INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONE: 2240-1
CABLE: CENTRATOM - TELEX 460392 - I

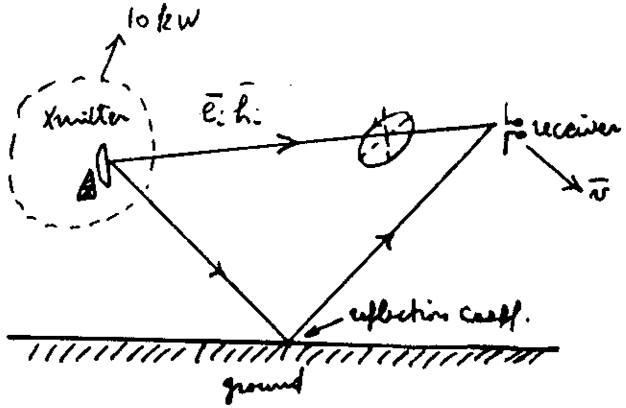
SMR.379/41

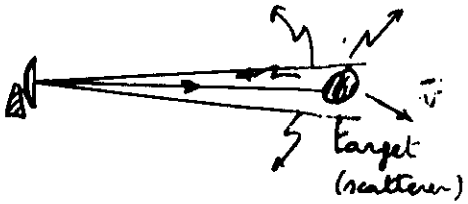
COURSE ON BASIC TELECOMMUNICATIONS SCIENCE

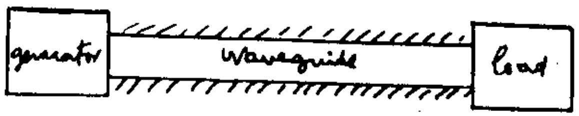
9 January - 3 February 1989

"REFRESHER COURSE"

J. VAN - BLADEL
Rijksuniversiteit Gent



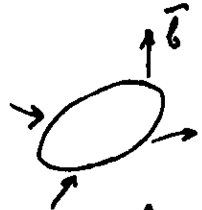




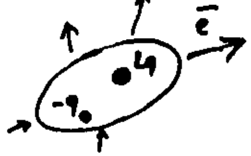
Divergence

$$\operatorname{div} \bar{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

$$\oiint_S \bar{v}\cdot\bar{u}_n \, dS = \iiint_V \operatorname{div}\bar{v} \, dV \simeq \operatorname{div}\bar{v}\,\Delta$$



$\operatorname{div}\bar{b} = 0$ "as much goes in as goes out"



$\operatorname{div}\bar{v} > 0$ "more goes out than goes in". The flux diverges

$\operatorname{div}\bar{v} < 0$ net convergence

Curl

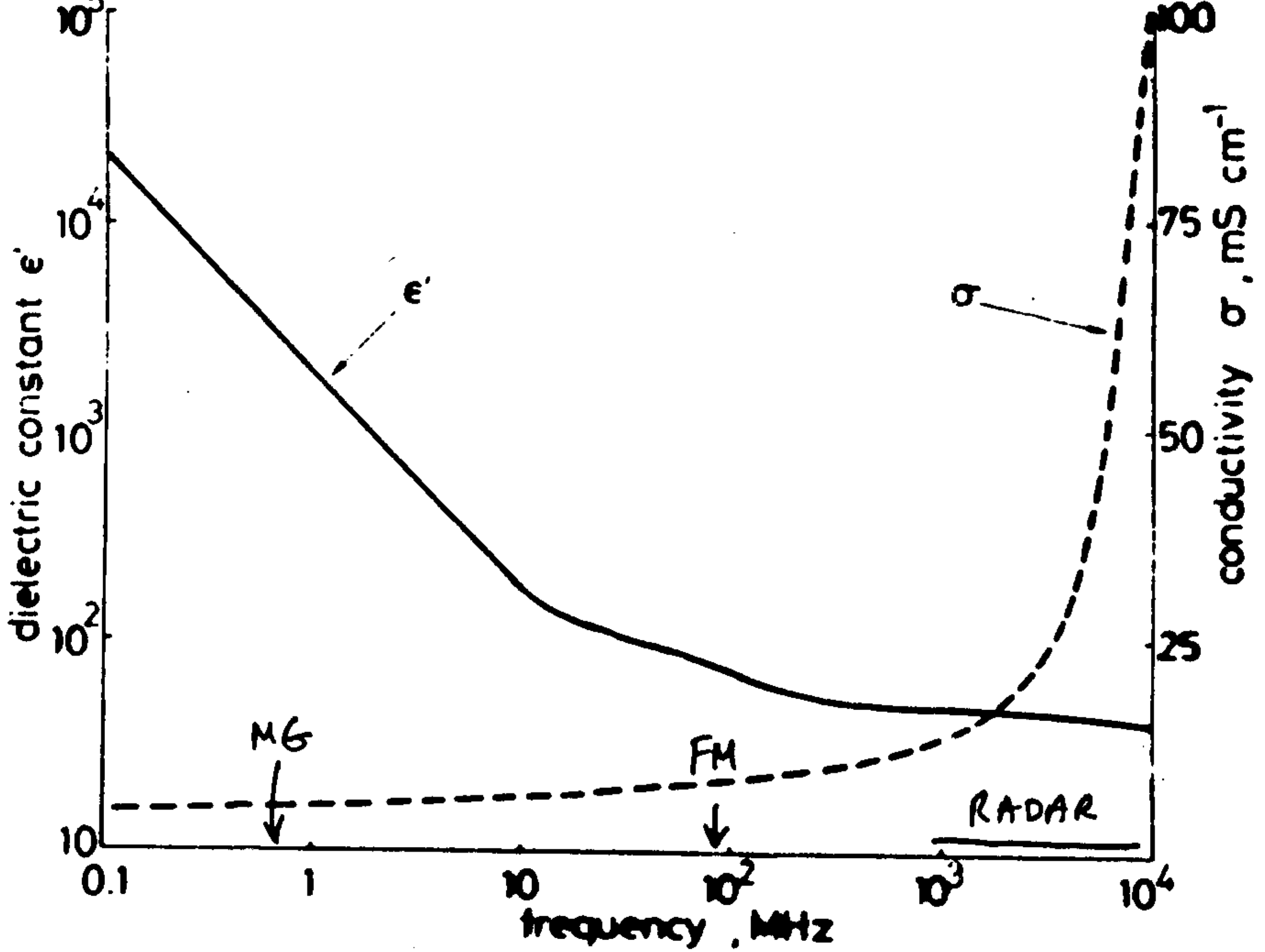
$$\operatorname{curl}\bar{v} = \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}\right)\bar{u}_x + \left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}\right)\bar{u}_y + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right)\bar{u}_z$$

$(\operatorname{curl}\bar{v})_z < 0$

$$\oint_c \bar{v}\cdot\bar{u}_c \, dc = \iint_S \bar{u}_n \cdot \operatorname{curl}\bar{v} \, dS$$

curl = tendency to rotate (in hydrodynamics)

curl = 0 irrotational flow
paddle-wheel stationary



Dielectric constant ε' and conductivity σ (mS cm⁻¹) of muscle tissue as functions of frequency

[illegible] of [illegible] in biological [illegible]*

		Muscle, Skin, and Tissues with High Water Content								
Frequency (MHz)	Wavelength in Air (cm)	Dielectric Constant (ϵ_H)	Conductivity σ_H (S/meter)	Wavelength λ_H (cm)	Depth of Penetration (cm)	Reflection Coefficient Air-Muscle Interface		Reflection Coefficient Muscle-Fat Interface		Q
						r	φ	r	φ	
1	[illegible]	2,000	0.400	436	91.3	0.982	+179	--	--	0.28
10	3,[illegible]	160	0.625	118	21.6	0.956	+178	--	--	0.14
27.12	1,[illegible]	113	0.602	68.1	14.3	0.925	+177	0.651	-11.13	0.28
40.[illegible]	738	97.3	[illegible]	51.3	11.2	0.913	+176	0.652	-10.21	0.32
[illegible]	300	70.7	[illegible]	27	6.66	[illegible]	+175	[illegible]	- 7.96	0.45
[illegible]	150	[illegible].5	[illegible]	16.6	4.79	[illegible]	+175	0.612	- 8.06	0.63
[illegible]	100	54	1.15	11.9	3.[illegible]	[illegible]	+175	[illegible]	- 8.14	0.78
433	[illegible].3	53	1.18	8.76	3.57	[illegible]	+175	0.562	- 7.[illegible]	[illegible]
[illegible]	[illegible]	52	1.25	5.34	3.18	0.779	+176	0.532	- 5.[illegible]	1.73
915	[illegible]	51	[illegible]	4.46	3.34	0.772	+177	0.519	- 4.[illegible]	2.[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	2.[illegible]	2.42	[illegible]	+177	[illegible]	- 3.[illegible]	2.[illegible]
[illegible]	12.2	[illegible]	2.17	1.76	1.[illegible]	0.754	+177	[illegible]	- [illegible]	2.96
[illegible]	[illegible]	46	2.27	1.45	1.61	0.751	+178	[illegible]	- 3.[illegible]	3.38
[illegible]	6	[illegible]	4.55	0.99	0.7[illegible]	0.749	+177	[illegible]	- 4.[illegible]	2.69
[illegible]	5.17	[illegible]	4.93	0.775	0.720	0.746	+177	[illegible]	- 4.29	2.83
[illegible]	[illegible]	[illegible]	[illegible].33	0.578	0.413	0.744	+176	0.513	- 6.[illegible]	2.13
[illegible]	3	[illegible]	10.[illegible]	0.464	0.343	0.743	+176	[illegible]	- [illegible]	2.22

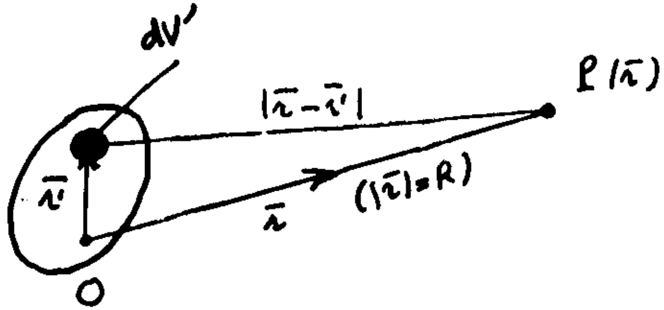
*[illegible] of [illegible] C.

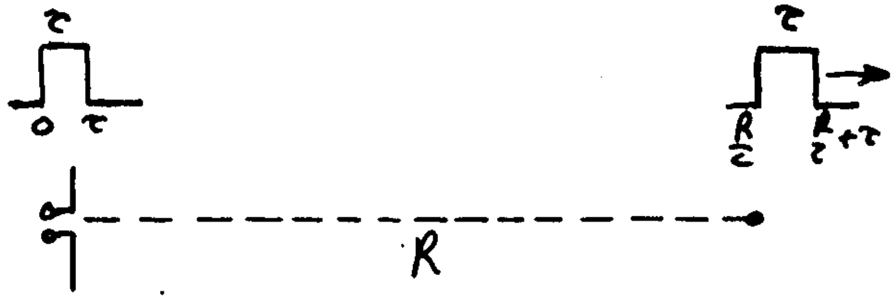
Properties of microwaves in biological media*

		Fat, Bone, and Tissues with Low Water Content								
Frequency (MHz)	Wavelength in Air (cm)	Dielectric Constant (ϵ_L)	Conductivity σ_L (mS/m)	Wavelength λ_L (cm)	Depth of Penetration (cm)	Reflection Coefficient Air-Fat Interface		Reflection Coefficient Fat-Muscle Interface		Q_{av}
						r	ϕ	r	ϕ	
1	30,000	--	--	--	--	--	--	--	--	--
10	3,000	--	--	--	--	--	--	--	--	--
27.12	1,106	20	10.9 - 43.2	241	159	0.660	+174	0.651	+169	1.11
40.68	738	14.6	12.6 - 52.8	187	118	0.617	+173	0.652	+170	1.00
100	300	7.45	19.1 - 75.9	106	60.4	0.511	+168	0.650	+172	0.87
200	150	5.95	25.8 - 94.2	59.7	39.2	0.458	+168	0.612	+172	1.10
300	100	5.7	31.6 - 107	41	32.1	0.438	+169	0.592	+172	1.38
433	69.3	5.6	37.9 - 118	28.8	26.2	0.427	+170	0.562	+173	1.73
750	40	5.6	49.8 - 138	16.8	23	0.415	+173	0.532	+174	2.48
915	32.8	5.6	55.6 - 147	13.7	17.7	0.417	+173	0.519	+176	2.81
1500	20	5.6	70.8 - 171	8.41	13.9	0.412	+174	0.506	+176	3.86
2450	12.2	5.5	96.4 - 213	5.21	11.2	0.406	+176	0.500	+176	4.84
3000	10	5.5	110 - 234	4.25	9.74	0.406	+176	0.495	+177	5.33
5000	6	5.5	162 - 309	2.63	6.67	0.393	+176	0.502	+175	6.49
5900	5.17	5.05	186 - 338	2.29	5.24	0.388	+176	0.502	+176	6.21
8000	3.75	4.7	256 - 438	1.73	4.61	0.371	+176	0.513	+173	6.09
10000	3	4.5	324 - 549	1.41	3.38	0.363	+175	0.518	+174	5.73

*Based on data from Schwan and Piersol (24) for temperature of 37° C.

$$\phi(\bar{r}, t) = \frac{1}{4\pi\varepsilon_0} \iiint \frac{\rho\left(\bar{r}', t - \frac{|\bar{r}-\bar{r}'|}{c}\right)}{|\bar{r}-\bar{r}'|} dV'$$





Dielectric constants and loss factors of common materials

Material	Dielectric constant (ε) at 10 MHz		Dielectric constant (ε) at 3000 MHz	Loss factor (ε tan δ) at 10 MHz		Loss factor (ε tan δ) at 3000 MHz
Foods						
[illegible]	50		40	1300		12
suet	4.5		2.5	4.2		0.18
Insulators and glasses						
fused quartz	3.78		3.78	0.0004		0.0002
pyrex (soda-borosilicate)	4.84		4.82	0.016		0.026
soda glass (20% Na_2O, 80% SiO_2)	6.3		–	0.19		-
Minerals						
marble, dry	9.0		9.0	0.33		0.22
ruby mica (muscovite)	5.4		5.4	0.0016		0.0016
soil, dry sandy	2.56		2.55	0.04		0.016
Natural fibres						
cotton, 200 g/m², 7% regain		1.5 at 27 MHz			0.03 at 27 MHz	
wool, 500 g/m², 20% regain		1.2 at 27 MHz			.01 at 27 MHz	
Oils and waxes						
beeswax, yellow	2.45		2.39	0.020		0.018
cable oil	2.2		2.2	0.009		0.004
paraffin wax	2.25		2.25	0.00045		0.00045
Paper and board						
paper (10% moisture measured in machine direction)			–	0.4		0.4
board (10% moisture measured in machine direction)			–	0.8		0.4
Plastics						
cellulose acetate (acetate rayon)				0.07		0.08
melamine-formaldehyde (melamine moulding compound)	5.5		4.2	0.23		0.22
phenol-formaldehyde (Bakelite)	4.3		3.7	0.18		0.15
polyamide (Nylon 66)	3.2		3.0	0.09		0.04
polyester	4.0		4.0	0.04		0.04
polyethylene (polythene)	2.25		2.25	0.0004		0.001
polymethyl methacrylate (Perspex, plexiglas)	2.7		2.6	0.027		0.015
polystyrene	2.55		2.55	0.0005		0.0085
polytetrafluoroethylene (PTFE)	2.1		2.1	0.0008		0.0003
polyvinylchloride (PVC), pure	2.9		2.8	0.03		0.02
polyvinylchloride (PVC), 40% plasticiser	3.7		2.9	0.4		0.1
Rubbers						
natural rubber	2.4		2.4	0.009		0.007
vulcanised rubber	2.5		2.5	0.08		0.03
Water						
ice, pure at −12°C	3.7		3.2	0.07		0.003
water, pure at 25°C	78.0		78.0	0.36		12.0
water, 0.5 molar NaCl	80.0		78.5	100.00		18.00
Wood						
beech, yellow (measured perpendicular to grain)	2.6		2.0	0.1		0.07
mahogany (measured perpendicular to grain)	2.5		1.9	0.07		0.06

[illegible]

BASIC TELECOM – third week

```
C=================================================================
C       This is a simple ray tracing program
C       The refractive index is a function of the height only
        common delta
        write(6,20)
20      format(' Exponential decay factor:')
        read(5,*) delta
        write(6,1)
1       format(' Starting angle in degrees:')
        read(5,*) stangle
        stangle=stangle*3.141592654/180.
        write(6,2)
2       format(' Unit step in the horizontal direction:')
        read(5,*) delx
        write(6,10)
10      format(' Number of steps:')
        read(5,*) nmax
        x=0.
        y=0.
        yex=0.
        write(6,3)
3       format(' Horizontal distance vers s height:')
        write(6,*) x,y
C
C
        ta=tan(stangle)
100     dely=delx*ta
        y=y+dely
        yex=yex+delx*tan(stangle-x/delta)
        x=x+delx
        write(6,*) x,y,yex
        call refindex(y,res)
        ta=(ta+res*delx)/(1.-ta*res*delx)
        itot=itot+1
        if(itot.eq.nmax) stop
        goto 100
C
        end
C-----------------------------------------------------------------
        subroutine refindex(y,res)
        common delta
C
C       This subroutine provides (dN/dy)/N
C
C       example: exponential profile
C               N=Aexp(-y/delta)
C
        res = -1./delta
        return
        end
C=================================================================
C       This is a simple ray tracing program
C       The refractive index is a function of the height only
        common an0,an1,delta
        write(6,20)
20      format(' Exponential decay factor:')
```

```
      read(5,*) delta
      write(6,30)
30    format(' Refractive index profile constants N0 and N1:')
      read(5,*) an0,an1
      write(6,*) an0,an1
      write(6,1)
1     format(' Starting angle in degrees:')
      read(5,*) stangle
      stangle=stangle*3.141592654/180.
      write(6,2)
2     format(' Unit step in the horizontal direction:')
      read(5,*) delx
      write(6,10)
10    format(' Number of steps:')
      read(5,*) nmax
      x=0.
      y=0.
      yex=0.
      write(6,3)
3     format(' Horizontal distance versus height:')
      write(6,*) x,y
C
C
      ta=tan(stangle)
100   dely=delx*ta
      y=y+dely
      x=x+delx
      write(6,*) x,y
      call refindex(y,res)
      ta=(ta+res*delx)/(1.-ta*res*delx)
      itot=itot+1
      if(itot.eq.nmax) stop
      goto 100
C
      end
C-------------------------------------------------------------
      subroutine refindex(y,res)
      common an0,an1,delta
C
C     This subroutine provides (dN/dy)/N
C
C     example: exponential profile on top of constant
C              N=N0 + N1exp(-y/delta)
C
      dummy=an1*exp(-y/delta)
      res=(-dummy/delta)/(an0+dummy)
      return
      end
C=============================================================
C     This program implements the method of moments
C     for a straight wire on potential 1
C
C     The charge density is taken to be piecewise constant
C     and a point-matching technique is used
C
      program wire
C
      real*4 a(100,100),x(100),r(100)
C
C     a: matrix of the problem
C     x: unknown charge density
C        (normalised on a factor 4*pi*epsilon0)
C     r: right-hand member (=1 in this particular case)
C
C     Data input
C
```

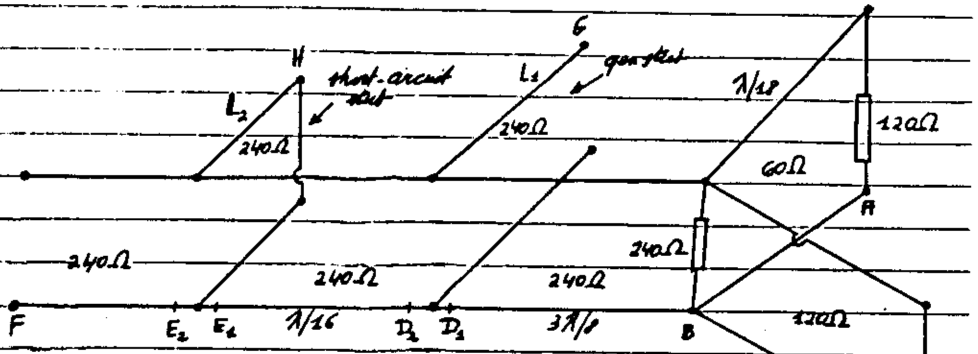
```
210   write(6,1)
1     format(' Number of divisions on the wire:')
      read(5,*) ndiv
      if(ndiv.gt.100) goto 200
C
      write(6,2)
2     format(' Total lenght of the wire:')
      read(5,*) tl
C
      write(6,3)
3     format(' Radius of the wire:')
      read(5,*) rad
C
C     Elementary division
C
      del=tl/ndiv
C
C     diagonal elements i=j
C
      do 10 i=1,ndiv
        a(i,i)=2.*alog(del/rad)
10    continue
C
C     non-diagonal elements
C
      do 20 i=1,ndiv
      do 20 j=1,ndiv
        if(i.eq.j) goto 20
        a(i,j)=1./iabs(i-j)
20    continue
C
C     right-hand member
C
      do 30 i=1,ndiv
        r(i)=1.
30    continue
C
C     Solution of the system of equations
C
      call eqn(a,r,x,ndiv)
C
C     Write the final solution to a file result.dat
C
C     We now introduce the factor 4*pi*epsilon0 = (10**-9)/9.
C     The results in RESULT.DAT are expressed in Coulomb*(10**-13),
C     hence the multiplication factor becomes (10**4)/9.
C
      fak=1.e4/9.
      open(unit=1,file='result.dat')
      write(1,*) (fak*x(i),i=1,ndiv)
      close(unit=1)
C
C     Calculate the total charge on the wire
C
      totchar=0.
      do 500 i=1,ndiv
        totchar=totchar+x(i)
500   continue
      totchar=totchar*fak*del
      write(6,501) totchar
501   format(' The total charge is: ',G20.6,' 10**-13 Coulomb')
C
      stop
C
C     To many unknowns
C
```

```
200     write(6,66)
66      format(' Number of unknowns is limited to 100!!')
        goto 210
C
        end
C-------------------------------------------------------------
        subroutine eqn(a,r,x,n)
C
C       Simple routine to solve linear system of equations
C       r(i)=Aij*xj
C       n:number of equations
C       maximum number is now 100!
C
        real*4 a(100,100),x(100),r(100)
C
        do 60 k=1,n-1
        do 10 i=k+1,n
           r(i)=a(k,k)*r(i)-r(k)*a(i,k)
        do 10 j=k+1,n
             a(i,j)=a(k,k)*a(i,j)-a(i,k)*a(k,j)
10      continue
        top=0.
        do 70 i=k+1,n
        do 70 j=k+1,n
          if(abs(a(i,j)).gt.top) top=abs(a(i,j))
70      continue
        do 80 i=k+1,n
        r(i)=r(i)/top
        do 80 j=k+1,n
        a(i,j)=a(i,j)/top
80      continue
60      continue
C
C       elimination finished - calcute unknowns
C
        x(n)=r(n)/a(n,n)
        do 30 k=1,n-1
        do 20 j=1,k
          r(n-k)=r(n-k)-a(n-k,n-j+1)*x(n-j+1)
20      continue
        x(n-k)=r(n-k)/a(n-k,n-k)
30      continue
C
        return
        end
C=================================================================
        program stubs
C
C       This program solves the double stub matching problem
C       It starts a given real characteristic impedance and a
C       given load impedance.
C       The distance between this load and the first stub and the
C       distance between the stubs must also be supplied to
C       the program
C       A proper warning is displayed if no solution turns out to be
C       possible
C
        write(6,1)
1       format(' Characteristic impedance (real):')
        read(5,*) r0
        write(6,2)
2       format(' Real and imaginary part of load:')
        read(5,*) rr,xr
        write(6,3)
3       format(' Distance from load to first stub:')
        read(5,*) an
        write(6,4)
4       format(' Distance between the stubs:')
        read(5,*) anb
        sig=1.
        call double(r0,rr,xr,an,anb,wavl1,wavl2,sig)
        write(6,5)
5       format(' First solution:')
        if(sig.ne.0.) write(6,102) wavl1,wavl2
        if(sig.eq.0.) write(6,101)
101     format(' No solution possible!')
        sig=-1.
        call double(r0,rr,xr,an,anb,wavl1,wavl2,sig)
        write(6,6)
6       format(' Second solution:')
        if(sig.ne.0.) write(6,102) wavl1,wavl2
        if(sig.eq.0.) write(6,101)
102     format(' First stub: ',G20.6,'  Second stub: ',G20.6)
        stop
        end
C-------------------------------------------------
        subroutine double(r0,rr,xr,an,anb,wavl1,wavl2,sig)
        wavl1=0.
        wavl2=0.
        pi=3.141592654
        theta=anb*720.
        call rangle(theta)
        call reflect(r0,0.,rr,xr,gm,gad)
        gad=gad+180.
        call rangle(gad)
        call spiral(gm,gad,an,0.,1.,gm,gad)
        call imped(1.,0.,gm,gad,r1,x1)
        thetar=theta*pi/180.
        d=r1*(1.-cos(thetar))
        if(d.gt.2..or.d.eq.0.) goto 100
        a=2.*r1/(r1+1.)-cos(thetar)
        b=sin(thetar)
        d=r1/(1.+r1)
        e=1./(1.+r1)
        c=d-e
        aa=a**2+b**2
        bb=2.*(b**2*d+a*d-a*e)
        cc=b**2*d**2+d*d-2.*d*e+e*e-(b*e)**2
        disc=bb**2-4*aa*cc
        if(disc.le.0..or.b.eq.0.) disc=0.
        x=(bb+sig*sqrt(disc))/(2.*aa)
        if(disc.ne.0.) y=(a*x-c)/b
        if(disc.eq.0.) y=sig*sqrt(abs(e**2-(x-d)**2))
        phir=atan2(y,x)
        phi=phir*180./pi
        gam=x/cos(phir)
        call imped(1.,0.,gam,phi,r1,x12)
        b1=x12-x1
        call reflect(1.,0.,0.,b1,gm1,ang)
        if(ang.lt.0.) ang=360.+ang
        wavl1=(360.-ang)/720.
        call spiral(gam,phi,anb,0.,1.,gm,gad)
        if(gad.lt.-90.) gad=-180.-gad
        call imped(1.,0.,gm,gad,r1,x1)
        b1=-x1
        call reflect(1.,0.,0.,b1,gm2,ang)
        if(ang.lt.0.) ang=360.+ang
        wavl2=(360.-ang)/720.
        return
100     sig=0.
        return
        end
```

```
C---------------------------------------------------------
      subroutine rectan(ra,thetad,x,y)
C
      pi=3.141592654
      thetar=thetad*pi/180.
      x=ra*cos(thetar)
      y=ra*sin(thetar)
      return
      end
C---------------------------------------------------------
      subroutine imped(r0,x0,gm,gad,rs,xs)
C
      call rectan(gm,gad,gr,gi)
      call polar(1.+gr,gi,an,anad)
      call polar(r0,x0,z0m,z0ad)
      call polar(1.-gr,-gi,ad,adad)
      call rectan(z0m*an/ad,z0ad+anad-adad,rs,xs)
      return
      end
C---------------------------------------------------------
      subroutine reflect(r0,x0,rr,xr,gm,gad)
C
      call polar(rr-r0,xr-x0,an,anad)
      call polar(rr+r0,xr+x0,ad,adad)
      gm=an/ad
      gad=anad-adad
      call rangle(gad)
      return
      end
C---------------------------------------------------------
      subroutine spiral(gm1,gad1,an,alpha,beta,gm2,gad2)
C
      dgad=720.*an
      gad2=gad1-dgad
      call rangle(gad2)
      pi=3.141592654
      attn=-4.*pi*alpha*an/beta
      gm2=gm1*exp(attn)
      return
      end
C---------------------------------------------------------
      subroutine rangle(thetad)
C
1     if(thetad.lt.360.) goto 2
      thetad=thetad-360.
      goto 1
2     if(thetad.ge.0.) goto 3
      thetad=thetad+360.
      goto 2
3     if(thetad.gt.180.) thetad=thetad-360.
      return
      end
C---------------------------------------------------------
      subroutine polar(x,y,ra,thetad)
C
      ra=sqrt(x*x+y*y)
      thetad=atan2(y,x)*180./3.141592654
      return
      end
C=========================================================
      program findif
C
C     This program solves Laplaces equation for the potential
C     by means of finite differences
C
C     It is written here to solve the symmetric stripline problem i.e.
C     a perfectly conducting strip of width bs (in arbitrary units)
C     between two perfectly conducting ground planes. The distance
C     between these planes and the strip is hsg (in arbitrary units)
C
C     In order to solve this problem the configuration is put into a
C     box of potential zero. The boundaries to the left and to the right
C     of the strip are specified by a distance bz (arbitrary units)
C     This distance must be large enough to yield good results
C     The number of divisions on the strip and between the strip and
C     the planes must also be large enough (hsg and bs must be > 15)
C     We also determine the impedance for that configuration. The medium
C     between the planes is air.
C     Typical example: strip width bs=15
C                      distance between strip and ground hsg=15
C                      distance between strip and box bz=60
C      This is a strip where the width is half the distance between
C      the planes. Characteristic impedance: 100 Ohm (see literature)
C                numerical result after about 500 iterations: 96.6 Ohm
C
      real*4 v(202,202)
      integer*4 bs,hsg,hs,bz
C
C     Data input
C
      write(6,1)
1     format(' Strip width in units:')
      read(5,*) bs
      write(6,2)
2     format(' Distance above the ground:')
      read(5,*) hsg
      hs=hsg-1
      write(6,4)
4     format(' Distance left and right of strip:')
      read(5,*) bz
C
      ista=0
600   write(6,5)
5     format(' Number of cycles [0=stop],number of iter./cycle:')
      read(5,*) ncy,niter
      if(ncy.eq.0) goto 888
C
      icy=0
      iter=0
      if(ista.ne.0) goto 200
C
C     initialise
C
      nx=bs+2*bz+2+1
      ny=hsg+hs+1+1
      do 20 i=1,nx
      do 20 j=1,ny
        v(i,j)=0.
20    continue
      iy=hsg+1
      do 21 i=bz+2,bz+bs+2
        v(i,iy)=1.
21    continue
C
C     Start iterations
C
200   do 22 j=2,ny-1
      do 22 i=2,nx-1
        if(j.ne.iy) goto 50
        if((i.lt.bz+2).or.(i.gt.bz+bs+2)) goto 50
        v(i,j)=1.
        goto 22
```

```
50        v(i,j)=(v(i-1,j)+v(i+1,j)+v(i,j+1)+v(i,j-1))/4.
22        continue
          iter=iter+1
          if(iter.ne.niter) goto 200
C
          if(ista.ne.0) goto 100
C
C         Calculation of the total capacitance with respect to the
C         plane below the strip
C
          write(6,301)
301       format(' Capacitance calculation:')
C
C         Numerical calculation of the capacitance
C
          ista=1
100       idis=2
C
          sum=0.
          iy=hsg+1+idis
          ibegin=bz+2-idis
          iend=bz+bs+2+idis
          do 303 i=ibegin,iend
            fak=1.
            if((i.eq.ibegin).or.(i.eq.iend)) fak=0.5
            sum=sum+fak*(v(i,iy+1)-v(i,iy-1))
303       continue
          iy=hsg+1-idis
          do 304 i=ibegin,iend
            fak=1.
            if((i.eq.ibegin).or.(i.eq.iend)) fak=0.5
            sum=sum-fak*(v(i,iy+1)-v(i,iy-1))
304       continue
          ix=bz+2-idis
          ibegin=hsg+1-idis
          iend=hsg+1+idis
          do 305 i=ibegin,iend
            fak=1.
            if((i.eq.ibegin).or.(i.eq.iend)) fak=0.5
            sum=sum+fak*(v(ix-1,i)-v(ix+1,i))
305       continue
          ix=bz+bs+2+idis
          do 306 i=ibegin,iend
            fak=1.
            if((i.eq.ibegin).or.(i.eq.iend)) fak=0.5
            sum=sum-fak*(v(ix-1,i)-v(ix+1,i))
306       continue
C
          sum=0.5*sum
          sum=(1000./(36.*3.141592634))*sum
          write(6,600) -sum
600       format(' The numerical calculation yields: ',G20.6)
C
          icy=icy+1
          if(icy.eq.ncy) goto 800
          iter=0
          goto 200
C
888       write(6,889)
889       format(' Impedance of the stripline configuration:')
          write(6,*) -10000./(6.*sum)
          stop
          end
C=================================================================
```

Typical double stub matching problem



problem: Determine the lengths L_1 and L_2 of the stubs in such a way that matching in F is obtained with respect to 240Ω.

solution (see also on Smith chart)

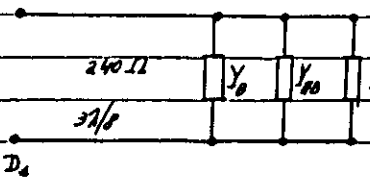
1. $Z_A = 120\Omega \Rightarrow Z'_A = Z_A/R_0 = 2 \Rightarrow Y'_A = 1/2$

2. $A \to B$: $AB = \lambda/18$ or $40°$; consequently $Y'_{AB} = 0.55 + j0.275$
   $Y_{AB} = (0.55 + j0.275)/60\Omega$

3. $Z_C = 60\Omega \Rightarrow Z'_C = 0.5 \Rightarrow Y'_C = 2$

4. $C \to B$: $CB = \lambda/8$ or $90°$; $Y'_{CB} = 0.8 - j0.6$
   $Y_{CB} = (0.8 - j0.6)/120\Omega$

5. situation in B:



$Y_{total} = Y_0 + Y_{AB} + Y_{CB}$

$Y'_{total} = Y_{total} * 240\Omega$

$= 4 \times (0.55 + j0.275)$

$+ 2 \times (0.8 - j0.6)$

$+ 1$

$= 4.8 - 0.1j$

$Y_0 = \frac{1}{Z_0} = \frac{1}{240\Omega}$

6. $B \to D_1$: $BD_1 = 3\lambda/8$ or $270°$; $y'_{D_1} = 0.4 + j\,0.925$

We now apply the matching procedure

Step 1 : the circle with real part = 1 must be rotated in the towards load direction over $E_1D_2 = \lambda/16$ or 45° degrees (see Smith chart)

Step 2 : going from D_1 to D_2 only changes the imaginary part of the admittance
whatever the length L_1, going from D_1 to D_2 we stay on a circle with real part = 0.4

Step 3 : we determine the intersection of the circles from step 1 and step 2 ; we have 2 solutions:

$y'^{(1)}_{D2} = 0.4 + 0.8j$ → admittance $B'^{(1)}_1$ of stub 1 = $-0.125j$

$y'^{(2)}_{D2} = 0.4 + 4j$ → admittance $B'^{(2)}_1$ of stub 1 = $3.075j$

Step 4 : $D_2E_1 = \lambda/16$ or $45°$ → $y'^{(1)}_{E1} = 1 + 1.575j$

$y'^{(2)}_{E1} = 1 - 6.4j$

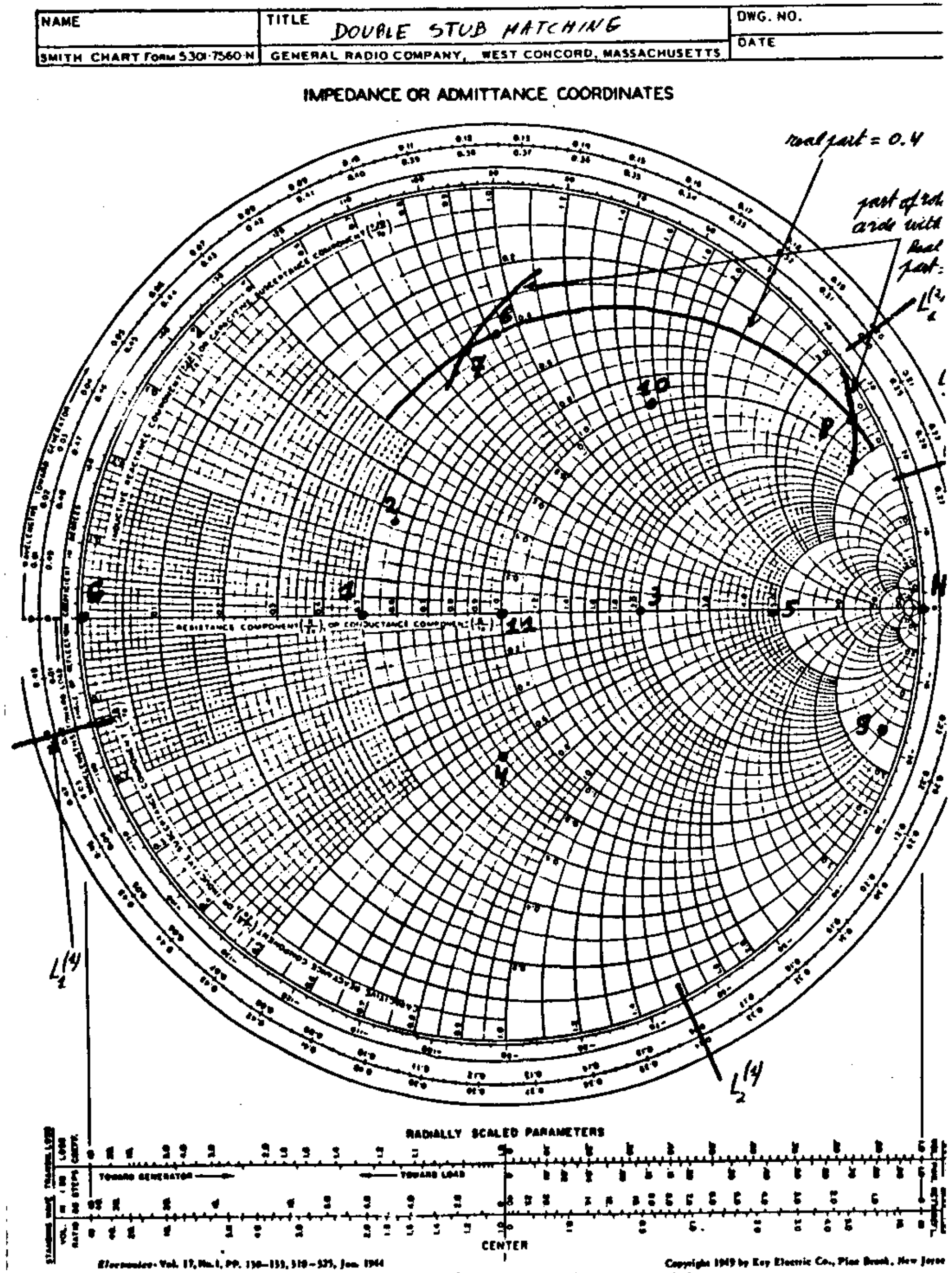
Step 5 : $y'_{E2} = 1$ (matching) → admittance $B'^{(1)}_2$ of stub 2 = $-1.575j$

admittance $B'^{(2)}_2$ of stub 2 = $6.4j$

Step 6 : length of the stubs

stub 1 starts in G ; open stub → $y' = 0$ (left hand side of Smith chart)
$B'^{(1)}_1 = -0.125j$ corresponds with 0.48λ (from G to $L^{(1)}_1$ on Smith chart)
$B'^{(2)}_1 = 3.075j$ corresponds with 0.2λ ($G \to L^{(2)}_1$)

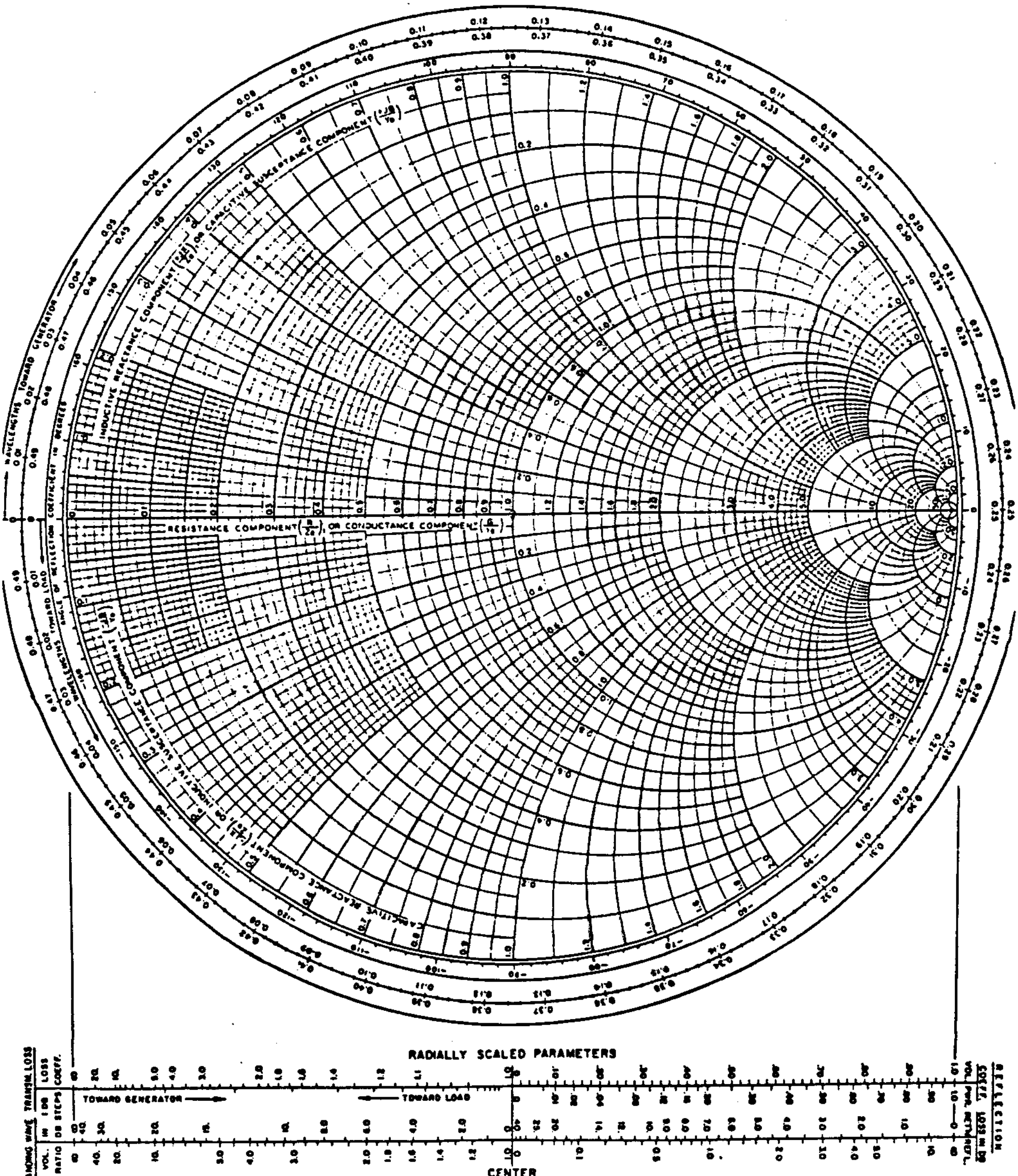
stub 2 starts in H ; short-circuited stub → $y' = \infty$ (right hand side of Smith chart)
$B'^{(1)}_2 = -1.575j$ corresponds with 0.09λ ($H \to L^{(1)}_2$)
$B'^{(2)}_2 = 6.4j$ corresponds with 0.476λ ($H \to L^{(2)}_2$)



1: y'_H 2: y'_{HB} 3: y'_C 4: y'_{CB} 5: y'_{total} 6: y'_{D1}
7: $y'^{(1)}_{D2}$ 8: $y'^{(2)}_{D2}$ 9: $y'^{(2)}_{E1}$ 10: $y'^{(1)}_{E1}$ 11: $y'_{E2} = 1$

NAME	TITLE	DATE
SMITH CHART Form 5301-7560-N	GENERAL RADIO COMPANY, WEST CONCORD, MASSACHUSETTS	

IMPEDANCE OR ADMITTANCE COORDINATES



Electronics- Vol. 17, No. 1, PP. 130–133, 318–325, Jan. 1944

Table 1.
Basic program for solution of double-stub tuners

```
      SUBROUTINE DOUBLE (RO,RR,XR,AN,ANB,WAVL1,WAVL2,SIG)
C     DETERMINES THE LENGTHS (IN WAVELENGTHS) OF
C     THE TWO SHORT-CIRCUIT STUBS FOR A DOUBLE-STUB TUNER.
C     INPUTS:
C         RO= CHARACTERISTIC IMPEDANCE (REAL).
C         RR= REAL PART OF LOAD IMPEDANCE.
C         XR= IMAGINARY PART OF LOAD IMPEDANCE.
C         AN= DISTANCE (IN WAVELENGTHS) FROM THE LOAD
C           TO THE FIRST STUB.
C         ANB=DISTANCE (IN WAVELENGTHS) FROM THE FIRST
C           STUB TO THE SECOND STUB.
C         SIG= 1. --- GIVES ONE VALID SOLUTION.
C         SIG=-1. --- GIVES ANOTHER VALID SOLUTION.
C     OUTPUTS:
C         WAVL1= LENGTH OF FIRST STUB (IN WAVELENGTHS).
C         WAVL2= LENGTH OF SECOND STUB (IN WAVELENGTHS).
          WAVL1=WAVL2=0.
          PI=4.*ATAN(1.)
          THETA=ANB*720.
          CALL RANGLE(THETA)
C     ENTER SMITH CHART AT LOAD.
          CALL REFLECT (RO,0.,RR,XR,GM,GAD)
C     ROTATE TO LOAD ADMITTANCE.
          GAD=GAD+180.
          CALL RANGLE(GAD)
C     ROTATE AN-WAVELENGTHS CLOCKWISE ON CONSTANT SWR
      CIRCLE.
          CALL SPIRAL(GM,GAD,AN,O.,1.,GM,GAD)
C     OBTAIN NORMALIZED ADMITTANCE AT FIRST STUB.
          CALL IMPED (1.,0.,GM,GAD,RL,XL)
C     TEST INPUT DATA TO SEE IF A SOLUTION IS POSSIBLE.
          D=RL*(1.-COSD(THETA))
          IF(D.GE.2..OR.D.EQ.0.)PRINT5
          IF(D.GE.2..OR.D.EQ.0.)RETURN
    5     FORMAT(1X,"SOLUTION NOT POSSIBLE.")
C     SOLVE SIMULTANEOUS EQUATIONS FOR TWO CIRCLES.
          A=2.*RL/(RL+1.)-COSD(THETA)
          B=SIND(THETA)
          D=RL/(1.+RL)
          E=1./(1.+RL)
          C=D-E
          AA=A**2+B**2
          BB=2.*(B**2*D+A*D-A*E)
          CC=B**2*D**2+D**2-2.*D*E+E*E-(B*E)**2
          DISC=BB**2-4*AA*CC
          IF(DISC.LE.0..OR.B.EQ.0.)DISC=0.
          X=(BB+SIG*SQRT(DISC))/(2.*AA)
          IF(DISC.NE.0.)Y=(A*X-C)/B
          IF(DISC.EQ.0.)Y=SIG*SQRT(ABS(E**2-(X-D)**2))
          PHI=ATAN2(Y,X)*180./PI
      SUBROUTINE DOUBLE 73/72 OPT=0 TRACE DEBUG FTN4.6+420

          GAM=X/COSD(PHI)
          CALL IMPED(1.,0., GAM,PHI,RL,XL2)
C     OBTAIN LENGTH OF STUB NEAREST LOAD --- WAVL1.
          BL=XL2-XL
          CALL REFLECT(1.,0.,0.,BL,GM1,ANG)
          IF(ANG.LT.0.)ANG=360.+ANG
          WAVL1=(360.-ANG)/720.
C     ROTATE ANB WAVELENGTHS CLOCKWISE ON CONSTANT SWR
      CIRCLE.
          CALL SPIRAL (GAM,PHI,ANB,0.,1.,GM,GAD)
          IF(GAD.LT.-90.)GAD=-180.-GAD
C     OBTAIN LENGTH OF STUB FARTHEST FROM LOAD --- WAVL2.
          CALL IMPED(1.,0.,GM,GAD,RL,XL)
          BL=-XL
          CALL REFLECT(1.,0.,0.,BL,GM2,ANG)
          IF(ANG.LT.0.)ANG=360.+ANG
          WAVL2=(360.-ANG)/720.
          RETURN
          END
```

```
          PROGRAM TEST(INPUT,OUTPUT)
          RO=50.$RR=150.$XR=-50.$AN=.216$ANB=0.0
          DO 2 I=1,2
          SIG=FLOAT(2*I-3)
          PRINT 3
    3     FORMAT(T6,"RO",T16,"RR",T26,"XR",T36,"AN",
          1T46,"ANB",T56,"SIG")
          PRINT 4,RO,RR,XR,AN,ANB,SIG
    4     FORMAT(/,1X,6F10.4,//)
          CALL DOUBLE(RO,RR,XR,AN,ANB,WAVL1,WAVL2,SIG)
          PRINT 1
    1     FORMAT(T6,"WAVL1"T16,"WAVL2")
          PRINT5,WAVL1,WAVL2
    5     FORMAT(1X,2F10.4)
    2     CONTINUE
          STOP
          END
```

Table 2:
Subroutines called for in DOUBLE program

```
          SUBROUTINE RECTAN(RA,THETAD,X,Y)
C
C         CHANGE FROM POLAR TO RECTANGULAR COORDINATES
C
C         INPUT PARAMETERS:
C             RA=MAGNITUDE
C             THETAD=ANGLE IN DEGREES
C         OUTPUT PARAMETERS:
C             X=ABSCISSA
C             Y=ORDINATE
C
              X=RA*COSD(THETAD)
              Y=RA*SIND(THETAD)
              RETURN
              END
```

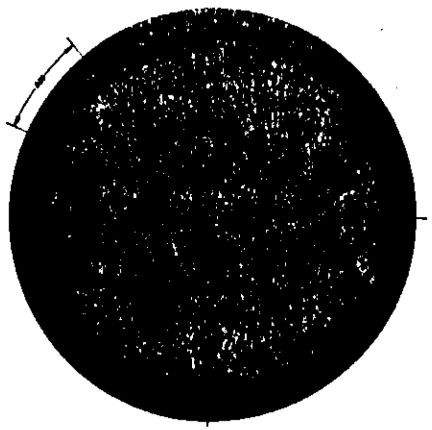
```
          SUBROUTINE IMPED(RO,X0,GM,GAD,RS,XS)
C
C         LEAVE SMITH CHART BY CHANGING FROM REFLECTION
C         COEFFICIENT TO IMPEDANCE
C
C         INPUT PARAMETERS:
C             RO=REAL PART OF CHARACTERISTIC IMPEDANCE
C             X0=IMAGINARY PART OF CHARACTERISTIC IMPEDANCE
C             GM=MAGNITUDE OF REFLECTION COEFFICIENT
C             GAD=ANGLE OF REFLECTION COEFFICIENT IN DEGREES
C         OUTPUT PARAMETERS:
C             RS=REAL PART OF SENDING-END IMPEDANCE
C             XS=IMAGINARY PART OF SENDING-END IMPEDANCE
C
              CALL RECTAN(GM,GAD,GR,GI)
              CALL POLAR(1.+GR,GI,AN,ANAD)
              CALL POLAR(R0,X0,ZOM,ZOAD)
              CALL POLAR(1.-GR,-GI,AD,ADAD)
              CALL RECTAN(ZOM*AN/AD,ZOAD+ANAD-ADAD,RS,XS)
              RETURN
              END
```

```
          SUBROUTINE REFLECT(R0,X0,RR,XR,GM,GAD)
C
C         ENTER SMITH CHART BY OBTAINING REFLECTION COEFFICIENT
C
C         INPUT PARAMETERS:
C             R0=REAL PART OF CHARACTERISTIC IMPEDANCE
C             X0=IMAGINARY PART OF CHARACTERISTIC IMPEDANCE
C             RR=REAL PART OF RECEIVING-END IMPEDANCE
C             XR=IMAGINARY PART OF RECEIVING-END IMPEDANCE
C         OUTPUT PARAMETERS:
C             GM=MAGNITUDE OF REFLECTION COEFFICIENT
C             GAD=ANGLE OF REFLECTION COEFFICIENT IN DEGREES
                BETWEEN -180 and 180
C
              CALL POLAR(RR-R0,XR-X0,AN,ANAD)
              CALL POLAR(RR+R0,XR+X0,AD,ADAD)
              GM=AN/AD
              GAD=ANAD-ADAD
              CALL RANGLE(GAD)
              RETURN
              END
```

```
          SUBROUTINE SPIRAL(GM1,GAD1,AN,ALPHA,BETA,GM2,GAD2)
C
C         MOVES OPERATING POINT ALONG SPIRAL IN SMITH CHART FROM
C         STARTING REFLECTION COEFFICIENT TO FINISHING COEFFICIENT
C
C         INPUT PARAMETERS:
C             GM1=MAGNITUDE OF STARTING REFLECTION COEFFICIENT
C             GAD1=ANGLE IN DEGREES OF STARTING REFLECTION COEF-
          FICIENT
C             AN=LENGTH OF LINE IN WAVELENGTHS
C                =POSITIVE FOR MOVEMENT TOWARD GENERATOR
C                =NEGATIVE FOR MOVEMENT TOWARD LOAD
C             ALPHA=ATTENUATION CONSTANT
C             BETA=PHASE CONSTANT
C         OUTPUT PARAMETERS:
C             GM2=MAGNITUDE OF FINISHING REFLECTION COEFFICIENT
C         GAD2=ANGLE (DEGREES) OF FINISHING REFLECTION COEFFI-
          CIENT BETWEEN -180 AND 180
C
C
              DGAD=720.*AN
              GAD2=GAD1-DGAD
              C\LL RANGLE(GAD2)
              PI=3.141592654
              ATTN=-4.*PI*ALPHA*AN/BETA
              GM2=GM1*EXP(ATTN)
              RETURN
              END
```

```
          SUBROUTINE RANGLE(THETAD)
C
C         CHANGES THETAD TO RANGE -180 TO +180
C
C         INPUT:
C             THETAD=ANY ANGLE IN DEGREES
C         OUTPUT:
C             THETAD=SAME ANGLE CONVERTED TO RANGE
C               -180.LT.THETAD.LE.+180
C
    1         IF(THETAD.LT.360.)GO TO 2
              THETAD=THETAD-360.
              GO TO 1
    2         IF(THETAD.GE.0.)GO TO 3
              THETAD=THETAD+360.
              GO TO 2
    3         IF(THETAD.GT.180.)THETAD=THETAD-360.
              RETURN
              END
```

```
          SUBROUTINE POLAR(X,Y,RA,THETAD)
C
C         CHANGE FROM RECTANGULAR TO POLAR COORDINATES
C
C         INPUT PARAMETERS:
C             X=ABSCISSA
C             Y=ORDINATE
C         OUTPUT PARAMETERS:
C             RA=MAGNITUDE
C             THETAD=ANGLE IN DEGREES BETWEEN -180 AND 180
C
              RA=SQRT(X*X+Y*Y)
              THETAD=ATAN2(Y,X)*180./3.141592654
              RETURN
              END
```



2. *If the circle of constant susceptance passing through Y_4/Y_0 doesn't intercept circle A, the computer reports that a solution is impossible.*

3. *The circle A intercepts Y_3/Y_0 and Y_3'/Y_0 which are th[illegible] two possible solutions. Subtracting Y_4/Y_0 from Y_3/Y_0 pro[illegible]vides WAVL1.*

Y_1/Y_0 = 1. (line 62). Finally, also find the required wavelength WAVL2 of this stub by the single-stub method[1] (lines 63 to 65).

How to use subroutine DOUBLE

The input and output arguments for subroutine DOUBLE are explained in the comment cards of the listing in Table 1. When you use SIG = 1.0 for the last argument of the subroutine, you will obtain solution #1 for WAVL1 and WAVL2, and when you use SIG = −1.0 for this argument, you will obtain the other possible solution. When first using subroutine DOUBLE, it might be wise to add print statements after statements 25, 29, 51, and 61. In this way, you can verify the entire analysis by plotting these points.

As an example of the use of DOUBLE, consider the problem where Z_0 = 100 Ω, Z_a = 200 - j100 Ω, the distance from load to the first stub is 0.05 wavelengths, and the separation of the stubs is 0.30 wavelengths. In the main program, use the statement:

CALL DOUBLE (100., 200., −100., 0.05, 0.30, WAVL1, WAVL2, 1.0)

and then print out the values of WAVL1 and WAVL2. This constitutes one of two possible solutions. If you call DOUBLE again with −1.0 instead of 1.0 for the last argument, you obtain the other possible solution.

1. WAVL1 = 0.1005, WAVL2 = 0.0970
2. WAVL1 = 0.2084, WAVL2 = 0.3557

For another design problem, consider the case where the frequency varies over a given range. In this case, the arguments XR, AN, and ANB take on values that depend on the frequency. The CALL DOUBLE statement and the print statements can be put in the same Do-loop that defines X_a AN, and ANB. In this way, you obtain hundreds of solutions in a matter of seconds.

For a final example, consider a case that is difficult and inaccurately solved by the usual Smith-chart method. Suppose Z_0 = 500 Ω, Z_a = 1000 + j500 Ω, AN = 0.125, and

4. *Moving Y_3/Y_0 clockwise along a circle of consta[illegible] VSWR for ANB wavelengths provides Y_2/Y_0. The admi[illegible]tance of the second stub is found by (Y_2/Y_0 −1).*

ANB = 0.256. Carrying out the Smith-chart solution, [illegible] find that the two crossings of the A circle with the b [illegible] circle are close together and near the center of the [illegible] The computer program always gives the same [illegible] accurate to as many places as desired.

1. WAVL1 = 0.1202, WAVL2 = 0.2400
2. WAVL1 = 0.1250, WAVL2 = 0.2500

The author thanks his students who patiently tested [illegible] subroutine for all possible variations of inputs. ■■

References

1. W. J. Remillard, "Nine FORTRAN Subroutines Do Smith-Chart Chores," Micro[illegible] Vol. 16, No. 12, pp. 172-177, (December, 1977).
2. W. C. Johnson, Transmission Lines and Networks, McGraw-Hill Book Co[illegible] New York, (1950).
3. D. W. Dearholt and W. R. McSpadden, Electromagnetic Wave Propagation, Mc[illegible] Hill Book Company, New York, ([illegible]).

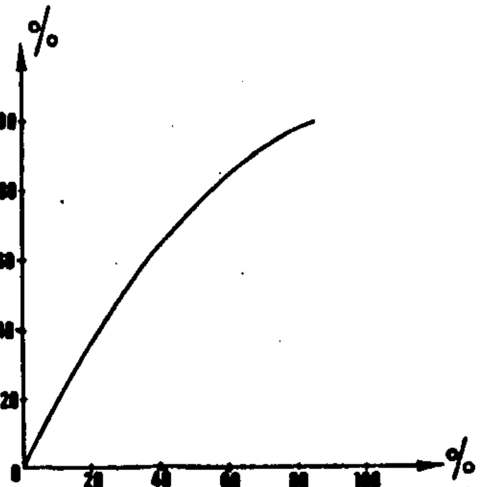


Fig 1

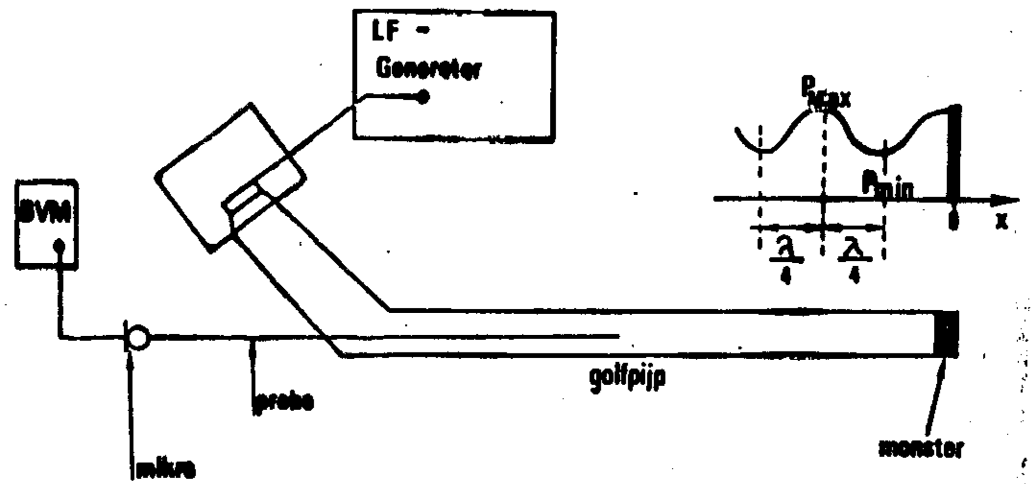


Fig 2