      Integer          wl, wllo, wlhi, delta
      Real             wlr, wlloa, wlhia, func, f, mode_p, wl1, wl2
      Real             wl_low, wl_hi, func1, func2, v
      Real             t, d, er, pi, a, b, c, ratjzj1, ratkzk1
      Real             pre_al, pre_b, alpha, beta
c **** Open data file, 'TM.DAT', that will store the data as it is
c **** produced so it can be accessed later on.
      open (unit=20, file='TM.DAT', status='unknown')
c **** Get input data
      write(6,10)
 10   format(' ENTER VALUE OF p FOR TM0np MODE')
      write(20,10)
      read(5,*) mode_p
      write(20,*) mode_p
      write(6,20)
 20   format(' ENTER DIAMETER OF SAMPLE IN MILS (REAL #)')
      write(20,20)
      read(5,*) d
      write(20,*) d
      write(6,30)
 30   format(' ENTER HEIGHT OF SAMPLE IN MILS (REAL #)')
      write(20,30)
      read(5,*) t
      write(20,*) t
      write(6,40)
 40   format(' ENTER RELATIVE DIELECTRIC CONSTANT (REAL #)')
      write(20,40)
      read(5,*) er
      write(20,*) er
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4.0
      c = (pi ** 2) * (d ** 2)
c **** Define limits, wllo and wlhi, for the range of wavelength to be
c **** analyzed.
      wlloa = sqrt(c / b)
      wllo = wlloa + 1
      wlhia = sqrt(a / b)
      wlhi = wlhia - 1
c **** Define interval, delta, for wavelength at which the calculations
c **** are to be carried out.  There will be approximately 150 points
c **** for calculations.
```

```
      delta = (wllo + wlhi) / 150
c **** The value of the variable 'v' is used to determine if it is the
c **** first run through the DO loop.  If so, the subroutine BISECTOR
c **** cannot be called since at least two runs through the loop are
c **** required.
      v = 0
c **** Begin DO loop
      do 80 wl = wllo, wlhi, delta
         v = v + 1
c ******* Convert the wavelength to a real number.
         wlr = wl * 1.0
         wl1 = wl2
         wl2 = wlr
c ******* Calculate the frequency in GHz.
         f = (30000.0 / 2.54) / wlr
c ******* Calculate variables alpha and beta
         pre_al = (a / (wlr ** 2)) - b
         pre_b = b - (c / (wlr ** 2))
         alpha = sqrt(pre_al)
         beta = sqrt(pre_b)
c ******* Call the subroutine J1JZ to calculate the ratio of the
c ******* Bessel functions J0 / J1 for the above calculated alpha.
         call J1JZ(alpha, ratjzj1)
c ******* Call the subroutine K1KZ to calculate the ratio of the
c ******* Bessel functions K0 / K1 for the above calculated beta.
         call K1KZ(beta, ratkzk1)
c ******* Calculate the function, variable func.
         func = ((alpha * ratjzj1) / er ) + (beta * ratkzk1)
c ******* Output the results.
         write(6,50) wl
 50      format(' WAVELENGTH = ', i4, ' MILS')
         write(20,50) wl
         write(6,60) f
 60      format(' FREQUENCY = ', f9.5, 'GHz')
         write(20,60) f
         write(6,70) func
 70      format(' FUNCTION = ', f7.2)
         write(20,70) func
         write(6,*) '                              '
         if (v .eq. 1)  go to 77
c ******* If v is greater than one, then save the last two values
c ******* of the function to calculate if a zero crossing is
c ******* between these two values.  If so, then call subroutine
c ******* BISECTOR to calculate the point of zero crossing.
c ******* Also save the corresponding wavelengths.  Note that wl_low
c ******* corresponds to func1 and wl_hi to func2.
         func1 = func2
         func2 = func
         if (func2 .gt. 0.0) go to 72
         if (func1 .gt. 0.0) go to 75
         go to 77
 72      if (func1 .lt. 0.0) go to 75
         go to 77
 75      wl_low = wl1
         wl_hi = wl2
         call BISECTOR(wl_low, wl_hi, func1, func2, t, d, er, mode_p,
     &                 f)
 77      write(20,*) '                              '
```

```
 80       continue
c ******* End do loop
       write(6,*) ' RESULTS ARE STORED IN FILE TM.DAT'
c **** Close data file 'TM.DAT'
       close(20)
       stop
       end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

       Subroutine J1JZ(arg, ratjzj1)
c ***********************************************************************
c **** This subroutine takes in the input, arg, which in this case   ***
c **** is the value of the variable alpha which is calculated in the ***
c **** main program.  The output of the subroutine is the value of   ***
c **** the ratio of the Bessel functions J0 to J1.                   ***
c ***********************************************************************
c **** Declare variables.
       Real             t(10), rjz, rj1, ratjzj1, sq_arg
       Integer          n
c **** Start calculations
       if (arg .gt. 3.0) goto 200
       t(1) = (arg / 3.0) ** 2
c **** Begin do loop
       do 190 n = 1, 5
 190      t(n+1) = t(n) * t(1)
c **** End do loop
       rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &       t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &       t(6)
       rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &       * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &       0.00001109 * t(6)
       rj1 = rj1 * arg
       go to 220
 200   continue
       t(1) = 3.0 / arg
c **** Begin do loop
       do 210 n = 1, 5
 210      t(n+1) = t(n) * t(1)
c **** End do loop
       fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &      0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &      + 0.00014476 * t(6)
       tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &      0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &      + 0.00013558 * t(6)
       sq_arg = fz / sqrt(arg)
       rjz = sq_arg * cos(tz)
       f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &      0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &      - 0.00020033 * t(6)
       t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &      - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
     &      - 0.00029166 * t(6)
       sq_arg = f1 / sqrt(arg)
       rj1 = sq_arg * cos(t1)
```

```
 220   continue
       ratjzj1 = rjz / rj1
       return
       end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

       Subroutine K1KZ(arg, ratkzk1)
c ***************************************************************************
c *** This subroutine takes in the input, arg, which in this case is ***
c *** the value of the variable beta which is calculated in the main ***
c *** program.  The output of the subroutine is the value of the     ***
c *** ratio of the Bessel functions K0 to K1.                        ***
c ***************************************************************************
c **** Declare variables
       Integer          n
       Real             t(10), rk1, rkz, ratkzk1
c **** Start calculations
       if (arg .gt. 2.0) go to 250
       t(1) = (arg / 3.75) ** 2
c **** Begin do loop
       do 230 n = 1, 5
 230      t(n+1) = t(n) * t(1)
c **** End do loop
       riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &       t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &       * t(6)
       ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &       t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &       * t(6)
       ri1 = ri1 * arg
       t(1) = (arg / 2.0) ** 2
c **** Begin do loop
       do 240 n = 1, 5
 240      t(n+1) = t(n) * t(1)
c **** End do loop
       argh = arg / 2.0
       rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &       0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &       + 0.00010750 * t(5) + 0.00000740 * t(6)
       rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &       0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &       -  0.00110404 * t(5) - 0.00004686 * t(6)
       rk1 = (rk1 / arg)
       go to 270
 250   continue
       t(1) = 2.0 / arg
c **** Begin do loop
       do 260 n = 1, 5
 260      t(n+1) = t(n) * t(1)
c **** End do loop
       rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
     &       - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &       t(5) + 0.00053208 * t(6)
       rkz = rkz / (((arg) ** 0.5) * exp(arg))
       rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &       + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
```

```
     &       t(5) - 0.00068245 * t(6)
        rk1 = rk1 / (((arg) ** 0.5) * exp(arg))
 270    continue
        ratkzk1 = rkz / rk1
        return
        end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

        Subroutine BISECTOR(xa, xb, funca, funcb, t, d, er, mode_p, f)
c ***********************************************************************
c ***  This subroutine calculates the value of the zero crossing      ***
c ***  between the two points provided by the main program when it    ***
c ***  calls this subroutine.                                         ***
c ***********************************************************************
c **** Declare variables.
        Integer          iterations
        Real             xa, xb, xc, funca, funcb, funcc
        Real             root, t, d, er, f, mode_p
c **** Calculate xc, the midpoint between xa and xb, which will
c **** become one of the new limits.
        iterations = 0
 280    xc = (xa + xb) / 2.0
        iterations = iterations + 1
c **** The difference in the sign of xa and xb can be when the function
c **** goes to plus or minus infinity.  Hence the zero cannot be found.
c **** If the loop is repeated more than 100 times, it is very likely
c **** that the case is not that of zero crossing.  Therefore, get out
c **** of the loop.
        if (iterations .gt. 100) go to 335
c **** Calculate the value of the function for xc, i.e. funcc.
        call FUNC(xc, t, d, er, mode_p, funcc)
c **** If the value of the function at both xa and xc are either
c **** positive or negative, then let the new value of xa be equal to
c **** xc.  Now calculate the new value of funca (the value of the
c **** variable func at the point xa).  If on the other hand, the
c **** value of the function at both xb and xc are of the same sign,
c **** then let the new value of xb be equal to xc and now calculate
c **** the new value of funcb(the value of the variable func at point
c **** xb).  Repeat this till the value of function is approximately
c **** equal to zero.  Then let the root be equal to xc and output the
c **** results.
        if (funcc .lt. 0.0000) go to 300
        if (funca .lt. 0.0000) go to 310
 290    xa = xc
        call FUNC(xa, t, d, er, mode_p, funca)
        if (abs(funca) .lt. 0.0001) go to 320
        go to 280
 300    if (funca .lt. 0.0000) go to 290
 310    xb = xc
        call FUNC(xb, t, d, er, mode_p, funcb)
        if (abs(funcb) .lt. 0.0001) go to 320
        go to 280
 320    root = xc
c **** Output the results.
        write(6,*)'                                                    '
        write(6,*)'WAVELENGTH AT WHICH THE FUNCTION IS EQUAL TO '
```

```
      write(6,*)'ZERO = ', root, ' MILS CORRESPONDING TO '
      write(6,*)'FREQUENCY = ', f, ' GHz'
      write(6,*)'                          '
      write(20,*)'                         '
      write(20,*)'WAVELENGTH AT WHICH THE FUNCTION IS EQUAL TO '
      write(20,*)'ZERO = ', root, ' MILS CORRESPONDING TO '
      write(20,*)'FREQUENCY = ', f, 'GHz'
      write(20,*)'                         '
 335  return
      end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

      Subroutine FUNC(wlr, t, d, er, mode_p, function)
c ***********************************************************************
c *** This subroutine calculates the value of the variable function  ***
c *** for the particular wavelength that is provided as an argument  ***
c *** by the subroutine BISECTOR.  It calls upon the subroutines     ***
c *** J1JZ and K1KZ to provide the values of the ratios of the       ***
c *** Bessel functions J0 / J1 and the ratio K0 / K1.                ***
c ***********************************************************************
c **** Declare variables.
      Real             wlr, t, d, er, pi, a, b, c
      Real             pre_al, pre_b, alpha, beta
      Real             ratj1jz, ratk1kz, mode_p
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4.0
      c = (pi ** 2) * (d ** 2)
c **** Calculate variables alpha and beta
      pre_al = (a / (wlr ** 2)) - b
      pre_b = b - (c / (wlr ** 2))
      alpha = sqrt(pre_al)
      beta = sqrt(pre_b)
c **** Call subroutine J1JZ to calculate the value of the Bessel
c **** functions ratio J0 /J1.
      call J1JZ(alpha, ratjzj1)
c **** Call subroutine K1KZ to calculate the value of the Bessel
c **** functions ratio K0 / K1.
      call K1KZ(beta, ratkzk1)
c **** Calculate the variable 'function'.
      function = ((alpha * ratjzj1) / er ) + (beta * ratkzk1)
c **** Return to subroutine BISECTOR the value of the variable
c **** function.
      return
      end

c ***********************************************************************
c                              END OF PROGRAM
c ***********************************************************************
```

```
      Program HE
c ***************************************************************************
c ***                    File Name: HE.FOR                                ***
c *** This program finds the values of the function,fn, for a whole      ***
c *** range of wavelengths, provided the diameter, thickness of puck,    ***
c *** and the dielectric constant are provided (for HE1np modes). It     ***
c *** then calculated the frequencies at which the various modes are     ***
c *** resonating.  These are calculated by calculating where the         ***
c *** function, func, goes to zero.                                      ***
c ***************************************************************************
c **** Declare variables
      Integer          wl, wllo, wlhi, delta
      Real             wlr, wlloa, wlhia, func, f, mode_p, fn1, fn2
      Real             pre_a1, pre_b, alpha, beta, wl_low, wl_hi
      Real             lhs, rhs, wl1, wl2, func1, func2, v
      Real             t, d, er, pi, a, b, c, ratj1derj1, ratk1derk1
c **** Open data file, 'HE.DAT', that will store the data as it is
c **** produced so it can be accessed later on.
      open (unit=20, file='HE.DAT', status='unknown')
c **** Get input data
      write (6,10)
 10   format(' ENTER VALUE OF p FOR HE1np MODE (REAL #)')
      write (20,10)
      read(5,*) mode_p
      write(20,*) mode_p
      write(6,20)
 20   format(' ENTER DIAMETER OF SAMPLE IN MILS (REAL #)')
      write(20,20)
      read(5,*) d
      write(20,*) d
      write(6,30)
 30   format(' ENTER HEIGHT OF SAMPLE IN MILS (REAL #)')
      write(20,30)
      read(5,*) t
      write(20,*) t
      write(6,40)
 40   format(' ENTER RELATIVE DIELECTRIC CONSTANT (REAL #)')
      write(20,40)
      read(5,*) er
      write(20,*) er
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4
      c = (pi ** 2) * (d ** 2)
c **** Define limits, wllo and wlhi, for the range of wavelength to be
c **** analyzed.
      wlloa = sqrt(c / b)
      wllo = wlloa + 1
      wlhia = sqrt(a / b)
      wlhi = wlhia - 1
c **** Define interval, delta, for wavelength at which the calculations
c **** are to be carried out. There will be approximately 250 points
c **** for calculations.
      delta = (wllo + wlhi) / 250
```

```
c **** The value of the variable 'v' is used to determine if it is the
c **** first run through the DO loop.  If so, the subroutine BISECTOR
c **** cannot be called since at least two runs through the loop are
c **** required.
      v = 0
c **** Begin DO loop
      do 80 wl = wllo, wlhi, delta
         v = v + 1
c ******* Convert the wavelength to a real number
         wlr = wl * 1.0
         wl1 = wl2
         wl2 = wlr
c ******* Calculate the frequency in GHz
         f = (30000.0 / 2.54) / wlr
         pre_al = (a / (wlr ** 2)) - b
         pre_b = b - (c / (wlr ** 2))
         alpha = sqrt(pre_al)
         beta = sqrt(pre_b)
c ******* Call the subroutine J1DERJ1 to calculate the ratio of the
c ******* Bessel functions derivative(J1) to J1 for the above calculated
c ******* alpha.
         call J1DERJ1(alpha, ratj1derj1)
c ******* Call the subroutine K1DERK1 to calculate the ratio of the
c ******* Bessel functions derivative(K1) to K1 for the above calculated
c ******* beta.
         call K1DERK1(beta, ratk1derk1)
c ******* Calculate the function, func
         fn1 = (ratj1derj1 / alpha) + (ratk1derk1 / beta)
         fn2 = ((ratj1derj1 * er) / alpha) + (ratk1derk1 / beta)
         lhs = fn1 * fn2
         rhs = ((er / (alpha ** 2)) + (1 / (beta ** 2))) *
     &         ((1 / (alpha ** 2)) + (1 / (beta ** 2)))
         func = lhs - rhs
c ******* Output the results.
         write(6,50) wl
 50      format(' WAVELENGTH = ', i4, ' MILS')
         write(20,50) wl
         write(6,60) f
 60      format(' FREQUENCY = ', f9.5, ' GHz')
         write(20,60) f
         write(6,70) func
 70      format(' FUNCTION = ',f7.2)
         write(20,70) func
         write(6,*) '
         if (v .eq. 1)  go to 77
c ******* If v is greater than one, then save the last two values
c ******* of the function to calculate if a zero crossing is
c ******* between these two values.  If so, then call subroutine
c ******* BISECTOR to calculate the point of zero crossing.
c ******* Also save the corresponding wavelengths.  Note that 'wl_low'
c ******* corresponds to 'func1' and 'wl_hi' to 'func2'.
         func1 = func2
         func2 = func
         if (func2 .gt. 0.0) go to 72
         if (func1 .gt. 0.0) go to 75
         go to 77
 72      if (func1 .lt. 0.0) go to 75
         go to 77
```

```
 75        wl_low = wl1
           wl_hi = wl2
           call BISECTOR(wl_low,wl_hi,func1,func2,t,d,er,mode_p,f)
 77        write(20,*) '                                '
 80        continue
c ****     End do loop
        write(6,*) ' RESULTS ARE STORED IN FILE HE.DAT'
        write(20,*) ' RESULTS ARE STORED IN FILE HE.DAT'
c **** Close data file 'HE.DAT'
        close(20)
        stop
        end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---
c
        Subroutine J1DERJ1(arg, ratj1derj1)
c ***********************************************************************
c *** This subroutine takes in the input, arg, which in this case is ***
c *** the value of the variable alpha which is calculated in the main***
c *** program.  The output of the subroutine is the value of the     ***
c *** ratio of the first derivative of J1 to the Bessel function J1. ***
c ***********************************************************************
c **** Declare variables.
        Real              t(10), rjz, rj1, ratj1derj1, rj1der, sq_arg
        Integer           n
c **** Start calculations
        if (arg .gt. 3.0) goto 200
        t(1) = (arg / 3.0) ** 2
c **** Begin do loop
        do 190 n = 1, 5
 190       t(n+1) = t(n) * t(1)
c **** End do loop
        rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &        t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &        t(6)
        rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &        * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &        0.00001109 * t(6)
        rj1 = rj1 * arg
        go to 220
 200    continue
        t(1) = 3.0 / arg
c **** Begin do loop
        do 210 n = 1, 5
 210       t(n+1) = t(n) * t(1)
c **** End do loop
        fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &       0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &       + 0.00014476 * t(6)
        tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &       0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &       + 0.00013558 * t(6)
        sq_arg = fz / sqrt(arg)
        rjz = sq_arg * cos(tz)
        f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &       0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &       - 0.00020033 * t(6)
```

```
      t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &      - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
     &      - 0.00029166 * t(6)
        sq_arg = f1 / sqrt(arg)
        rj1 = sq_arg * cos(t1)
 220    continue
        rj1der = rjz - (rj1 / arg)
        ratj1derj1 = rj1der / rj1
        return
        end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

        Subroutine K1DERK1(arg, ratk1derk1)
c ***********************************************************************
c *** This subroutine takes in the input, arg, which in this case is ***
c *** the value of the variable beta which is calculated in the main ***
c *** program.  The output of the subroutine is the value of the     ***
c *** ratio of the first derivative of K1 to the Bessel function K1. ***
c ***********************************************************************
c **** Declare variables
        Integer          n
        Real             t(10), rk1, rkz, ratk1derk1, rk1der
c **** Start calculations
        if (arg .gt. 2.0) go to 250
        t(1) = (arg / 3.75) ** 2
c **** Begin do loop
        do 230 n = 1, 5
 230       t(n+1) = t(n) * t(1)
c **** End do loop
        riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &       t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &       * t(6)
        ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &       t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &       * t(6)
        ri1 = ri1 * arg
        t(1) = (arg / 2.0) ** 2
c **** Begin do loop
        do 240 n = 1, 5
 240       t(n+1) = t(n) * t(1)
c **** End do loop
        argh = arg / 2.0
        rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &       0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &       + 0.00010750 * t(5) + 0.00000740 * t(6)
        rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &       0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &       -  0.00110404 * t(5) - 0.00004686 * t(6)
        rk1 = (rk1 / arg)
        go to 270
 250    continue
        t(1) = 2.0 / arg
c **** Begin do loop
        do 260 n = 1, 5
 260       t(n+1) = t(n) * t(1)
c **** End do loop
```

```
        rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
     &       - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &       t(5) + 0.00053208 * t(6)
        rkz = rkz / (sqrt(arg) * exp(arg))
        rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &       + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
     &       t(5) - 0.00068245 * t(6)
        rk1 = rk1 / (sqrt(arg) * exp(arg))
 270    continue
        rk1der = -rkz - (rk1 / arg)
        ratk1derk1 = rk1der / rk1
        return
        end
c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

        Subroutine BISECTOR (xa,xb,funca,funcb,t,d,er,mode_p,f)
c ***********************************************************************
c **** This subroutine calculates the value of the zero crossing      ***
c **** between the two points provided by the main program when it    ***
c **** calls this subroutine.                                         ***
c ***********************************************************************
c **** Declare variables
        Integer          iterations
        Real             ratj1derj1, ratk1derk1, mode_p
        Real             xa, xb, xc, funca, funcb, funcc
c **** Start calculations
c **** Call subroutine 'func' to calculate the value of the function
c **** at the two limits of the wavelength that were provided.
c **** Note that variable 'funca' is the value of the function at xa
c **** similarly 'funcb' is the value of the function at xb.
        iterations = 0
        call func(xa, t, d, er, mode_p, funca, f)
        call func(xb, t, d, er, mode_p, funcb, f)
c **** Calculate xc, the midpoint between xa and xb, which will
c **** become one of the new limits.
 280    xc = (xa + xb) / 2.0
        iterations = iterations + 1
        if (iterations .gt. 20) go to 335
c **** Calculate the value of the function for xc, i.e. funcc.
        call func(xc, t, d, er, mode_p, funcc, f)
c       write(6,*)'xa', xa, xb, xc, funca, funcb, funcc
c **** If the value of the function at both xa and xc are either
c **** positive or negative, then let the new value of xa be equal to
c **** xc.  Now calculate the new value of funca.  If on the other
c **** hand, the value of the function at both xb and xc are of the
c **** same sign, then let the new value of xb be equal to xc and now
c **** calculate the new value of funcb.  Repeat this till the value of
c **** function is approximately equal to zero.  Then let the root be
c **** equal to xc and output the results.
        if (funcc .lt. 0.0000) go to 300
        if (funca .lt. 0.0000) go to 310
 290    xa = xc
        call func(xa, t, d, er, mode_p, funca, f)
        if (abs(funca) .lt. 0.0001) go to 320
        go to 280
 300    if (funca .lt. 0.0000) go to 290
 310    xb = xc
```

```
      call func(xb, t, d, er, mode_p, funcb, f)
      if (abs(funcb) .lt. 0.0001) go to 320
      go to 280
 320  root = xc
c **** Output the results.
      write(6,*)' WAVELENGTH AT WHICH THE FUNCTION IS EQUAL TO '
      write(6,*)' ZERO = ', root, ' MILS', ' CORRESPONDING TO '
      write(6,*)' FREQUENCY = ', f, ' GHz'
      write(6,*)'                                                '
      write(20,*)'                                                '
      write(20,*)'WAVELENGTH AT WHICH THE FUNCTION IS EQUAL TO '
      write(20,*)'ZERO = ', root, ' MILS', ' CORRESPONDING TO '
      write(20,*)'FREQUENCY = ',f , ' GHz'
      write(20,*)'                                                '
 335  return
      end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

      Subroutine FUNC(wlr, t, d, er, mode_p, function, freq)
c ***************************************************************************
c **** This subroutine calculates the value of the variable function ***
c **** for the particular wavelength that is provided as an argument ***
c **** by the subroutine BISECTOR.  It calls upon the subroutines  ***
c **** J1DERJ1 and K1DERK1 to provide the values of the ratios of    ***
c **** the Bessel functions J1/derivative(J1) and the ratio          ***
c **** K1/derivative(K1).
c ***************************************************************************
c **** Declare variables.
      Real             wlr, t, d, er, pi, a, b, c
      Real             pre_al, pre_b, alpha, beta
      Real             fn1, fn2, lhs, rhs, ratj1derj1
      Real             ratk1derk1, freq, mode_p
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4.0
      c = (pi ** 2) * (d ** 2)
c **** Calculate the frequency in GHz.
      freq = (30000.0 / 2.54) / wlr
c **** Calculate variables alpha and beta
      pre_al = (a / (wlr ** 2)) - b
      pre_b = b - (c / (wlr ** 2))
      alpha = sqrt(pre_al)
      beta = sqrt(pre_b)
c **** Call the subroutine J1DERJ1 to calculate the ratio of the
c **** Bessel functions J1 / derivative(J1) for the above calculated
c **** alpha.
      call J1DERJ1(alpha, ratj1derj1)
c **** Call the subroutine K1DERK1 to calculate the ratio of the
c **** Bessel functions K1 / derivative(K1) for the above calculated
c **** beta.
      call K1DERK1(beta, ratk1derk1)
      fn1 = (ratj1derj1 / alpha) + (ratk1derk1 / beta)
      fn2 = ((ratj1derj1 * er) / alpha) + (ratk1derk1 / beta)
      lhs = fn1 * fn2
      rhs = ((er / (alpha ** 2)) + (1 / (beta ** 2))) *
```

```
     &       ((1 / (alpha ** 2)) + (1 / (beta ** 2)))
c **** Calculate the variable 'function'.
       function = lhs - rhs
c **** Return to subroutine BISECTOR the value of the variable
c **** 'function'.
       return
       end

c ***************************************************************************
c                              END OF PROGRAM
c ***************************************************************************
```

3. Program 'Q0.FOR'

This program is first used to calculate the points at which the 3dB frequencies should be measured, given the value of the coupling, β (see Appendix B)., When these 3dB frequencies are input to the program, it then calculates the Q_0 and the f_0 value.

```
       Program UnloadedQ
c ***************************************************************************
c ****                    File Name: Q0.FOR                                ***
c **** This program calculates Q0 from values read off the amplitude      ***
c **** and polar display. It is based on the equations given in the       ***
c **** article "Swept-frequency microwave Q-factor measurement" by       ***
c **** J.E. Aitken.                                                      ***
c ***************************************************************************
c **** Declare variables
       Character          B
       Real               Dm, beta, pre_D0, D0, Q0
       Real               f0, flo, fhi
c **** Get input data
c **** By looking at the polar display resonance, it can be seen
c **** whether beta is less than or greater than one.
       write(6,*) ' LOOK AT THE POLAR DISPLAY RESONANCE'
       write(6,*) ' ENTER "G" IF BETA > 1, "L" IF BETA < 1'
       read(5,10) B
 10    format (A1)
       write(6,*) ' LOOK AT THE RESONANCE ON THE AMPLITUDE '
       write(6,*) ' DISPLAY.  READ OFF HOW MANY dBs THE  '
       write(6,*) ' BOTTOM OF THE RESONANCE CURVE IS BELOW '
       write(6,*) ' THE REFERENCE LEVEL.'
       write(6,*) '                                                      '
       write(6,*) ' ENTER Dm (MAX. REFL. COEFF. AS A POSITIVE NUMBER)'
       read(5,*) Dm
       if (B .eq. 'g') go to 100
       if (B .eq. 'G') go to 100
       beta = (1 - 10 ** (-Dm / 20)) / (1 + 10 ** (-Dm / 20))
```

```
      go to 120
 100  beta = (1 + 10 ** (-Dm / 20)) / (1 - 10 ** (-Dm / 20))
 120  write(6,*) ' BETA =', beta
c **** Calculate the value of D0 which is 3 dB below the reference
c **** level.
      pre_D0 = ((2 - (2 * beta) + (beta ** 2)) /
     &     (2 + (2 * beta) + (beta ** 2))) ** 0.5
      D0 = 20.0 * alog10(pre_D0)
c **** Output the value of D0.
      write(6,*) ' READ FREQ. AT ',D0,' dB BELOW THE REFERENCE POINT'
c **** Input the three dB points.
      write(6,*) ' ENTER THE 3 dB LOW FREQUENCY VALUE, flo'
      read(5,*) flo
      write(6,*) ' ENTER THE 3 dB HIGH FREQUENCY VALUE, fhi'
      read(5,*) fhi
c **** Calculate the center frequency
      f0 = (flo + fhi) / 2
c **** Calculate the unloaded Q
      Q0 = f0 / (fhi - flo)
c **** Output results.
      write(6,*) ' Q0 = ', Q0
      write(6,*) ' CENTER FREQUENCY = ', f0
      stop
      end
c ***************************************************************************
c                          END OF PROGRAM
c ***************************************************************************
```

4. Program 'ER.FOR'

This program calculates the value of the relative dielectric constant for the TE_{011} resonance when it is provided with the diameter and height of the sample, the value of p (the third subscript of the mode) and the frequency of resonance of the TE_{011} mode.

```
      Program ER
c ***************************************************************************
c ****                    File Name: ER.FOR                                ****
c **** This program finds the relative dielectric constant for a        ****
c **** given frequency of resonance for TE0np mode, provided the        ****
c **** diameter, the height of the sample and the value of p are        ****
c **** also given.                                                      ****
c ***************************************************************************
c **** Declare variables
      Integer          erlow_i, erhigh
      Real             mode_p, wlr, func, erlow
      Real             pre_b, beta, pre_al, alpha, t, d, er, pi
```

```
      Real              a, b, c, ratj1jz, ratk1kz, f, v
c **** Open data file 'ER.DAT' that will store the data as it is
c **** produced so it can be accessed later.
      open (unit=20, file='ER.DAT', status='unknown')
c **** Get input data
      write(6,10)
 10   format(' ENTER DIAMETER OF SAMPLE IN MILS (REAL #)')
      write(20,10)
      read(5,*) d
      write(20,*) d
      write(6,20)
 20   format(' ENTER HEIGHT OF SAMPLE IN MILS (REAL #)')
      write(20,20)
      read(5,*) t
      write(20,*) t
      write(6,30)
 30   format(' ENTER VALUE OF p (REAL #) FOR TE0np MODE')
      write(20,30)
      read(5,*) mode_p
      write(20,*) mode_p
      write(6,40)
 40   format(' ENTER FREQUENCY OF RESONANCE FOR TE0np MODE (GHz)')
      write(20,40)
      read(5,*) f
      write(20,*) f
c **** Calculate the wavelength in mils
      wlr = (30000.0 / f) / 2.54
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      b = ((pi * d * mode_p / t) ** 2) / 4
      c = (pi ** 2) * (d ** 2)
c **** Calculate variables 'alpha' and 'beta'
      pre_b = b - (c / (wlr ** 2))
      beta = sqrt(pre_b)
c **** Call the subroutine K1KZ to calculate the ratio of the Bessel
c **** functions K0 / K1 for the above calculated beta.
      call K1KZ(beta, ratkzk1)
c **** Calculate the lower limit of the range of the relative
c **** dielectric constant.  This limit is calculated so that the square
c **** of alpha does not become negative and hence alpha is not
c **** imaginary.
      erlow = ((wlr * mode_p) / (2 * t)) ** 2
      erlow_i = erlow
      erlow_i = erlow_i + 1
c **** The upper limit of the range of the relative dielectric constant
c **** is arbitrarily set to 100.
      erhigh = 100
c **** Calculate the interval, delta, for the range of the dielectric
c **** constant at which the calculations will be done.  In this case
c **** there will be approximately 50 points of data.
      delta = (erhigh - erlow_i) / 50
c **** The value of the variable 'v' is used to determine if it is the
c **** first run through the DO loop.  If so, the subroutine BISECTOR
c **** cannot be called since at least two runs through the loop are
c **** required.
      v = 0
c **** Begin do loop
      Do 80 er = erlow_i, erhigh, delta
```

```
            v = v + 1
            er1 = er2
            er2 = er
            a = ((pi * d) ** 2) * er
c ******** Calculate variable alpha
            pre_al = (a / (wlr ** 2)) - b
            alpha = sqrt(pre_al)
c ******** Call the subroutine J1JZ to calculate the ratio of the
c ******** Bessel functions J0 / J1 for the above calculated alpha.
            call J1JZ(alpha, ratjzj1)
c ******** Calculate the function, variable func
            func = (alpha * ratjzj1) + (beta * ratkzk1)
c ******** Output the results
            write(6,50) er
 50         format(' IF RELATIVE DIELECTRIC CONSTANT =', f7.2)
            write(20,50) er
            write(6,60) func
 60         format(' THEN VALUE OF FUNCTION IS =', f7.2)
            write(20,60) func
            write(6,*) '                                    '
            if (v .eq. 1)  go to 77
c ******** If v is greater than one, then save the last two values
c ******** of the function to calculate if a zero crossing is
c ******** between these two values.  If so, then call subroutine
c ******** BISECTOR to calculate the point of zero crossing.
c ******** Also save the corresponding dielectric constant values.
c ******** Note that 'er_low' corresponds to 'func1' and 'er_hi' to
c ******** 'func2'.
            func1 = func2
            func2 = func
            if (func2 .gt. 0.0) go to 72
            if (func1 .gt. 0.0) go to 75
            go to 77
 72         if (func1 .lt. 0.0) go to 75
            go to 77
 75         er_low = er1
            er_hi = er2
            call BISECTOR(er_low, er_hi, func1, func2, t, d, mode_p, f)
 77         write(20,*) '                                    '
 80         continue
c ******** End do loop
       write(6,*) 'RESULTS ARE STORED IN FILE ER.DAT'
c **** Close data file 'ER.DAT'
       close(20)
       stop
       end

c *-------**-------**------**------**------**------**------**------**---
c *-------**-------**------**------**------**------**------**------**---

       Subroutine J1JZ(arg, ratjzj1)
c ***********************************************************************
c **** This subroutine takes in the input, arg, which in this case   ***
c **** is the value of the variable alpha which is calculated in the ***
c **** main program.  The output of the subroutine is the value of   ***
c **** the ratio of the Bessel functions J0 to J1.                   ***
c ***********************************************************************
```

```
c **** Declare variables.
      Real            t(10), rjz, rj1, ratjzj1, sq_arg
      Integer         n
c **** Start calculations
      if (arg .gt. 3.0) goto 200
      t(1) = (arg / 3.0) ** 2
c **** Begin do loop
      do 190 n = 1, 5
 190     t(n+1) = t(n) * t(1)
c **** End do loop
      rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &       t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &       t(6)
        rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &       * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &       0.00001109 * t(6)
        rj1 = rj1 * arg
        go to 220
 200    continue
        t(1) = 3.0 / arg
c **** Begin do loop
        do 210 n = 1, 5
 210       t(n+1) = t(n) * t(1)
c **** End do loop
        fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &       0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &       + 0.00014476 * t(6)
        tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &       0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &       + 0.00013558 * t(6)
        sq_arg = fz / sqrt(arg)
        rjz = sq_arg * cos(tz)
        f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &       0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &       - 0.00020033 * t(6)
        t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &       - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
     &       - 0.00029166 * t(6)
        sq_arg = f1 / sqrt(arg)
        rj1 = sq_arg * cos(t1)
 220    continue
        ratjzj1 = rjz / rj1
        return
        end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

        Subroutine K1KZ(arg, ratkzk1)
c ***********************************************************************
c *** This subroutine takes in the input, arg, which in this case is ***
c *** the value of the variable beta which is calculated in the main ***
c *** program.  The output of the subroutine is the value of the     ***
c *** ratio of the Bessel functions K0 to K1.                        ***
c ***********************************************************************
c **** Declare variables
        Integer         n
        Real            t(10), rk1, rkz, ratkzk1
```

```
c **** Start calculations
      if (arg .gt. 2.0) go to 250
      t(1) = (arg / 3.75) ** 2
c **** Begin do loop
      do 230 n = 1, 5
 230     t(n+1) = t(n) * t(1)
c **** End do loop
      riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &      t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &      * t(6)
      ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &      t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &      * t(6)
      ri1 = ri1 * arg
      t(1) = (arg / 2.0) ** 2
c **** Begin do loop
      do 240 n = 1, 5
 240     t(n+1) = t(n) * t(1)
c **** End do loop
      argh = arg / 2.0
      rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &      0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &      + 0.00010750 * t(5) + 0.00000740 * t(6)
      rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &      0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &      - 0.00110404 * t(5) - 0.00004686 * t(6)
      rk1 = (rk1 / arg)
      go to 270
 250  continue
      t(1) = 2.0 / arg
c **** Begin do loop
      do 260 n = 1, 5
 260     t(n+1) = t(n) * t(1)
c **** End do loop
      rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
     &      - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &      t(5) + 0.00053208 * t(6)
      rkz = rkz / (((arg) ** 0.5) * exp(arg))
      rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &      + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
     &      t(5) - 0.00068245 * t(6)
      rk1 = rk1 / (((arg) ** 0.5) * exp(arg))
 270  continue
      ratkzk1 = rkz / rk1
      return
      end

c *-------**-------**-------**-------**-------**-------**-------**---
c *-------**-------**-------**-------**-------**-------**-------**---

      Subroutine BISECTOR(xa, xb, funca, funcb, t, d, mode_p, f)
c ***********************************************************************
c **** This subroutine calculates the value of the zero crossing     ***
c **** between the two points provided by the main program when it   ***
c **** calls this subroutine.                                        ***
c ***********************************************************************
```

```
c **** Declare variables.
      Integer          iterations
      Real             xa, xb, xc, funca, funcb, funcc
      Real             root, t, d, f, mode_p
c **** Calculate xc, the midpoint between xa and xb, which will
c **** become one of the new limits.
      iterations = 0
 280  xc = (xa + xb) / 2.0
      iterations = iterations + 1
c **** The difference in the sign of xa and xb can be when the function
c **** goes to plus or minus infinity.  Hence the zero cannot be found.
c **** If the loop is repeated more than 100 times, it is very likely
c **** that the case is not that of zero crossing.  Therefore, get out
c **** of the loop.
      if (iterations .gt. 100) go to 335
c **** Calculate the value of the function for xc, i.e. funcc.
      call FUNC(xc, t, d, mode_p, funcc, f)
c **** If the value of the function at both xa and xc are either
c **** positive or negative, then let the new value of xa be equal to
c **** xc.  Now calculate the new value of funca (the value of the
c **** variable func at the point xa).  If on the other hand, the
c **** value of the function at both xb and xc are of the same sign,
c **** then let the new value of xb be equal to xc and now calculate
c **** the new value of funcb(the value of the variable func at point
c **** xb).  Repeat this till the value of function is approximately
c **** equal to zero.  Then let the root be equal to xc and output the
c **** results.
      if (funcc .lt. 0.0000) go to 300
      if (funca .lt. 0.0000) go to 310
 290  xa = xc
      call FUNC(xa, t, d, mode_p, funca, f)
      if (abs(funca) .lt. 0.0001) go to 320
      go to 280
 300  if (funca .lt. 0.0000) go to 290
 310  xb = xc
      call FUNC(xb, t, d, mode_p, funcb, f)
      if (abs(funcb) .lt. 0.0001) go to 320
      go to 280
 320  root = xc
c **** Output the results.
      write (6,*) 'THE RELATIVE DIELECTRIC CONSTANT AT WHICH THE'
      write (6,*) 'FUNCTION IS APPROXIMATELY EQUAL TO ZERO IS', root
      write (6,*) '                                           '
      write (20,*) 'THE RELATIVE DIELECTRIC CONSTANT AT WHICH THE'
      write (20,*) 'FUNCTION IS APPROXIMATELY EQUAL TO ZERO IS', root
      write (20,*) '                                           '
 335  return
      end

c*********************************************************************
      Subroutine FUNC(er, t, d, mode_p, function, f)
c *********************************************************************
c **** This subroutine calculates the value of the variable         ****
c **** 'function' for the particular relative dielectric constant   ****
c **** that is provided as an argument by the subroutine bisector.  ****
c **** It calls upon the subroutines J1JZ and K1KZ to provide the   ****
c **** values of the ratios of the Bessel functions J0/J1 and K0/K1.****
c *********************************************************************
```

```
c **** Declare variables.
      Real            wlr, mode_p, f, pi, pre_al, pre_b
      Real            alpha, beta
      Real            t, d, er, a, b, c, ratj1jz, ratk1kz
c **** Define constant pi
      pi = 4.0 * atan(1.0)
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4
      c = (pi ** 2) * (d ** 2)
c **** Calculate the wavelength in mils.
      wlr = (30000.0 / 2.54) / f
      pre_al = (a / (wlr ** 2)) - b
      pre_b = b - (c / (wlr ** 2))
      alpha = sqrt(pre_al)
      beta = sqrt(pre_b)
c **** Call subroutine J1JZ to calculate the value of the Bessel
c **** functions ratio J0 /J1.
      call J1JZ(alpha, ratjzj1)
c **** Call subroutine K1KZ to calculate the value of the Bessel
c **** functions ratio K0 / K1.
      call K1KZ(beta, ratkzk1)
c **** Calculate the variable 'function'.
      function = (alpha * ratjzj1) + (beta * ratkzk1)
c **** Return to subroutine BISECTOR the value of the variable
c **** function.
      return
      end

c ***************************************************************************
c                          END OF PROGRAM
c ***************************************************************************
```

5. Program 'TANDLT.FOR'

This program calculates the loss tangent for the material for TE_{011} resonance. The input data required by the program is the dimensions of the sample, the value of p (the third subscript of the mode), the relative dielectric constant calculated by the program 'ER.FOR', the Q_0 and the f_0 value (calculated by 'Q0.FOR').. The loss tangent can be calculated at any temperature in the range of 77K to 300K. The conductivity equation takes into account the anomalous surface resistance effect. For the anomalous resistance equation, reference [23] was used.

```
      Program Tangent_delta
c ***************************************************************
c ****                  File Name: TANDLT.FOR                  ***
c **** This program is used to calculate the tandelta value for TE0np***
c **** mode for a particular freq of resonance, relative dielectric  ***
c **** constant, diameter and height of the sample.              ***
c ***************************************************************
c **** Declare variables
      Character              answer
      Real                   fn_alpha, fn_beta, AA, BB, x, y, z
      Real                   t, tm, d, er, pi, a, b, c
      Real                   rjz, rj1, rj2, u0, cond, e0, Bden
      Real                   Q0, temp, mode_n, mode_p
      Real                   sig_to_l, meanpath, l_to_delta
      Real                   resistivity, Rclassic, Ranam, alfa
      Real                   wlr, freq, rkz, rk1, rk2
      Real                   pre_al, pre_b, beta, alpha
      Double Precision       tandelta
c **** Get input data
      write(6,10)
 10   format(' ENTER DIAMETER OF PUCK IN MILS (REAL #)')
      read(5,*) d
      write(6,20)
 20   format (' ENTER HEIGHT OF PUCK IN MILS (REAL #)')
      read(5,*) t
c **** Convert height of puck into meters = tm
      tm = t * 2.54 / 1e5
 30   write(6,40)
 40   format(' ENTER VALUE OF p FOR TE0np MODE')
      read(5,*) mode_p
      write(6,50)
 50   format(' ENTER FREQUENCY OF RESONANCE OF TE0np MODE (GHz)')
      read(5,*) f
      wlr = (30000 / 2.54) / f
c **** Convert the frequency to Hz
      freq = f * 1.0e9
      write(6,60)
 60   format(' ENTER RELATIVE DIELECTRIC CONSTANT (REAL #)')
      read(5,*) er
      write(6,70)
 70   format(' ENTER UNLOADED Q (REAL #)')
      read(5,*) Q0
c **** Define constant pi
      pi = 4.0 * atan(1.0)
c **** Define permeability of free space = u0 (henry/m)
      u0 = 4.0 * pi * 1e-7
c **** Enter temperature of operation in kelvin.
      write(6,*) 'ENTER TEMPERATURE OF PUCK IN KELVIN'
      read(5,*) temp
c **** Start calculations
c **** Conductivity of gold for temperatures ranging from room
c **** temperature down to liquid nitrogen temperature is given by
c **** the following formula.  See Chapter IV, section 3 for details.
      cond = (0.877 * 4.562e7) / ( 1.0 + (0.00339 * (temp - 293.0)))
      write(6,*)'                                              '
      write(6,*)'CONDUCTIVITY OF GOLD AT',temp, 'K IS',cond, ' MHO/M'
      write(6,*)'                                              '
c **** Resistivity (units of ohm.m) is inverse of conductivity
```

```
      resistivity = 1.0 / cond
c **** Define sigma to l ratio (i.e. conductivity to mean free path
c **** ratio in units of mho / sq m) for gold, which is constant for
c **** a material.
      sig_to_l = 0.43e15
c **** Calculate the mean free path (unit of meters) for gold at
c **** this temperature.
      meanpath = 1.0 / (sig_to_l * resistivity)
c **** Calculate the skin depth, delta (unit of meters).
      delta = 1.0 / sqrt(pi * freq * u0 / resistivity)
c **** Calculate the classical surface resistance (unit of ohms).
      Rclassic = resistivity / delta
      write(6,*) 'CLASSICAL SURFACE RESISTANCE', Rclassic, ' OHMS'
c **** Calculate the mean free path to delta ratio (unitless).
      l_to_delta = meanpath / delta
c **** Calculate the value of alfa and z, as defined in Jim Matey's
c **** internal correspondence dated 4 July 1988 [23]. The function
c **** 'z' is defined as function 'a' in the correspondence.
      alfa = (l_to_delta ** 2) * 3.0 / 2.0
      z = Rclassic * alfa ** 0.1666
c **** Calculating the x and y coordinates of the graph in the
c **** correspondence by using the polynomial for y.
      x = alog(alfa)
      y = -0.038 + (0.131 * x) - (0.006 * x * x)
c **** Calculate the anomalous surface resistance of gold in ohms.
      y = -y
      Ranam = z * exp(y)
      write(6,*) 'ANOMALOUS SURFACE RESISTANCE = ', Ranam, ' OHMS'
c **** Define permittivity of free space = e0 (farads/m)
      e0 = 8.854 * 1e-12
      a = (pi ** 2) * (d ** 2) * er
      b = ((pi * d * mode_p / t) ** 2) / 4
      c = (pi ** 2) * (d ** 2)
c **** Calculate variables alpha and beta.
      pre_al = (a / (wlr ** 2)) - b
      pre_b = b - (c / (wlr ** 2))
      alpha = sqrt(pre_al)
      beta = sqrt(pre_b)
c **** Call subroutine J1JZ to calculate the Bessel functions J0 (i.e.
c **** variable rjz), J1 (i.e. variable rj1) and J2 (i.e. variable rj2)
c **** for the above calculated alpha.
      call J1JZ(alpha, rj2, rj1, rjz)
c **** Call subroutine K1KZ to calculate the Bessel functions K0 (i.e.
c **** variable rkz), K1 (i.e. variable rk1) and K2 (i.e. variable rk2)
c **** for the above calculated beta.
      call K1KZ(beta, rk2, rk1, rkz)
c **** Calculate the functions fn_alpha and fn_beta
      fn_alpha = (rj1 ** 2) / ((rj1 ** 2) - (rjz * rj2))
      fn_beta = ((rkz * rk2) - (rk1 ** 2)) / (rk1 ** 2)
      AA = 1 + (fn_alpha * fn_beta / er)
c **** Bden is the denominator of BB
      Bden = 2.0 * pi * (freq ** 3) * (u0 ** 2) * e0 * (tm ** 3) * er
c **** If the temperature of the sample is greater than 200K, then use
c **** the classical surface resistance, Rclassic.
      if (temp .lt. 200.0) go to 80
      write(6,*) '                                                '
      write(6,*) 'USING CLASSICAL SURFACE RESISTANCE VALUE'
      BB = Rclassic * (1.0 + (fn_alpha * fn_beta)) / Bden
```

```
      go to 90
c **** If the temperature of the sample is less than 200K, then use
c **** the anomalous surface resistance, Ranam.
 80   write(6,*) 'USING ANOMALOUS SURFACE RESISTANCE VALUE'
      BB = Ranam * (1.0 + (fn_alpha * fn_beta)) / Bden
c **** Calculate the loss tangent
 90   tandelta = (AA / Q0) - BB
      write(6,100) tandelta
 100  format('THE CALCULATED LOSS TANGENT = ',D16.9)
      write(6,*) '                                                '
      write(6,*) 'DO YOU WANT TO RUN IT AGAIN FOR SAME '
      write(6,*) 'SAMPLE DIMENSIONS ? ENTER Y(es) OR N(O)'
      read(5,110) answer
 110  format (A1)
      if (answer .eq. 'Y') go to 30
      if (answer .eq. 'y') go to 30
      stop
      end

c *-------**------**------**------**-------**-------**------**------**---
c *-------**------**------**------**-------**-------**------**------**---

      Subroutine J1JZ(arg, rj2, rj1, rjz)
c ***************************************************************************
c **** This subroutine takes in the input, arg, which in this case is
c **** the value of the variable alpha which is calculated in the main
c **** program.  The output of the subroutine is the value of the
c **** Bessel functions J0, J1 and J2.
c ***************************************************************************
c **** Declare variables
      Real             t(10), rjz, rj1, rj2, sq_arg
      Integer          n
      if (arg .gt. 3.0) go to 130
      t(1) = (arg / 3.0) ** 2
c **** Begin do loop
      do 120 n = 1, 5
 120     t(n+1) = t(n) * t(1)
c **** End do loop
      rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &      t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &      t(6)
      rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &      * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &      0.00001109 * t(6)
      rj1 = rj1 * arg
      go to 150
 130  continue
      t(1) = 3.0 / arg
c **** Begin do loop
      do 140 n = 1, 5
 140     t(n+1) = t(n) * t(1)
c **** End do loop
      fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &     0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &     + 0.00014476 * t(6)
      tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &     0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &     + 0.00013558 * t(6)
```

```
      sq_arg = fz / sqrt(arg)
      rjz = sq_arg * cos(tz)
      f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &     0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &     - 0.00020033 * t(6)
      t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &     - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
     &     - 0.00029166 * t(6)
      sq_arg = f1 / sqrt(arg)
      rj1 = sq_arg * cos(t1)
 150  continue
      rj2 = ((2.0 * rj1) / arg) - rjz
      return
      end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

      Subroutine K1KZ(arg, rk2, rk1, rkz)
c **********************************************************************
c **** This subroutine takes in the input, arg, which in this case is
c **** the value of the variable beta which is calculated in the main
c **** program.  The output of the subroutine is the value of the
c **** Bessel functions K0, K1 and K2.
c **********************************************************************
c **** Declare variables.
      Integer          n
      Real             t(10), rk2, rk1, rkz
      if (arg .gt. 2.0) go to 180
      t(1) = (arg / 3.75) ** 2
c **** Begin do loop
      do 160 n = 1, 5
 160     t(n+1) = t(n) * t(1)
c **** End do loop
      riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &      t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &      * t(6)
      ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &      t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &      * t(6)
      ri1 = ri1 * arg
      t(1) = (arg / 2.0) ** 2
c **** Begin do loop
      do 170 n = 1, 5
 170     t(n+1) = t(n) * t(1)
c **** End do loop
      argh = arg / 2.0
      rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &      0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &      + 0.00010750 * t(5) + 0.00000740 * t(6)
      rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &      0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &      -  0.00110404 * t(5) - 0.00004686 * t(6)
      rk1 = (rk1 / arg)
      go to 200
 180  continue
      t(1) = 2.0 / arg
c **** Begin do loop
```

```
      do 190 n = 1, 5
 190     t(n+1) = t(n) * t(1)
c **** End do loop
      rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
     &       - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &       t(5) + 0.00053208 * t(6)
      rkz = rkz / (((arg) ** 0.5) * exp(arg))
      rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &       + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
     &       t(5) - 0.00068245 * t(6)
      rk1 = rk1 / (((arg) ** 0.5) * exp(arg))
 200  continue
      rk2 = rkz + ((2.0 * rk1) / arg)
      return
      end
c ***************************************************************************
c                              END OF PROGRAM
c ***************************************************************************
```

6. Program 'COND.FOR'

This program is used to calculate the conductivity of the conducting plates for the TE011 resonance. The input data needed is the dimensions of the sample, the value of f_0 (calculated by 'Q0.FOR'), the ε_r value (calculated by 'ER.FOR') and the loss tangent value (calculated by 'TANDLT.FOR').

```
      Program Conductivity
c ***************************************************************************
c ***                    File Name: COND.FOR                                ***
c *** This program is used to calculate the conductivity value for         ***
c *** TE011 mode for a particular freq of resonance, relative              ***
c *** dielectric constant, diameter, length of puck and loss              ***
c *** tangent.                                                            ***
c ***************************************************************************
c ***  Declare variables
      Character              answer
      Real                   fn_alpha, fn_beta, AA, BB
      Real                   t, tm, d, er, pi, a, b, c
      Real                   rjz, rj1, rj2, u0, cond, e0, Bden
      Real                   Q0, mode_p
      Real                   resistivity, Rs
      Real                   wlr, freq, rkz, rk1, rk2
      Real                   pre_al, pre_b, beta, alpha, tandelta
```

```
c ***  Get input data
       write(6,10)
 10    format(' ENTER DIAMETER OF PUCK IN MILS (REAL #)')
       read(5,*) d
       write(6,20)
 20    format (' ENTER THICKNESS OF PUCK IN MILS (REAL #)')
       read(5,*) t
c ***  Convert length of puck into meters = tm
       tm = t * 2.54 / 1e5
 30    write(6,40)
 40    format(' ENTER VALUE OF p FOR TE0np MODE')
       read(5,*) mode_p
       write(6,50)
 50    format(' ENTER FREQUENCY OF RESONANCE OF TE0np MODE (GHz)')
       read(5,*) f
       wlr = (30000 / 2.54) / f
c ***  Convert the frequency to GHz
       freq = f * 1.0e9
       write(6,60)
 60    format(' ENTER RELATIVE DIELECTRIC CONSTANT (REAL #)')
       read(5,*) er
       write(6,70)
 70    format(' ENTER UNLOADED Q (REAL #)')
       read(5,*) Q0
       write(6,80)
 80    format(' ENTER TANDELTA (REAL #)')
       read(5,*) tandelta
c ***  Define constant pi
       pi = 4.0 * atan(1.0)
c ***  Define permeability of free space = u0 (henry/m)
       u0 = 4.0 * pi * 1e-7
c ***  Define permittivity of free space = e0 (farads/m)
       e0 = 8.854 * 1e-12
       a = (pi ** 2) * (d ** 2) * er
       b = ((pi * d * mode_p / t) ** 2) / 4
       c = (pi ** 2) * (d ** 2)
       pre_a1 = (a / (wlr ** 2)) - b
       pre_b = b - (c / (wlr ** 2))
       alpha = sqrt(pre_a1)
       beta = sqrt(pre_b)
c ***  Call subroutine J1JZ to calculate the Bessel functions J0 (i.e.
c ***  variable rjz), J1 (i.e. variable rj1) and J2 (i.e. variable rj2)
c ***  for the above calculated alpha.
       call J1JZ(alpha, rj2, rj1, rjz)
c ***  Call subroutine K1KZ to calculate the Bessel functions K0 (i.e.
c ***  variable rkz), K1 (i.e. variable rk1) and K2 (i.e. variable rk2)
c ***  for the above calculated beta.
       call K1KZ(beta, rk2, rk1, rkz)
c ***  Calculate the functions fn_alpha and fn_beta
       fn_alpha = (rj1 ** 2) / ((rj1 ** 2) - (rjz * rj2))
       fn_beta = ((rkz * rk2) - (rk1 ** 2)) / (rk1 ** 2)
       AA = 1 + (fn_alpha * fn_beta / er)
       BB = (AA / Q0) - tandelta
c ***  Bden is the denominator of BB
       Bden = 2.0 * pi * (freq ** 3) * (u0 ** 2) * e0 * (tm ** 3) * er
       Rs = BB * Bden / (1.0 + (fn_alpha * fn_beta))
       cond = pi * freq * u0 / (Rs ** 2)
       cond = cond / 1e7
```

```
      write(6,90) cond
 90   format(' CONDUCTIVITY OF PLATES = ', f9.2, ' e+7 mho/m ')
      write(6,*) '                                                '
      write(6,*) ' DO YOU WANT TO RUN IT AGAIN FOR SAME '
      write(6,*) ' SAMPLE DIMENSIONS ? ENTER Y(es) OR N(O)'
      read(5,100) answer
 100  format (A1)
      if (answer .eq. 'Y') go to 30
      if (answer .eq. 'y') go to 30
      stop
      end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

      Subroutine J1JZ(arg, rj2, rj1, rjz)
c ***************************************************************************
c ***  This subroutine takes in the input, arg, which in this case is***
c ***  the value of the variable alpha which is calculated in the    ***
c ***  main program.  The output of the subroutine is the value of   ***
c ***  the Bessel functions J0, J1 and J2.                           ***
c ***************************************************************************
c ***  Declare variables
      Real             t(10), rjz, rj1, rj2, sq_arg
      Integer          n
      if (arg .gt. 3.0) go to 130
      t(1) = (arg / 3.0) ** 2
c ***  Begin do loop
      do 120 n = 1, 5
 120     t(n+1) = t(n) * t(1)
c ***  End do loop
      rjz = 1.0 - 2.2499997 * t(1) + 1.2656208 * t(2) - 0.3163866 *
     &      t(3) + 0.0444479 * t(4) - 0.0039444 * t(5) + 0.0002100 *
     &      t(6)
      rj1 = 0.5 - 0.56249985 * t(1) + 0.21093573 * t(2) - 0.03954289
     &      * t(3) + 0.00443319 * t(4) - 0.00031761 * t(5) +
     &      0.00001109 * t(6)
      rj1 = rj1 * arg
      go to 150
 130  continue
      t(1) = 3.0 / arg
c ***  Begin do loop
      do 140 n = 1, 5
 140     t(n+1) = t(n) * t(1)
c ***  End do loop
      fz = 0.79788456 - 0.00000077 * t(1) - 0.00552740 * t(2) -
     &     0.00009512 * t(3) + 0.00137237 * t(4) - 0.00072805 * t(5)
     &     + 0.00014476 * t(6)
      tz = arg - 0.78539816 - 0.04166397 * t(1) - 0.00003954 * t(2) +
     &     0.00262573 * t(3) - 0.00054125 * t(4) - 0.00029333 * t(5)
     &     + 0.00013558 * t(6)
      sq_arg = fz / sqrt(arg)
      rjz = sq_arg * cos(tz)
      f1 = 0.79788456 + 0.00000156 * t(1) + 0.01659667 * t(2) +
     &     0.00017105 * t(3) - 0.00249511 * t(4) + 0.00113653 * t(5)
     &     - 0.00020033 * t(6)
```

```
      t1 = arg - 2.35619449 + 0.12499612 * t(1) + 0.00005650 * t(2)
     &      - 0.00637879 * t(3) + 0.00074348 * t(4) + 0.00079824 * t(5)
     &      - 0.00029166 * t(6)
        sq_arg = f1 / sqrt(arg)
        rj1 = sq_arg * cos(t1)
 150    continue
        rj2 = ((2.0 * rj1) / arg) - rjz
        return
        end

c *-------**------**------**------**------**------**------**------**---
c *-------**------**------**------**------**------**------**------**---

        Subroutine K1KZ(arg, rk2, rk1, rkz)
c ***********************************************************************
c ***  This subroutine takes in the input, arg, which in this case is***
c ***  the value of the variable beta which is calculated in the main***
c ***  program.  The output of the subroutine is the value of the    ***
c ***  Bessel functions K0, K1 and K2.                                ***
c ***********************************************************************
c ***  Declare variables.
        Integer          n
        Real             t(10), rk2, rk1, rkz
        if (arg .gt. 2.0) go to 180
        t(1) = (arg / 3.75) ** 2
c ***  Begin do loop
        do 160 n = 1, 5
 160       t(n+1) = t(n) * t(1)
c ***  End do loop
        riz = 1.0 + 3.5156229 * t(1) + 3.0899424 * t(2) + 1.2067492 *
     &       t(3) + 0.2659732 * t(4) + 0.0360768 * t(5) + 0.0045813
     &       * t(6)
        ri1 = 0.5 + 0.87890594 * t(1) + 0.51498869 * t(2) + 0.1508493 *
     &       t(3) + 0.02658733 * t(4) + 0.00301532 * t(5) + 0.00032411
     &       * t(6)
        ri1 = ri1 * arg
        t(1) = (arg / 2.0) ** 2
c ***  Begin do loop
        do 170 n = 1, 5
 170       t(n+1) = t(n) * t(1)
c ***  End do loop
        argh = arg / 2.0
        rkz = -alog(argh) * riz - 0.57721566 + 0.42278420 * t(1) +
     &       0.23069756 * t(2) + 0.03488590 * t(3) + 0.00262698 * t(4)
     &       + 0.00010750 * t(5) + 0.00000740 * t(6)
        rk1 = arg * alog(argh) * ri1 + 1.0 + 0.15443144 * t(1) -
     &       0.67278579 * t(2) - 0.18156897 * t(3) - 0.01919402 * t(4)
     &       -  0.00110404 * t(5) - 0.00004686 * t(6)
        rk1 = (rk1 / arg)
        go to 200
 180    continue
        t(1) = 2.0 / arg
c ***  Begin do loop
        do 190 n = 1, 5
 190       t(n+1) = t(n) * t(1)
c ***  End do loop
        rkz = 1.25331414 - 0.07832358 * t(1) + 0.02189568 * t(2)
```

```
     &       - 0.01062446 * t(3) + 0.00587872 * t(4) - 0.00251540 *
     &       t(5) + 0.00053208 * t(6)
       rkz = rkz / (((arg) ** 0.5) * exp(arg))
       rk1 = 1.25331414 + 0.23498619 * t(1) - 0.03655620 * t(2)
     &       + 0.01504268 * t(3) - 0.00780353 * t(4) + 0.00325614 *
     &       t(5) - 0.00068245 * t(6)
       rk1 = rk1 / (((arg) ** 0.5) * exp(arg))
 200   continue
       rk2 = rkz + ((2.0 * rk1) / arg)
       return
       end
c ***************************************************************************
c                            END OF PROGRAM
c ***************************************************************************
```

APPENDIX D

HEATER AND THERMOMETER DETAILS

The heater was manually constructed. The bottom of the thermal resistor was wrapped in cigarette paper and then coated with General Electric adhesive and insulating varnish 7031. It is very important that there be no air bubbles between the cigarette paper and the brass thermal resistor, so that the heater wire is able to make good contact with the metal. The coating of 7031 was repeated 2-3 times and then it was left under a lamp to dry overnight. The next day a two feet wire of manganin with 12 ohm/foot resistivity was wrapped around the cigarette paper which was on the thermal resistor. An inch of the wire was left unwrapped on each end of the wire. These two ends were tied down so the wire does not unwrap. Again care was taken that the turns were close to each other and that the wire did not overlap. The wire was then coated with 7031 and left under the lamp for drying. The next day the two ends of the heater wire were soldered to two inches of copper wire on each side. The wire was then wrapped around the thermal resistor in such a way that the manganin wire was completely wrapped around the resistor. The two ends of the copper wire were tied down and another coating of 7031 was applied and left to dry. The next day the two ends of the copper wire were soldered to pins 3 and 4 on the connector bracket. It is important to check that the heater is not grounded.

The thermometer is a Lake Shore Cryotronics, Inc. doughnut shaped sensor (model # DT-470-CU-12A). This sensor was screwed above the fixture. The heater and the thermometer sensor can be seen in Fig. 4(d).

APPENDIX E

RAW DATA OBTAINED FROM THE EXPERIMENTS

I. Data for polycrystalline $ZrSnTiO_4$ Sample:

Diameter of sample = 500 mils

Height of sample = 272.5 mils

Data point #1

Temperature	291 K
D_m	-5.4 dB
β	0.301
D_0	-2.576 dB
f_{lo}	5.09084 GHz
f_{hi}	5.09228 GHz
Q_0	3535.7
f_0	5.09156 GHz
ε_r	37.952
$\tan\delta$	1.064×10^{-4}
σ	4.028×10^{7} mhos/m

Data point #2

Temperature	120 K
D_m	-9.45 dB
β	0.496
D_0	-4.119 dB
f_{lo}	5.08204 GHz
f_{hi}	5.08296 GHz
Q_0	5522.7
f_0	5.08250 GHz
ε_r	38.088
$\tan\delta$	0.6803×10^{-4}
σ	9.675×10^{7} mhos/m

Data point #3

Temperature	110 K
D_m	-10.05 dB
β	0.521
D_0	-4.310 dB
f_{lo}	5.08123 GHz
f_{hi}	5.08211 GHz
Q_0	5776.2
f_0	5.08167 GHz
ε_r	38.100
$\tan\delta$	0.6441×10^{-4}
σ	10.539×10^{7} mhos/m

Data point #4

Temperature	100 K
D_m	-10.95 dB
β	0.558
D_0	-4.577 dB
f_{lo}	5.08087 GHz
f_{hi}	5.08171 GHz
Q_0	6051.2
f_0	5.08129 GHz
ε_r	38.106
$\tan\delta$	0.6100×10^{-4}
σ	11.572×10^{7} mhos/m

Data point #5

Temperature	90 K
D_m	-11.8 dB
β	0.591
D_0	-4.808 dB
f_{lo}	5.07887 GHz
f_{hi}	5.07967 GHz
Q_0	6348.0
f_0	5.07927 GHz
ε_r	38.136
$\tan\delta$	0.5785×10^{-4}
σ	12.830×10^{7} mhos/m

Data point #6

Temperature	78.5 K
D_m	-13.1 dB
β	0.637
D_0	-5.124 dB
f_{lo}	5.07831 GHz
f_{hi}	5.07907 GHz
Q_0	6681.8
f_0	5.07869 GHz
ε_r	38.145
$\tan\delta$	0.5547×10^{-4}
σ	14.940×10^{7} mhos/m

II. Data for 99.5% Al_2O_3 Sample:

Diameter of sample = 705.5 mils

Height of sample = 353.5 mils

Data point #1

Temperature	293 K
D_m	-7.0 dB
β	0.382
D_0	-3.238 dB
f_{lo}	7.38811 GHz
f_{hi}	7.38974 GHz
Q_0	4533.6
f_0	7.38893 GHz
ε_r	9.847
$\tan\delta$	1.0021×10^{-4}
σ	4.0009×10^{7} mhos/m

Data point #2

Temperature	120 K
D_m	-16.9 dB
β	0.750
D_0	-5.824 dB
f_{lo}	7.43351 GHz
f_{hi}	7.43435 GHz
Q_0	8847.9
f_0	7.43393 GHz
ε_r	9.726
$\tan\delta$	0.3557×10^{-4}
σ	9.675×10^7 mhos/m

Data point #3

Temperature	110 K
D_m	-18.4 dB
β	0.785
D_0	-6.024 dB
f_{lo}	7.43442 GHz
f_{hi}	7.43524 GHz
Q_0	9070.4
f_0	7.43483 GHz
ε_r	9.724
$\tan\delta$	0.3571×10^{-4}
σ	10.539×10^{7} mhos/m

Data point #4

Temperature	100 K
D_m	-20.0 dB
β	0.818
D_0	-6.199 dB
f_{lo}	7.43546 GHz
f_{hi}	7.43624 GHz
Q_0	9531.8
f_0	7.43585 GHz
ε_r	9.721
$\tan\delta$	0.3333×10^{-4}
σ	11.572×10^{7} mhos/m

Data point #5

Temperature	90 K
D_m	-23.8 dB
β	0.879
D_0	-6.497 dB
f_{lo}	7.43620 GHz
f_{hi}	7.43694 GHz
Q_0	10048.7
f_0	7.43657 GHz
ε_r	9.719
$\tan\delta$	0.3098×10^{-4}
σ	12.830×10^{7} mhos/m

Data point #6

Temperature	77 K
D_m	-29 dB
β	1.073
D_0	-7.219 dB
f_{lo}	7.43623 GHz
f_{hi}	7.43688 GHz
Q_0	11442.1
f_0	7.43655 GHz
ε_r	9.719
$\tan\delta$	0.2291×10^{-4}
σ	14.940×10^{7} mhos/m

III. Data for single crystal MgO Sample:

Since the MgO sample was inhomogeneous, it's dielectric characteristics were calculated by repeating the experiment twice with two different orientations and then the average of the two values of the frequency of resonance and the unloaded Q were used to calculate the relative dielectric constant and the loss tangent.

Diameter of sample = 388 mils

Height of sample = 362 mils

Data point #1

	First Run	Second run	Average
Temperature	290 K	290 K	290 K
D_m	-9.1 dB	-13.2 dB	•
β	0.481	0.641	•
D_0	-4.004 dB	-5.147 dB	•
f_{lo}	9.94587 GHz	9.93504 GHz	•
f_{hi}	9.94703 GHz	9.93618 GHz	•
Q_0	8576.9	8718.2	8647.5
f_0	9.94645 GHz	9.93561 GHz	9.94103

Following are the results obtained for the MgO sample at 290 K using the average frequency of resonance and unloaded Q value:

Temperature	290 K
Q_0	8647.5
f_0	9.94103
ε_r	9.802
$\tan\delta$	0.5074×10^{-4}

Data point #2

	First Run	Second run	Average
Temperature	120 K	120 K	120 K
D_m	-20.8 dB	-30 dB	•
β	0.833	0.939	•
D_0	-6.274 dB	-6.758 dB	•
f_{lo}	10.02905 GHz	10.01852 GHz	•
f_{hi}	10.02987 GHz	10.01944 GHz	•
Q_0	12228.7	10897.9	11563.3
f_0	10.02946 GHz	10.01898 GHz	10.02422

Following are the results obtained for the MgO sample at 120 K using the average frequency of resonance and unloaded Q value:

Temperature	120 K
Q_0	11563.3
f_0	10.02422
ε_r	9.632
tanδ	0.4514×10^{-4}

Data point #3

	First Run	Second run	Average
Temperature	110 K	110 K	110 K
D_m	-27.5 dB	-17.3 dB	•
β	0.919	1.316	•
D_0	-6.677 dB	-7.624 dB	•
f_{lo}	10.03201 GHz	10.02156 GHz	•
f_{hi}	10.03287 GHz	10.0233 GHz	•
Q_0	11662.7	13005.9	12334.3
f_0	10.03244 GHz	10.02194 GHz	10.02719

Following are the results obtained for the MgO sample at 110 K using the average frequency of resonance and unloaded Q value:

Temperature	110 K
Q_0	12334.3
f_0	10.02719
ε_r	9.626
$\tan\delta$	0.4112×10^{-4}

Data point #4

	First Run	Second run	Average
Temperature	100 K	100 K	100 K
D_m	-40 dB	-10.0 dB	•
β	1.020	1.925	•
D_0	-7.058 dB	-7.118 dB	•
f_{lo}	10.03457 GHz	10.02402 GHz	•
f_{hi}	10.03546 GHz	10.02477 GHz	•
Q_0	11266.0	13188.6	12227.3
f_0	10.03502 GHz	10.02439 GHz	10.029705

Following are the results obtained for the MgO sample at 100 K using the average frequency of resonance and unloaded Q value:

Temperature	100 K
Q_0	12227.3
f_0	10.029705
ε_r	9.621
tanδ	0.4359×10^{-4}

Data point #5

	First Run	Second run	Average
Temperature	90 K	90 K	90 K
D_m	-24 dB	-12.9 dB	•
β	1.135	1.585	•
D_0	-7.370 dB	-7.576 dB	•
f_{lo}	10.03678 GHz	10.02615 GHz	•
f_{hi}	10.03762 GHz	10.02696 GHz	•
Q_0	11959.9	12368.9	12164.4
f_0	10.03720 GHz	10.26656 GHz	10.031928

Following are the results obtained for the MgO sample at 90 K using the average frequency of resonance and unloaded Q value:

Temperature	90 K
Q_0	12164.4
f_0	10.03193
ε_r	9.616
tanδ	0.4580×10^{-4}

Data point #6

	First Run	Second run	Average
Temperature	77 K	77 K	77 K
D_m	-17 dB	-9.7 dB	•
β	1.329	1.973	•
D_0	-7.632 dB	-7.036 dB	•
f_{lo}	10.03810 GHz	10.02761 GHz	•
f_{hi}	10.03886 GHz	10.02837 GHz	•
Q_0	13207.2	13193.4	13200.3
f_0	10.03848 GHz	10.27990 GHz	10.033235

Following are the results obtained for the MgO sample at 77 K using the average frequency of resonance and unloaded Q value:

Temperature	77 K
Q_0	13200.3
f_0	10.03323
ε_r	9.613
tanδ	0.4141×10^{-4}

III. Data for single crystal $LaAlO_3$ Sample:

The $LaAlO_3$ sample was also found to be inhomogeneous. Figure 16 shows the level of inhomogeneity. To find its complex dielectric constant, an orientation position which gave a value of 8900 for the unloaded Q was chosen and the experiment was conducted. The following are the results obtained for that orientation.

Diameter of sample = 375 mils

Height of sample = 418.5 mils

Data point #1

Temperature	293K
D_m	-6.4 dB
β	0.353
D_0	-2.997 dB
f_{lo}	6.23641 GHz
f_{hi}	6.23711 GHz
Q_0	8909.7
f_0	6.23676 GHz
ε_r	24.207
$\tan\delta$	0.5349×10^{-4}
σ	4.001×10^7 mhos/m

Data point #2

Temperature	120K
D_m	-5.5 dB
β	0.306
D_0	-2.619 dB
f_{lo}	6.28569 GHz
f_{hi}	6.28627 GHz
Q_0	10832.1
f_0	6.28598 GHz
ε_r	23.821
$\tan\delta$	0.5533×10^{-4}
σ	9.675×10^{7} mhos/m

Data point #3

Temperature	100K
D_m	-4.5 dB
β	0.253
D_0	-2.177 dB
f_{lo}	6.28938 GHz
f_{hi}	6.28997 GHz
Q_0	10663.2
f_0	6.28967 GHz
ε_r	23.792
$\tan\delta$	0.598×10^{-4}
σ	11.572×10^{7} mhos/m

Data point #4

Temperature	90K
D_m	-4.5 dB
β	0.253
D_0	-2.177 dB
f_{lo}	6.29137 GHz
f_{hi}	6.29192 GHz
Q_0	11433.7
f_0	6.29165 GHz
ε_r	23.777
$\tan\delta$	0.5493×10^{-4}
σ	12.83×10^{7} mhos/m

Data point #5

Temperature	77K
D_m	-4.5 dB
β	0.253
D_0	-2.177 dB
f_{lo}	6.29269 GHz
f_{hi}	6.29322 GHz
Q_0	11868.1
f_0	6.29295 GHz
ε_r	23.767
$\tan\delta$	0.5378×10^{-4}
σ	14.94×10^{7} mhos/m

APPENDIX F

EVALUATION OF EFFECT OF PLATE HOLE ON THE MEASUREMENTS

The sample used to evaluate the effect of the hole in the top plate of the fixture was the $ZrSnTiO_4$ sample with a diameter of 500 mils and a height of 272.5 mils at room temperature (290K). The results obtained are shown below:

Regular Top Plate Experiment	Top Plate With Hole Experiment
$D_m = -3.3$ dB	$D_m = -6.0$ dB
$\beta = 0.1877$	$\beta = 0.332$
$D_0 = -1.621$ dB	$D_0 = -2.831$ dB
$f_{lo} = 5.09623$ GHz	$f_{lo} = 5.09418$ GHz
$f_{hi} = 5.09771$ GHz	$f_{hi} = 5.09565$ GHz
$Q_0 = 3443.7$	$Q_0 = 3465.7$
$f_0 = 5.09670$ GHz	$f_0 = 5.094915$ GHz
$\varepsilon_r = 37.875$	$\varepsilon_r = 37.901$
$\tan\delta = 1.1435 \times 10^{-4}$	$\tan\delta = 1.1246 \times 10^{-4}$

SELECTED BIBLIOGRAPHY

Primary Sources:

1. Aitken, J.E., "Swept-frequency microwave Q-factor measurement", Proceedings IEE , vol. 123, no. 9, September, 1976, pp. 855-862.

2. Courtney, W.E., "Analysis and evaluation of a method of measuring the complex permittivity and permeability of microwave insulators", IEEE Transactions on Microwave Theory and Techniques, vol. MTT-18, no. 18, August, 1970, pp.476-485.

3. Dielectric Resonators, editors Darko Kajfez and Pierre Guillon, Artech House Inc., 1986, p. 105.

4. ibid, p.303

5. ibid, pp.339-341.

6. Kobayashi, Y. and Tanaka, S., "Resonant modes of a dielectric rod resonator short-circuited at both ends by parallel conducting plates", IEEE Transactions on Microwave Theory and Techniques, vol. MTT-28, no. 10, October, 1980, pp. 1077-1085.

7. Matula, R.A., "Electrical resistivity of copper, gold, palladium and silver", Journal of Physical and Chemical Reference Data, vol. 8, no. 4, 1979, pp.1147-1298

Secondary Sources:

8. Bartels, R.A. and Smith, P.A., "Pressure and temperature dependence of the static dielectric constants of KCl, NaCl, LiF, and MgO", Physical Review B, vol.7, no. 8, April, 1973, pp.3885-3892.

9. Bednorz, J.G. and Muller, K.A., Z.Phys., B64, 1986, p.189

10. Birnbaum, G, and Franeau, J., "Measurement of the dielectric constant and the loss of solids and liquids by a cavity perturbation method", Journal of Applied Physics, Vol. 20, August 1949, pp.817-818.

11. Bonetti, R.R. and Williams, A.E., "Preliminary design steps for thin film superconducting filters". To be presented at IEEE MTT-1990 Symposium, Dallas, Texas, May 8, 1990.

12. Cohn, S.B. and Kelly, K.C., "Microwave measurement of high dielectric constant materials", IEEE Transactions on Microwave Theory and Techniques, vol. MTT-14, no. 9, September, 1966, pp. 406-410.

13. Dube, D.C., Lanagan, M.T., Kim, J.H. and Jang, S.J., "Dielectric measurements on substrate materials at microwave frequencies using a cavity perturbation technique", Journal of Applied Physics, vol. 63, no. 7, April 1, 1988, pp.2466-2468.

14. Fontanella, J., Andeen, C. and Schuele, D., "Low-frequency dielectric constants of a-quartz, sapphire, MgF_2, and MgO", Journal of Applied Physics, vol.45, no. 7, July, 1974, pp. 2852-2854.

15. Ligthart, L. P., "A fast computational technique for accurate permittivity determination using transmission line methods". IEEE Transactions on Microwave Theory and Techniques, vol. MTT-31, no. 3, March, 1983, pp.249-254.

16. Martinelli, M., Rolla, P.A. and Tombari, E., "A method for dielectric loss measurements by a microwave cavity in fixed resonance condition", IEEE Transactions on Microwave Theory and Techniques, vol. MTT-33, no. 9, September, 1985, pp.779-783.

17. Prakash, A., Vaid, J.K. and Mansingh, A., "Measurement of dielectric parameters at microwave frequencies by cavity-perturbation technique", IEEE Transactions on Microwave Theory and Techniques, vol. MTT-27, no. 9, September, 1979, pp.791-795.

18. Szendrenyi, B.B., Kazi, K. and Mojzes, I., "An alternative broadband method for automatic measurement of the complex permeability and permittivity of materials at microwave frequencies", IEEE Transactions on Microwave Theory and Techniques Symposium, 1988 MTT-S, pp.743-746.

19. Tamura, H., Hiroyuki and Wakino, K., "Low temperature properties of microwave dielectrics", Japanese Journal of Applied Physics, vol. 28, 1989 Supplement 28-2, pp. 21-23.

20. Tamura, H., Nishikawa, T. Wakino, K. and Sudo, T., "Metalized MIC substrates using high K dielectric resonator materials", Microwave Journal, October, 1988, pp. 117-126.

Private Communications:

21. Private communication with Dr. James Matey, David Sarnoff Research Center, Princeton, NJ.

22. Private communication with Bogdan Brycki, David Sarnoff Research Center, Princeton, NJ

23. Private communication with Dr. James Matey, David Sarnoff Research Center, Princeton, NJ. Based on calculations performed July 4, 1988.

Reports:

24. Rockwell International Science Center Report, SC5357.2TR. Sponsored by ORNL. Contract 83-519261, p. 9.