

TR-O-0083

49

ビーム伝搬法による
分布屈折率型光導波路の解析

馬場 清一

1995. 3. 31

ATR光電波通信研究所

目次

1 概要	1
2 光導波路の伝搬特性解析	2
3 分布屈折率型光導波路解析の実際	2
3.1 拡散型光導波路の2次元化と導波モード	2
3.2 励振源	4
3.3 光導波路形状の定式化	6
4 分岐型光導波路に関する提案と設計	8
4.1 折れ曲がり光導波路の特性	8
4.2 等エネルギー3分岐・4分岐光導波路	8
5 あとがき	11
付録 プログラムリスト	16

1 概要

ATR光電波通信研究所では、腕時計型移動体通信器（リストフォン）の実現に向けて、モノリシックマイクロ波集積回路（MMIC）の研究が進められてきた。その研究成果は、線路一体型FETや多層課MMIC技術として広く知られ、本分野における業績は大きい。一方で、このような超小型移動端末の存在を可能とする通信ネットワークとして、光とミリ波伝送を融合した、光ファイバミリ波サブキャリア通信の研究を進めてきた。こうした、新しい通信ネットワークの概念を浸透させ、発展させるには、必要とされる機能をハードウェア化することが重要だと考える。こうした観点から、光マイクロ波融合なる集積回路の概念を導入する。既に報告者らは、MMIC技術をベースとした光マイクロ波集積回路（MMICに光検出機能を付与するO/Eトランステューサ）や、ミリ波信号処理回路（E/Eシグナルプロセッサ）を実現している。光ファイバミリ波サブキャリア通信システムには、E/OトランステューサやO/Oシグナルプロセッサ等のハードウェア化も必要と考えられる。これらは光集積回路技術をベースとするものである。ここでは、光集積回路において基本的な受動回路機能である光導波路について、その伝搬特性解析手法を述べる。

2 光導波路の伝搬特性解析

光集積回路技術の基礎として光導波路の解析は重要であるが、現実の光集積回路では、形状や解析精度などの観点から解析的手法が適用できず、数値的解析法に頼らざるをえなくなる場合がある。数値的解析には有限要素法、境界要素法、階段接続法、ビーム伝搬法がある。有限要素法は光導波路構造が伝搬方向に変化しない任意形状の光導波路の伝搬モード解析等に適している。これに対して、ビーム伝搬法はY分岐合波器やテーパ型光導波路等の伝搬方向に形状変化のある場合の波動伝搬特性の解析に威力がある。ここでは、ビーム伝搬法を取り上げた。また、ビーム伝搬法の数学的取り扱いは下記のテクニカルレポートに良く記述されているので、ご参考願いたい [1]。

参考文献

- [1] A T R テクニカルレポート TR-O-0073、大木和美「ビーム伝搬法によるステップ型光導波路の解析」

3 分布屈折率型光導波路解析の実際

実際の光集積回路では、例えば、電気光学結晶LiNbO₃にTi拡散することによりコア部を形成し光導波路とする。したがって、本導波路のコア部は分布屈折率型となる。ここでは、実際の拡散型光導波路の設計について述べる。

3.1 拡散型光導波路の2次元化と導波モード

図1に試作に用いる光導波路の断面形状を示す。また、ガウス分布を仮定すると、屈折率分布は次式で表される。

$$\begin{aligned} n(x) &= n_s + (n_f - n_s) \cdot \exp\left(-\frac{x^2}{d^2}\right) \quad \text{for } x \geq 0 \\ &= n_c \quad \text{for } x < 0 \end{aligned} \tag{1}$$

光線近似を利用してスラブ化する。すなわち、深さ方向の屈折率プロファイルを等価屈折率に置き換える。シングルモード導波路であるから、拡散深さは次式で表される。

$$V_d = \sqrt{2\pi} \cdot \left(m + \frac{3}{4}\right) = \frac{2\pi}{\lambda} \cdot d \cdot \sqrt{n_f^2 - n_s^2} \tag{2}$$

$$d = \frac{\lambda}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{n_f^2 - n_s^2}} \cdot \left(m + \frac{3}{4}\right) = 4.3199 \mu m \tag{3}$$

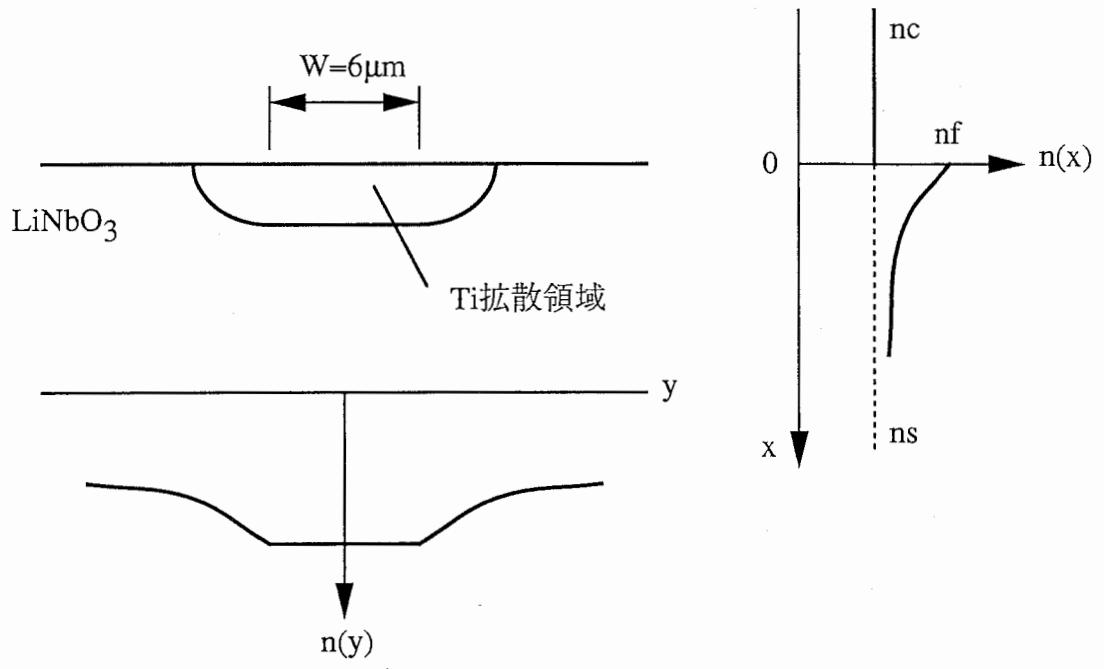


図1、分布屈折率型光導波路

但し、 $m = 1$ 、 $n_f = 2.212$ 、 $n_s = 2.202$ 、 $\lambda = 1.3\mu m$ である。

一方、試作に用いた光導波路の屈折率プロファイルから、拡散深さを求める以下のようにになり、計算結果とよく一致していることが判る。

$$d = 4.3044\mu m, \quad (x = \Delta n/e) \quad (4)$$

一般に、 m 次モードの等価屈折率は次式の超越方程式を解くことにより得られる。

$$2 \cdot V_d = \int_0^{\sqrt{-\ln b}} \sqrt{\exp(-\zeta^2) - b} \, d\zeta - (2m + \frac{3}{2})\pi = 0 \quad (5)$$

但し、

$$b = (N^2 - n_s^2)/(n_f^2 - n_s^2) \quad (6)$$

$$\zeta = \frac{x}{d} \quad (7)$$

$$v_d = K_0 \cdot d \sqrt{n_f^2 - n_s^2} \quad (8)$$

シングルモード伝搬条件時 ($m = 0$) の等価屈折率 N は、

$$d = 4.31991599$$

$$N = 2.20609901$$

$$V_d = 4.38658323$$

である。

従って、拡散型光導波路は、図 2 に示す屈折率分布を有するスラブ型光導波路として取り扱うことができる。ビーム伝搬法の計算時にはこの屈折率分布を仮定する。

3.2 励振源

伝搬特性の解析時には、伝搬の初期値（励振源）を付与する必要がある。一般には解析する光導波路の固有モードを求め、界分布を付与する。ここでは、計測及び評価においてシングルモードレーザ光源を利用することを考慮し、1 次元のガウス状ビームを仮定した。さらに、計測時にはレーザ光源と光導波路のカップリング係数を最大となるように調整することから、ガウス状ビームのスポットサイズと拡散深さは等しい ($S = d$) と仮定した。

$$E_y(x) = \exp\left(-\frac{x^2}{S^2}\right) \quad (9)$$

$$S = 4.3199\mu m$$

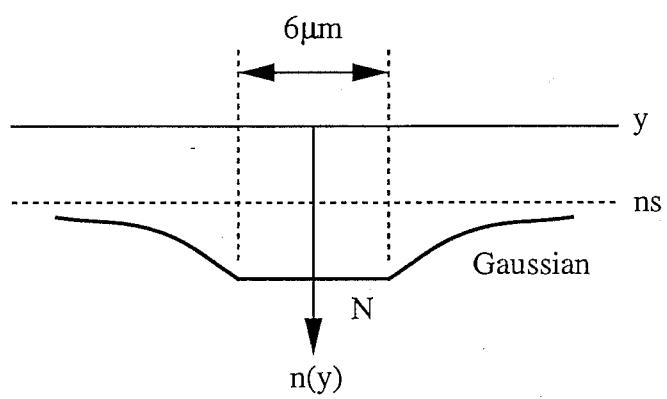


図2、等価屈折率法による1次元化

3.3 光導波路形状の定式化

計算機シミュレーションにより光導波路の伝搬特性を解析することから、光導波路、すなわち、コア部の屈折率分布を伝搬方向Zに対して定式化する。例えば、図3に示すような理想的な分布型光導波路（1次元直線）であれば、X Z平面内における屈折率は次式で表すことができる。

$$n(x) = n_s + (N - n_s) \cdot \exp\left(-\frac{(x - a)^2}{d^2}\right) \quad (10)$$

また、図4に示すようなY分岐では、次式のように書けばよい。

$$\begin{aligned} n(x) = n_s &+ (N - n_s) \cdot \exp\left(-\frac{(x - x_1)^2}{d^2}\right) \\ &+ (N - n_s) \cdot \exp\left(-\frac{(x - x_2)^2}{d^2}\right) \end{aligned} \quad (11)$$

実際の光導波路では、有限の幅Tがあるので、図5の如く条件分岐により屈折率情報を付与する。

$$\begin{aligned} \text{if } & ABS(x - a) \leq \frac{T}{2} \\ & n(x) = N \\ \text{else } & n(x) = n_s + (N - n_s) \cdot \exp\left(-\frac{(x-a)^2}{d^2}\right) \end{aligned} \quad (12)$$

さらに、方向性結合器等の光導波路が近接する構造では、Ti拡散領域が相互に重なり合い、屈折率分布は複雑になる。このときの条件分岐は以下のようにすればよい（図6）。

$$\begin{aligned} \text{if } & ABS(x - x_1) \leq \frac{T}{2} \quad \text{or} \quad ABS(x - x_2) \leq \frac{T}{2} \\ & n(x) = N \\ \text{else } & n(x) = n_s + (N - n_s) \cdot \left[\exp\left(-\frac{(x-x_1)^2}{d^2}\right) + \exp\left(-\frac{(x-x_2)^2}{d^2}\right) \right] \end{aligned} \quad (13)$$

上記を参考として、今回試作した3分岐光導波路は、次式のように定式化できる。ただし、1つのクラッド領域はこれに隣接するコア部（Ti拡散部）のみから影響を受けると仮定した。例えば、図7において、領域2はコア1とコア2の影響を受け、コア3には影響されない。あるいは各コアが重なった場合は、屈折率分

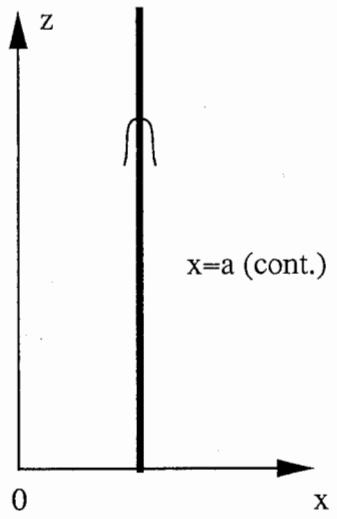


図3、理想的な光導波路（1次元直線）

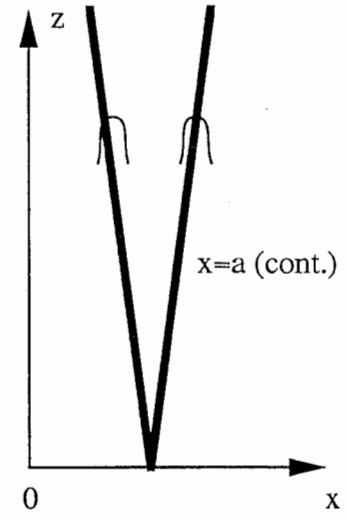


図4、理想的なY分岐光導波路

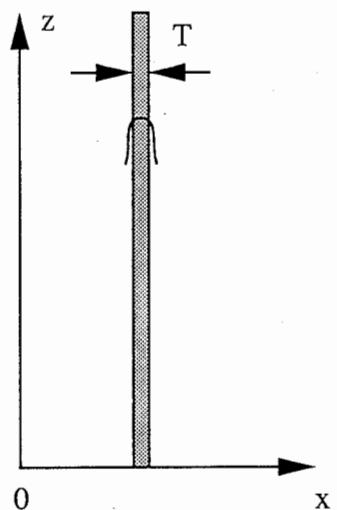


図5、有限の幅を持つ光導波路

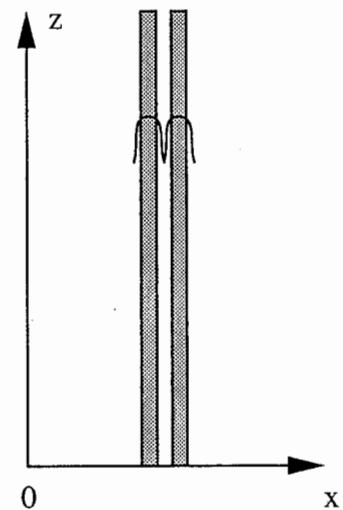


図6、光方向性結合器

布を重複させてはならない（これらは試作プロセスを想定している）。

$$\begin{aligned}
 n_5(x) &= N \\
 n_1(x) &= n_s + (N - n_s) \cdot \exp\left(-\frac{(x - x_1)^2}{d^2}\right) \\
 n_2(x) &= n_s + (N - n_s) \cdot \exp\left(-\frac{(x - x_1)^2}{d^2}\right) \\
 &\quad + (N - n_s) \cdot \exp\left(-\frac{(x - x_2)^2}{d^2}\right) \\
 n_3(x) &= n_s + (N - n_s) \cdot \exp\left(-\frac{(x - x_2)^2}{d^2}\right) \\
 &\quad + (N - n_s) \cdot \exp\left(-\frac{(x - x_3)^2}{d^2}\right) \\
 n_4(x) &= n_s + (N - n_s) \cdot \exp\left(-\frac{(x - x_3)^2}{d^2}\right)
 \end{aligned} \tag{14}$$

4 分岐型光導波路に関する提案と設計

分岐型光導波路は光波の合成・分配を行い、光スイッチ、光変調器、光ビームフォーミング等に用いられる。こうした光回路では光波を2分岐するだけでなく、4分岐、9分岐、16分岐などの多分岐も必要になる。従来、光波エネルギーを等分配・等合成を行うには2分岐(Y分岐)しか実現できず、それ以上の多分岐回路ではY分岐回路をカスケード接続を利用する他なかった（従って、2のべき乗数のみしか分岐数がとり得ない）。ここでは、3分岐ならびに4分岐回路に關しても、光波エネルギーを等分配・等合成を行う構造を提案し、試作実施を前提とした設計結果について述べる。

4.1 折れ曲がり光導波路の特性

分岐型光導波路の折れ曲がり角度の設定は、放射損失の観点から重要である。損失を小さくするために、分岐角度による放射損失依存性を計算した。これによれば、分岐角2度以上は放射損失が高く急激に増大する。従って、設計する分岐型光導波路の分岐角は2度以下とした（図8）。

4.2 等エネルギー3分岐・4分岐光導波路

図9に示す構造で、パラメータLを最適に設定すると等分岐・等合成が実現し得る[5]。図10以降にビーム伝搬法を用いた計算結果を示す。尚、設計プログ

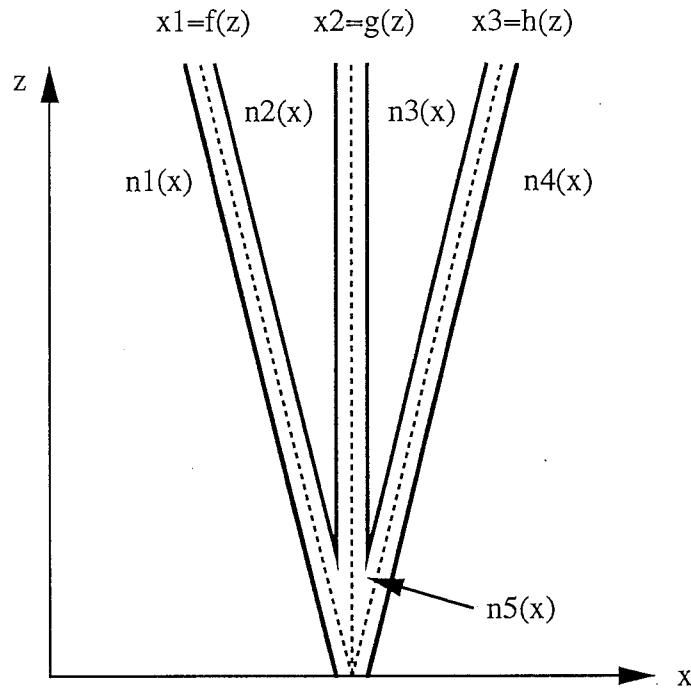


図 7、3 分岐光導波路の定式化

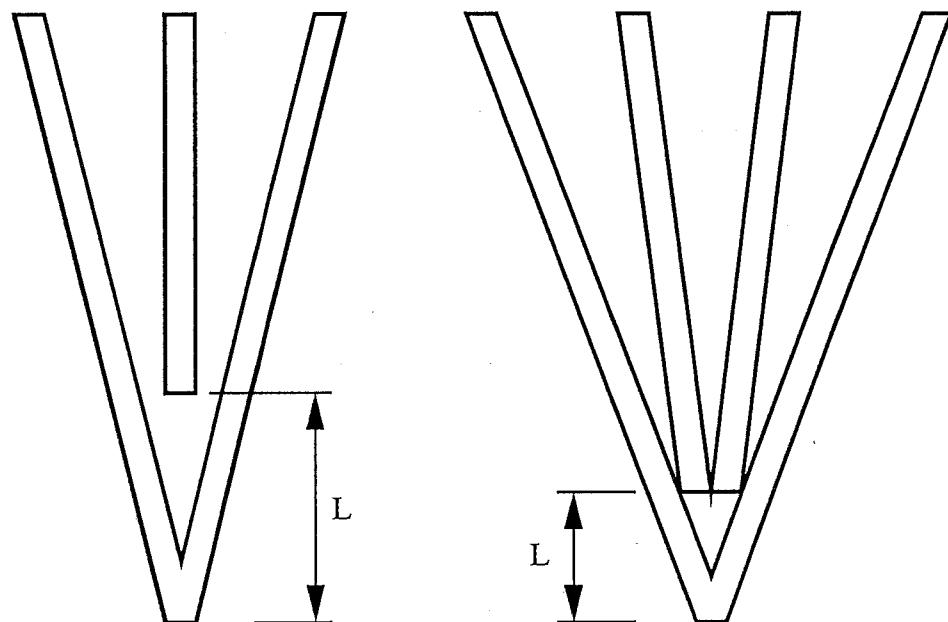


図 9、等エネルギー分岐型光導波路の提案

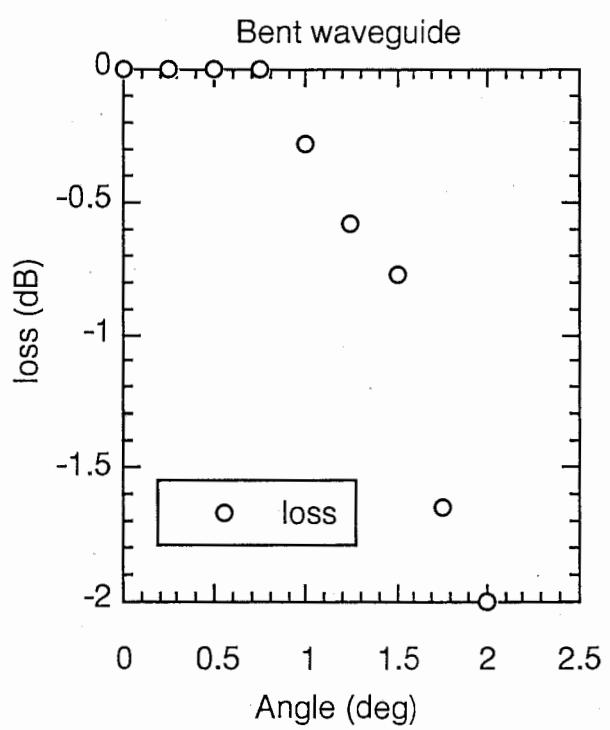
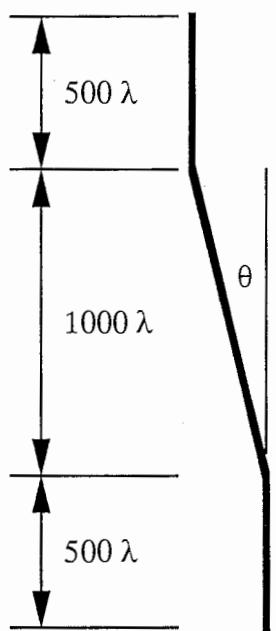


図8、計算した折れ曲がり光導波路の構造と分岐損失

ラムは自作で、そのプログラムリストは付録に添付した。計算結果において、分岐角の小さいものは単純な依存性ではない。これは、導波路間の結合、共振等の現象を示しているものと考えられる。また、分岐損が小さいのも本提案の特徴である。

5 あとがき

ここでは、分布屈折率型光導波路の数値計算としてビーム伝搬法について述べ、また分岐型光導波路構造に関する提案を行なった。本提案は、光集積回路の小型化、高機能化、高性能化に適用できる。

謝辞

本研究を進めるにあたって、直接ご指導頂いた小川博世主幹研究員（現在NTT無線システム研究所）ならびに今井主任研究員に深謝いたします。また、本研究の遂行にあたり、討論を通じて有益なるご教示を頂いた葉原会長、古濱社長（現在CRL）、猪股社長、赤池室長（現在東京理科大学）、小川英一室長ならびに無線通信第2研究室の皆様に感謝いたします。

参考文献

- [1] ATR テクニカルレポート TR-O-0073、大木和美「ビーム伝搬法によるステップ型光導波路の解析」
- [2] 岡本勝就「光導波路の基礎」、コロナ社(1992)
- [3] 山下栄吉監修、「電磁波問題の基礎解法」、電子情報通信学会(1987)
- [4] 西原浩、春名正光、楠原敏明「光集積回路」、オーム社(1985)
- [5] Seiichi Banba and Hiroyo Ogawa, "Novel Symmetrical Three-Branch Optical Waveguide with Equal Power Division", *IEEE Microwave and Guided Wave Letters*, Vol.2, N0.5 (1992)

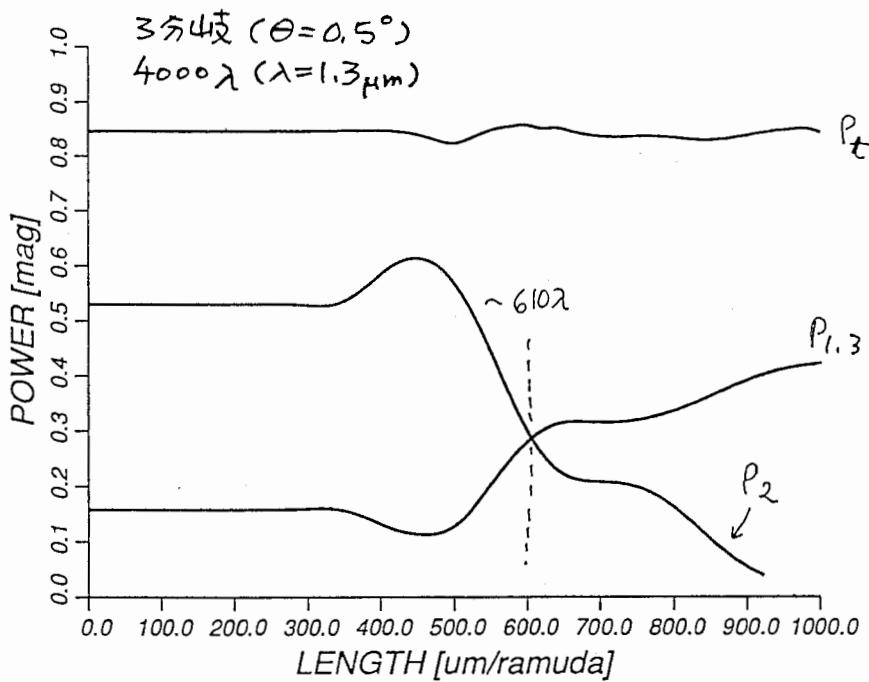


図10-a)、光エネルギー分配比の計算結果
(3分岐、 $\theta = 0.5^\circ$ 、伝搬距離=4000 λ)

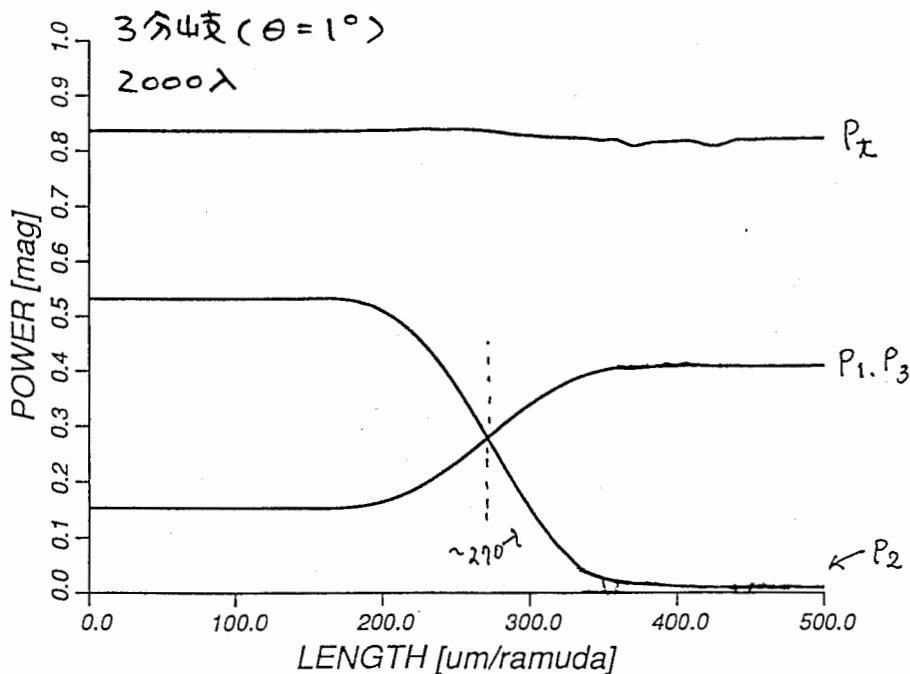


図10-b)、光エネルギー分配比の計算結果
(3分岐、 $\theta = 1.0^\circ$ 、伝搬距離=2000 λ)

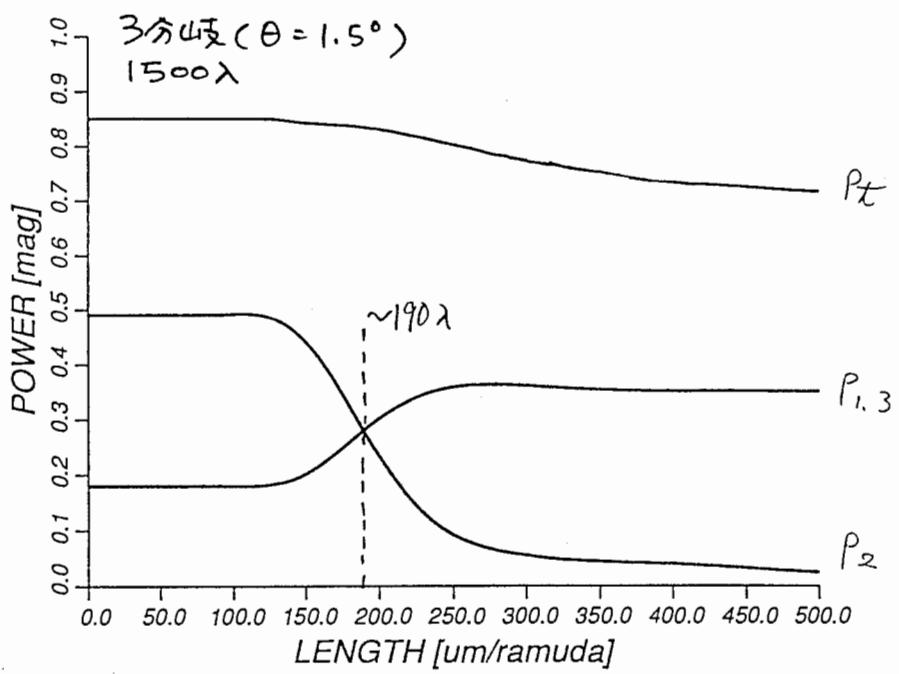


図10-c)、光エネルギー分配比の計算結果
(3分岐、 $\theta = 1.5^\circ$ 、伝搬距離=1500λ)

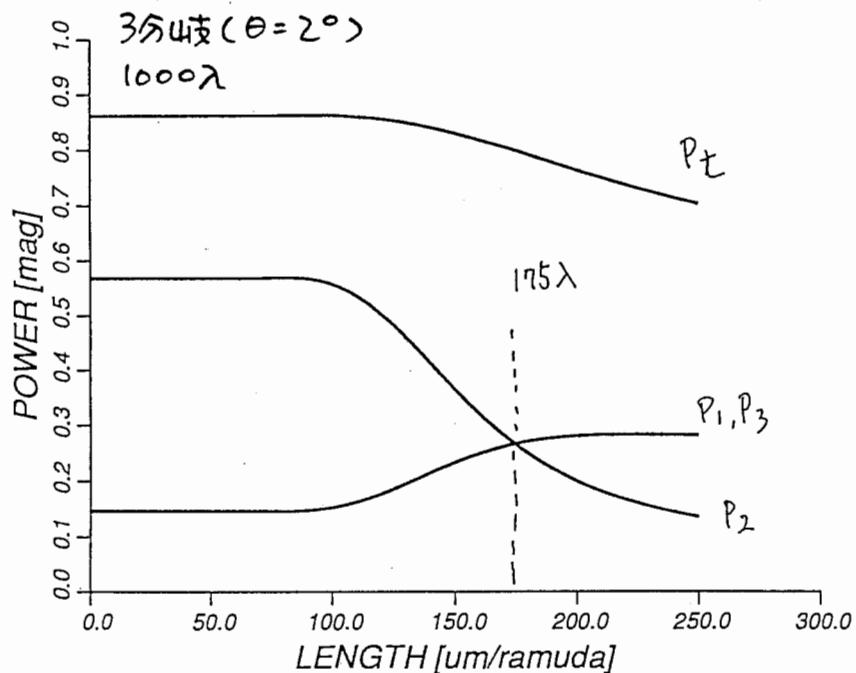


図10-d)、光エネルギー分配比の計算結果
(3分岐、 $\theta = 2.0^\circ$ 、伝搬距離=1000λ)

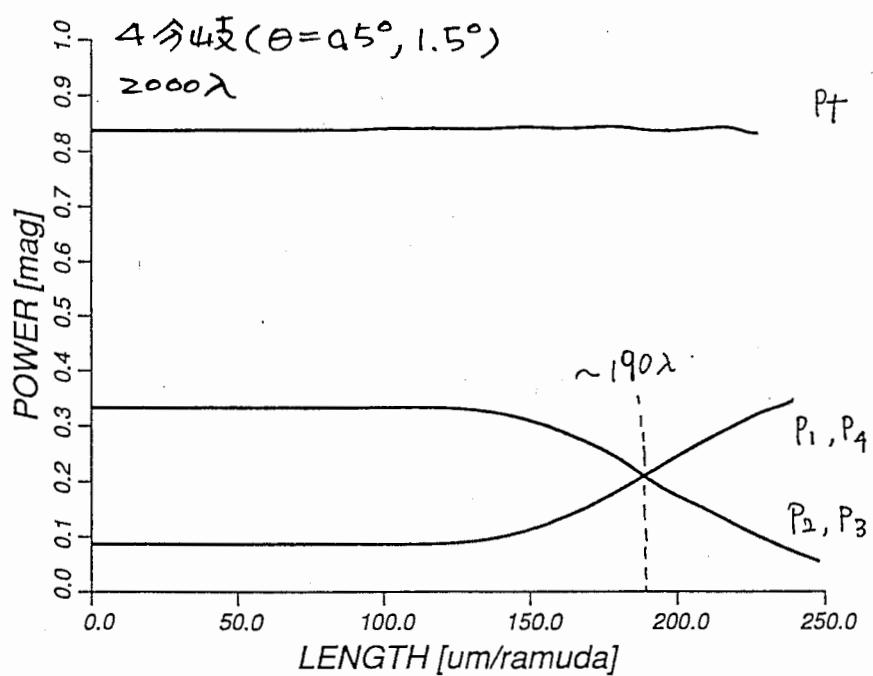


図10-e)、光エネルギー分配比の計算結果
(4分岐、 $\theta = 0.5^\circ, 1.5^\circ$ 、伝搬距離=2000 λ)

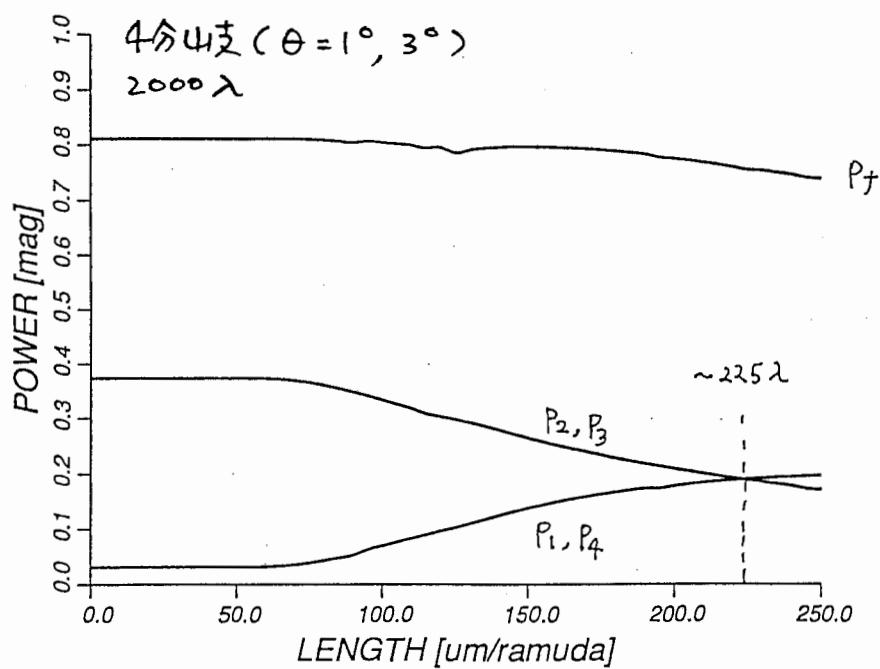
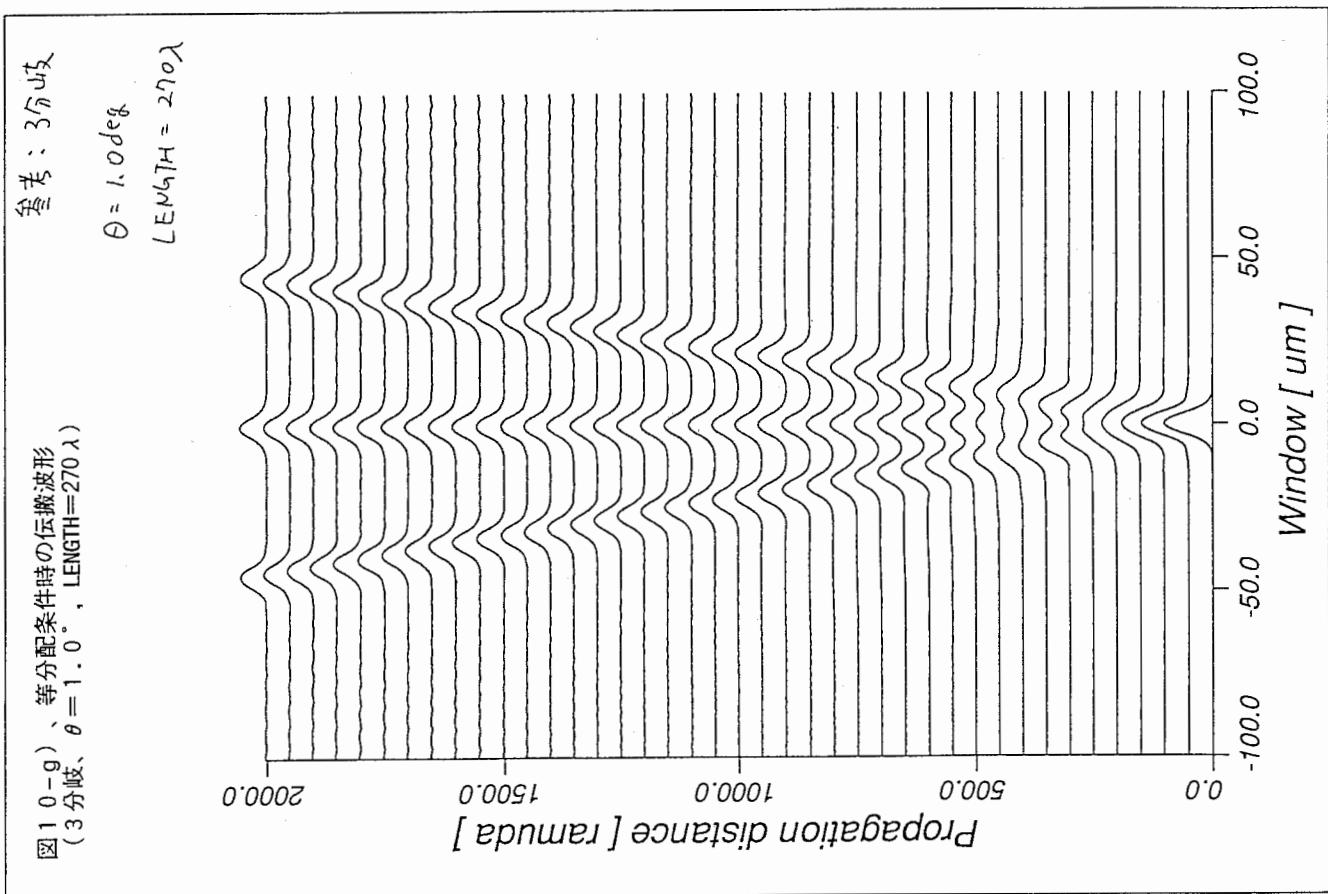
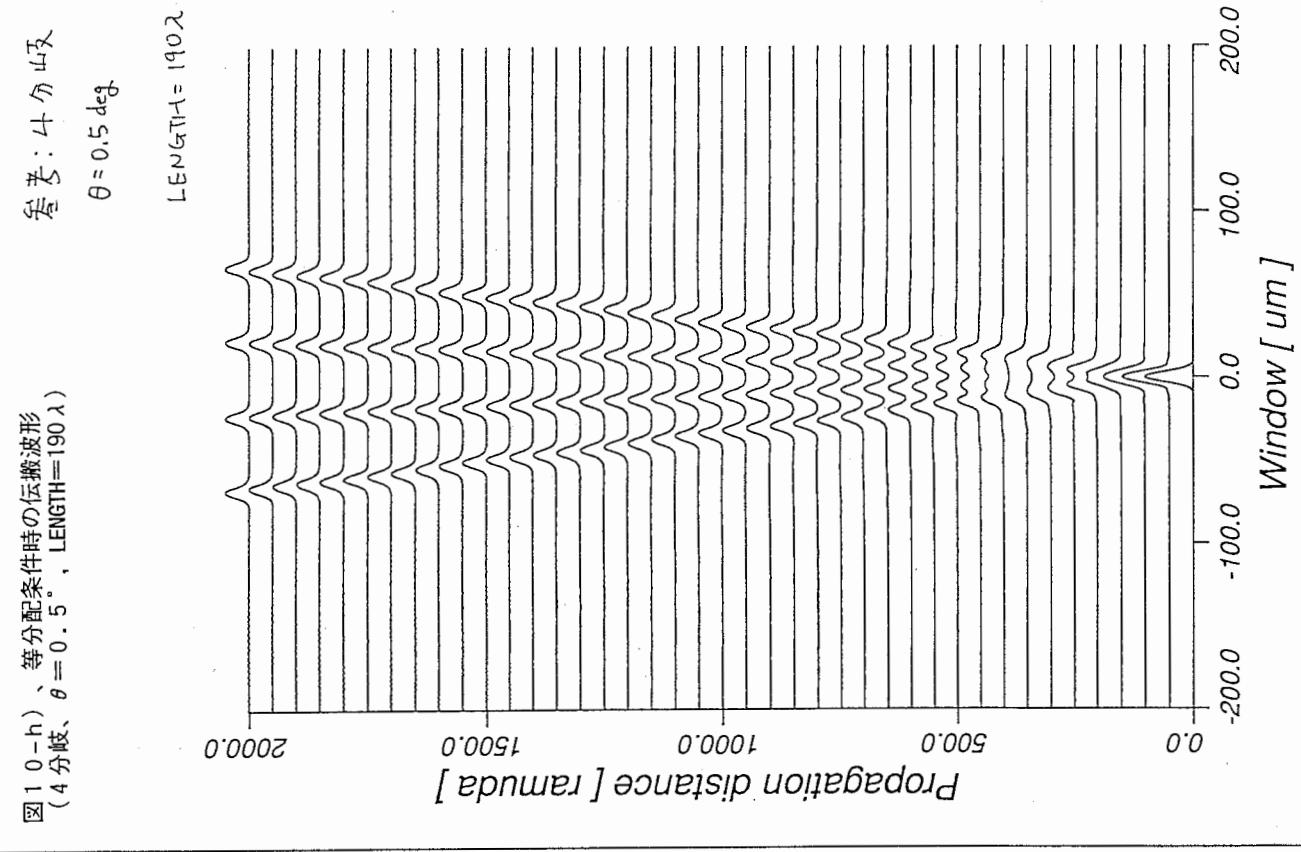


図10-f)、光エネルギー分配比の計算結果
(4分岐、 $\theta = 1.0^\circ, 3.0^\circ$ 、伝搬距離=2000 λ)



付録（プログラムリスト）

以下のプログラム構成になっている。makefile等を参考に利用して戴きたい。

makefile: make用

bpm.f: メインプログラム

init.f: 変数初期化を行う

inputdata.f: データファイル "bpm.dat" を読み込む

bpm.dat: 計算時に与えるデータ

sourcewave.f: 初期界分布（ガウス分布）

propagation.f: 空間伝搬させる

datafile.f: 計算経過をデータファイル "fort.50" に格納する

lenz.f: 各種光導波路の位相変化を計算する

auto.f: パラメータを変更して bpm.exe を実行する

bpmloss.f: データファイル "fort.50" を用いて各種光導波路の損失を計算する (fort.10 を出力する)

show2d.f, show3d.f: DISSPLAによるデータ表示プログラム (fort.50利用)

gr1_1.f: DISSPLAによるデータ表示プログラム (fort.10利用)

"makefile"

FC=fc

FFLAGS=

bpmc.exe: bpmc.o datafile.o init.o inputdata.o propagation.o lenz.o sourcrowave.o wavephase.o analysis.o
\$(FC) \$(FFLAGS) -o bpmc.exe bpmc.o datafile.o init.o inputdata.o lenz.o propagation.o sourcrowave.o wavephase.o analysis.o -lmslib

auto.exe: auto.o

\$(FC) \$(FFLAG) -o auto.exe auto.o

```

PROGRAM MAIN
C ****
C Beam propagation method
C Solver program
C 1991.07.12 - 1991.07.25
C S.Banba
C ****
IMPLICIT DOUBLE PRECISION (A-H,K-Z)
INTEGER IA,IB,IC,ID,IE,IAA,IBB,ICC
PARAMETER (IAA=1024, IBB=200, IC=20, ID=24626, ICC=5)
C ****
C
COMPLEX*16 DUM(IC),A1(IAA),A2(IAA),DATA(IBB,IAA)
DIMENSION DATAPRA(IC),N(IAA),KAI(IAA)
DIMENSION GRDAT(IAA,IBB),XDAT(IAA),YDAT(IBB),YDB(IBB),YMAG(IBB)
DIMENSION PEAK(ICC,IBB),POWER(ICC,IBB),IPOS(ICC,IBB),
#POWERDB(ICC,IBB),POWERMAG(ICC,IBB),POWERDB2(IBB),POWERMAG2(IBB)
COMMON/WORKSP/ RWKSP
REAL RWKSP(ID)
CALL IWKIN(ID)
C
CALL INIT(A1,A2,DATA,DUM,IAA,IBB,IC)
C
CALL INPUIDATA(SPOT,NO,NF,NC,RAMUDA,
#T,XSTART,XSTOP,ZSTEP,IA,IB,IE,DATAPRA,IC,ZZ,XSTEP,KO,
#BRANCH1,BRANCH2,LENGTH,DIFFUSION)
C
CALL SOURCEWAVE(SPOT,
#A1,IA,IB,IAA,IBB,XSTART,XSTEP,DATA)
CALL DFFICF(IA,A1,A2)
C
DO 1001 IBSTEP=1,IB-1
CALL PROPAGATION(ZSTEP,DUM,A2,KO,NO,XSTART,XSTOP,IA,IC)
CALL DFFICB(IA,A2,A1)
CALL DATAFILE(DATA,A1,IBSTEP,IA,IB,IC,IE,IAA,IBB,DATAPRA)
CALL LENZ(XSTART,XSTOP,XSTEP,T,NF,NO,NC,KO,ZSTEP,ZZ,A1,IA,
#BRANCH1,BRANCH2,LENGTH,DIFFUSION,N,KAI)
CALL WAVEPHASE(A1,NO,KO,ZZ,IA)
CALL DFFICF(IA,A1,A2)
CALL PROPAGATION(ZSTEP,DUM,A2,KO,NO,XSTART,XSTOP,IA,IC)
ZZ=ZZ+ZSTEP
1001 CONTINUE
C
CALL PROPAGATION(ZSTEP,DUM,A2,KO,NO,XSTART,XSTOP,IA,IC)
CALL DFFICB(IA,A2,A1)
CALL DATAFILE(DATA,A1,IBSTEP,IA,IB,IC,IE,IAA,IBB,DATAPRA)
C
CALL ANALYSIS(IAA,IBB,GRDAT,XDAT,YDAT,YDB,YMAG,PEAK,POWER,
#IPOS,POWERDB,POWERMAG,POWERDB2,POWERMAG2,IC,DATAPRA,ICC)
STOP
END

```

```
C ****
C Beam propagation method
C subroutine init.f
C S.Banba 1991.07.26
C ****
      SUBROUTINE INIT(A1,A2,DATA,DUM,IAA,IBB,IC)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      INTEGER IAA,IBB,IC
      COMPLEX*16 DUM(IC),A1(IAA),A2(IAA),DATA(IBB,IAA)
C
      DO 105 I=1,IAA
         A1(I)=(0.0,0.0)
         A2(I)=(0.0,0.0)
105   CONTINUE
      DO 106 I=1,IAA
         DO 106 J=1,IBB
            DATA(J,I)=(0.0,0.0)
106   CONTINUE
      DO 107 I=1,IC
         DUM(I)=(0.0,0.0)
107   CONTINUE
      RETURN
      END
```

```

C ****
C Beam propagation method
C subroutine inputdata.f
C S.Banba 1992.12.11
C ****
C SUBROUTINE INPUIDATA(SPOT,NO,NF,NC,RAMUDA,T,
#XSTART,XSTOP,ZSTEP,IA,IB,IE,DATAPRA,IC,ZZ,
#XSTEP,KO,BRANCH1,BRANCH2,LENGTH,DIFFUSION)
C
C IMPLICIT DOUBLE PRECISION (A-H,K-Z)
C INTEGER IA,IB,IC,IE
C DIMENSION DATAPRA(IC)
C
C OPEN(10,file='bpm.dat',status='old')
C READ(10,*) SPOT
C READ(10,*) NO,NF,NC,RAMUDA,T
C READ(10,*) XSTART,XSTOP,ZSTEP
C READ(10,*) IA,IB,IE
C READ(10,*) BRANCH1,BRANCH2,LENGTH,DIFFUSION
C
C DATAPRA(1)=XSTART
C DATAPRA(2)=XSTOP
C DATAPRA(3)=ZSTEP
C DATAPRA(5)=RAMUDA
C DATAPRA(6)=IA
C DATAPRA(7)=IB
C DATAPRA(8)=IE
C DATAPRA(4)=ZSTEP*DBLE(IB)
C DATAPRA(9)=BRANCH1
C DATAPRA(10)=BRANCH2
C DATAPRA(11)=LENGTH
C DATAPRA(12)=T
C
C ZZ=0.0
C XSTEP=(XSTOP-XSTART)/(IA-1)
C PI=3.14159265358979323846
C KO=2.0*PI/RAMUDA
C
C RETURN
C END

```

"bpm.dat"

4.319900199334370E-006		
2.20469999313354	2.20609998703003	2.20199990272522
1.300000008086499E-006	6.000000212225132E-006	
-9.999999747378752E-005	9.999999747378752E-005	1.300000008086499E-006
1024	2000	50
-1.000000000000000	1.000000000000000	70.0000000000000
4.319900199334370E-006		

```
C ****
C Beam propagation method
C subroutine sourcewave.f
C (Gauss beam)
C S.Banba 1992.12.10
C ****
C SUBROUTINE SOURCEWAVE(SPOT,A1,IA,IB,IAA,IBB,XSTART,XSTEP,DATA)
C
C IMPLICIT DOUBLE PRECISION (A-H,K-Z)
C COMPLEX*16 A1(IA),DATA(IBB,IAA)
C INTEGER I
C PI=3.14159265358979323846
C
C X=XSTART ! 50.0E-6
C DO 101 I=1,IA
C     EYX=DEXP(-1.0*(X/SPOT)**2) ! 0.5
C     A1(I)=EYX
C     DATA(1,I)=EYX
C     X=X+XSTEP
C 101 CONTINUE
C RETURN
C END
```

```

C ****
C Beam propagation method
C subroutine propagation.f
C S.Banba 1991.07.26
C ****
      SUBROUTINE PROPAGATION(ZSTEP,DUM,A2,KO,NO,XSTART,XSTOP,IA,IC)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      COMPLEX*16 DUM(IC),A2(IA)
      INTEGER I
      PI=3.14159265358979323846
C
      ZD=ZSTEP/2.0
      L=XSTOP-XSTART
      DO 201 I=1,IA
         IF((I.GT.0).AND.(I.LE.IA/2)) THEN
            KXN=2.0*PI*DBLE(I-1)/L
         ELSE
            KXN=2.0*PI*DBLE(I-IA-1)/L
         END IF
         DUMP=-KXN**2+(KO*NO)**2
C
         IF(DUMP.GE.0.0) THEN
            DUM(5)=-KXN**2/(DSQRT(DUMP)+KO*NO)
            DUM(6)=CDEXP((-ZD*DUM(5))*(0.D0,1.D0))*A2(I)
         ELSE
            DUM(6)=(0.0,0.0)
         END IF
         A2(I)=DUM(6)
201  CONTINUE
      RETURN
      END

```

```

C ****
C Beam propagation method
C subroutine datafile.f
C S.Banba 1991.07.26
C ****
      SUBROUTINE DATAFILE(DATA,A1,IBSTEP,IA,IB,IC,IE,IAA,IBB,DATAPRA)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      COMPLEX*16 A1(IAA),DATA(IBB,IAA)
      DIMENSION DATAPRA(IC)
      INTEGER JA,IBSTEP
      OPEN (50, FILE='fort.50', STATUS="UNKNOWN")
C
      DO 701 JA=1,IA
         A1(JA)=A1(JA)/DBLE(IA)
701   CONTINUE
      IF(MOD(IBSTEP,IE).EQ.0) THEN
         JE=INT(IBSTEP/IE)+1
         WRITE(6,*) IBSTEP,JE
         DO 702 JA=1,IA
            DATA(JE,JA)=A1(JA)
702   CONTINUE
      END IF
C
      IF(IBSTEP.NE.IB) THEN
         WRITE(6,*) IBSTEP
      ELSE
         DO 801 I=1,12
            WRITE(50,*) DATAPRA(I)
801   CONTINUE
         DO 802 JBB=1,INT(IB/IE)+1
            WRITE(6,*) JBB
            DO 802 JAA=1,IA
               A3=SNGL(CDABS(DATA(JBB,JAA)))
               WRITE(50,*) A3,JBB
802   CONTINUE
         CLOSE(50)
      END IF
      RETURN
END

```

```

C ****
C Beam propagation method
C subroutine lenz.f
C S.Banba 1992.12.11
C ****
C SUBROUTINE LENZ(XSTART,XSTOP,XSTEP,T,NF,NO,NC,KO,ZSTEP,ZZ,A1,
#IA,BRANCH1,BRANCH2,LENGTH,DIFFUSION,N,KAI)
IMPLICIT DOUBLE PRECISION (A-H,K-Z)
DIMENSION N(IA),KAI(IA)
COMPLEX*16 A1(IA)
INTEGER J

MICRON=1.0E-6
ANG1=BRANCH1*3.14159265358979323846/180.0
ANG2=BRANCH2*3.14159265358979323846/180.0
C
J=0
DO 301 X=XSTART,XSTOP,XSTEP
J=J+1
C ****
C Simple waveguide ( Bent waveguide )
C
X1=0.0

C IF(ZZ.LT.500.0*ZSTEP) THEN
C   X1=-50.0E-6
C ELSE IF((ZZ.GE.500.0*ZSTEP).AND.(ZZ.LT.1500.0*ZSTEP)) THEN
C   X1=(ZZ-500.0*ZSTEP)*TAN(ANG1)-50.0E-6
C ELSE
C   X1=1000.0*ZSTEP*TAN(ANG1)-50.0E-6
C END IF

C X1=ZZ*TAN(ANG1)
C IF(ABS(X-X1).LT.(T/2.0)) THEN
C   N(J)=NF
C ELSE
C   N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
C END IF

C ****
C Directional coupler
C
X1=0.0E-6
X2=9.0E-6

C IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))) THEN
C   N(J)=NF
C ELSE
C   N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
C   & +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2)
C END IF

C ****
C ****
C 3-Branching waveguide
C
X1=ZZ*TAN(ANG1)
X2=0.0
X3=ZZ*TAN(ANG2)

C IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
C & .OR.(ABS(X-X3).LT.(T/2.0))) THEN

```

```

C      N(J)=NF
C      ELSE
C          N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
C          &           +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2)
C          &           +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2)
C      END IF

C ****
C ****
C      Symmetrical 3-branch waveguide

X1=ZZ*TAN(ANG1)
X2=0.0
X3=ZZ*TAN(ANG2)

IF(ZZ.LT.LENGTH*ZSTEP) THEN
    IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X3).LT.(T/2.0))) THEN
        N(J)=NF

        ELSE IF(X.LT.X1) THEN ! !
            N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) ! !

        ELSE IF((X.GE.X1).AND.(X.LT.X3)) THEN ! !
            N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) ! !
            &           +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) ! !

        ELSE
            N(J)=NC+(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) ! !

        END IF

    ELSE
        IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
        &           .OR.(ABS(X-X3).LT.(T/2.0))) THEN
            N(J)=NF

            ELSE IF(X.LT.X1) THEN ! !
                N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) ! !

            ELSE IF((X.GE.X1).AND.(X.LT.X2)) THEN ! !
                N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) ! !
                &           +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2) ! !

            ELSE IF((X.GE.X2).AND.(X.LT.X3)) THEN ! !
                N(J)=NC+(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2) ! !
                &           +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) ! !

            ELSE
                N(J)=NC+(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) ! !

            END IF

        END IF

    ****
    KAI(J)=(N(J)-NO)*KO
    A1(J)=CDEXP(-ZSTEP*KAI(J)*(0.D0,1.D0))*A1(J)
C
301  CONTINUE
RETURN
END

```

```

C ****
C     Beam propagation method
C     subroutine lenz.f
C     S.Banba 1992.12.11
C ****
      SUBROUTINE LENZ(XSTART,XSTOP,XSTEP,T,NF,NO,NC,KO,ZSTEP,ZZ,A1,
#IA,BRANCH1,BRANCH2,LENGTH,DIFFUSION,N,KAI)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      DIMENSION N(IA),KAI(IA)
      COMPLEX*16 A1(IA)
      INTEGER J

      MICRON=1.0E-6
      ANG1=BRANCH1*3.14159265358979323846/180.0
      ANG2=BRANCH2*3.14159265358979323846/180.0
C
      J=0
      DO 301 X=XSTART,XSTOP,XSTEP
          J=J+1
C ****
C     Directional coupler
C
      X1=0.0E-6
      X2=9.0E-6

      IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))) THEN
          N(J)=NF
      ELSE
          N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
          & +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2)
      END IF

C ****
C ****
C     Symmetrical 4-branch waveguide
C
      X1=-ZZ*TAN(ANG1)
      X2=-ZZ*TAN(ANG2)
      X3=ZZ*TAN(ANG2)
      X4=ZZ*TAN(ANG1)

      IF(ZZ.LT.LENGTH*ZSTEP) THEN
          IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X4).LT.(T/2.0))) THEN
              N(J)=NF
          ELSE IF(X.LT.X1) THEN ! !
              N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !
          ELSE IF((X.GE.X1).AND.(X.LT.X4)) THEN ! !
              N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !
              & +(NF-NC)*DEXP(-(X-X4)**2/DIFFUSION**2) !
          ELSE
              N(J)=NC+(NF-NC)*DEXP(-(X-X4)**2/DIFFUSION**2) !
          END IF
      ELSE
          IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
          & .OR.(ABS(X-X3).LT.(T/2.0))
          & .OR.(ABS(X-X4).LT.(T/2.0))) THEN

```

```

N(J)=NF

ELSE IF(X.LT.X1) THEN !  

  N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !  

ELSE IF((X.GE.X1).AND.(X.LT.X2)) THEN !  

  N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !  

&   +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2) !  

ELSE IF((X.GE.X2).AND.(X.LT.X3)) THEN !  

  N(J)=NC+(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2) !  

&   +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) !  

ELSE IF((X.GE.X3).AND.(X.LT.X4)) THEN !  

  N(J)=NC+(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) !  

&   +(NF-NC)*DEXP(-(X-X4)**2/DIFFUSION**2) !  

ELSE !  

  N(J)=NC+(NF-NC)*DEXP(-(X-X4)**2/DIFFUSION**2) !  

END IF

```

END IF

C ****

```

KAI(J)=(N(J)-NO)*KO
A1(J)=CDEXP(-ZSTEP*KAI(J)*(0.D0,1.D0))*A1(J)

```

C
301 CONTINUE
RETURN
END

```

C ****
C     Beam propagation method
C     subroutine lenz.f
C     S.Banba 1992.12.11
C ****
      SUBROUTINE LENZ(XSTART,XSTOP,XSTEP,T,NF,NO,NC,KO,ZSTEP,ZZ,A1,
#IA,BRANCH1,BRANCH2,LENGTH,DIFFUSION,N,KAI)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      DIMENSION N(IA),KAI(IA)
      COMPLEX*16 A1(IA)
      INTEGER J

      MICRON=1.0E-6
      ANG1=BRANCH1*3.14159265358979323846/180.0
      ANG2=BRANCH2*3.14159265358979323846/180.0
C
      J=0
      DO 301 X=XSTART,XSTOP,XSTEP
         J=J+1
C ****
C     Simple waveguide ( Bent waveguide )
C
      X1=0.0 ! straight waveguide

      IF(ZZ.LT.500.0*ZSTEP) THEN
         X1=0.0
      ELSE IF((ZZ.GE.500.0*ZSTEP).AND.(ZZ.LT.1500.0*ZSTEP)) THEN
         X1=(ZZ-500.0*ZSTEP)*TAN(ANG1)
      ELSE
         X1=1000.0*ZSTEP*TAN(ANG1)
      END IF

      X1=ZZ*TAN(ANG1)

      IF(ABS(X-X1).LT.(T/2.0)) THEN
         N(J)=NF
      ELSE
         N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
      END IF

C ****
C     Directional coupler
C
      X1=0.0E-6
      X2=9.0E-6

C
      IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))) THEN
         N(J)=NF
      ELSE
         N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
         & +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2)
      END IF

C ****
C ****
C     3-Branching waveguide
C
      X1=ZZ*TAN(ANG1)
      X2=0.0
      X3=ZZ*TAN(ANG2)

C
      IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0)))

```

```

C & .OR. (ABS(X-X3).LT.(T/2.0))) THEN
C N(J)=NF
C ELSE
C   N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2)
C   & +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2)
C   & +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2)
C END IF

C ****
C ****
C      Symmetrical 3-branch waveguide
C
C X1=ZZ*TAN(ANG1)
C X2=0.0
C X3=ZZ*TAN(ANG2)
C
C IF(ZZ.LT.LENGTH*ZSTEP) THEN
C   IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X3).LT.(T/2.0))) THEN
C     N(J)=NF
C
C   ELSE IF(X.LT.X1) THEN ! !
C     N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !
C
C   ELSE IF((X.GE.X1).AND.(X.LT.X3)) THEN ! !
C     N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !
C     & +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) !
C
C   ELSE
C     N(J)=NC+(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) !
C
C   END IF
C
C ELSE
C   IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
C   & .OR.(ABS(X-X3).LT.(T/2.0))) THEN
C     N(J)=NF
C
C   ELSE IF(X.LT.X1) THEN ! !
C     N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !
C
C   ELSE IF((X.GE.X1).AND.(X.LT.X2)) THEN ! !
C     N(J)=NC+(NF-NC)*DEXP(-(X-X1)**2/DIFFUSION**2) !
C     & +(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2) !
C
C   ELSE IF((X.GE.X2).AND.(X.LT.X3)) THEN ! !
C     N(J)=NC+(NF-NC)*DEXP(-(X-X2)**2/DIFFUSION**2) !
C     & +(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) !
C
C   ELSE
C     N(J)=NC+(NF-NC)*DEXP(-(X-X3)**2/DIFFUSION**2) !
C
C   END IF
C
C END IF
C ****
C
C KAI(J)=(N(J)-NO)*KO
C A1(J)=CDEXP(-ZSTEP*KAI(J)*(0.D0,1.D0))*A1(J)

```

C
 301 CONTINUE
 RETURN
 END

```
C ****
C Beam propagation metod
C subroutine wavephase.f
C S.Banba 1991.07.26
C ****
SUBROUTINE WAVEPHASE(A1,NO,KO,ZZ,IA)
IMPLICIT DOUBLE PRECISION (A-H,K-Z)
COMPLEX*16 A1(IA)
INTEGER J
DO 501 J=1,IA
      A1(J)=A1(J)*CDEXP(-NO*KO*ZZ*(0.D0,1.D0))
501 CONTINUE
RETURN
END
```

```

C ****
C Beam Propagation Method
C Auto.exe auto.f
C S.Banba 1991.10.18
C ****
C IMPLICIT DOUBLE PRECISION (A-H,K-Z)
C SPOT=4.3199E-6
C NO=2.2047 ! 2.2047
C NF=2.2061
C NC=2.2020
C RAMUDA=1.3E-6
C T=6.0E-6
C XSTART=-100.0E-6
C XSTOP=100.0E-6
C ZSTEP=1.3E-6
C IA=1024
C IB=2000
C IE=50
C BRANCH1=-1.00
C BRANCH2=1.00
C DIFFUSION=4.3199E-6

C
C DO 100 LENGTH=0.0, 500.0, 10.0
C
C OPEN(10,FILE='bpm.dat',STATUS='UNKNOWN')
C WRITE(10,*) SPOT
C WRITE(10,*) NO,NF,NC,RAMUDA,T
C WRITE(10,*) XSTART,XSTOP,ZSTEP
C WRITE(10,*) IA,IB,IE
C WRITE(10,*) BRANCH1,BRANCH2,LENGTH,DIFFUSION
C CLOSE(10)
C
C CALL SYSTEM( 'bpmc.exe' )
100 CONTINUE
C CALL SYSTEM( 'pln fort.60' )
C CALL SYSTEM( 'cp fort.50 ../../' )
C STOP
C END

```

```

C ****
C Beam propagation metod
C BPM-loss bpmloss.f
C S.Banba 1991.07.26
C ****
C SUBROUTINE ANALYSIS(IAA,IBB,GRDAT,XDAT,YDAT,YDB,YMAG,PEAK,POWER,
#IPOS,POWERDB,POWERMAG,POWERDB2,POWERMAG2,IC,DATAPRA,ICC)
C
C IMPLICIT INTEGER(I-K),DOUBLE PRECISION(A-H,L-Z)
IMPLICIT INTEGER(I-K),REAL(A-H,L-Z)
C PARAMETER (IAA=1024,IBB=200,ICC=5,IC=20)
DIMENSION GRDAT(IAA,IBB),XDAT(IAA),YDAT(IBB),YDB(IBB),YMAG(IBB)
DIMENSION PEAK(ICC,IBB),POWER(ICC,IBB),IPOS(ICC,IBB),
#POWERDB(ICC,IBB),POWERMAG(ICC,IBB),POWERDB2(IBB),POWERMAG2(IBB)
DIMENSION DATAPRA(IC)
OPEN (50, FILE='fort.50', STATUS="OLD")
DO K=1,12
  READ(50,*) DATAPRA(K)
ENDDO
C
XMIN=DATAPRA(1)
XMAX=DATAPRA(2)
YSTEP=DATAPRA(3)
RAMUDA=DATAPRA(5)
IA=INT(DATAPRA(6))
IB=INT(DATAPRA(7))
IE=INT(DATAPRA(8))
IBE=INT(DATAPRA(7)/DATAPRA(8))+1
YMIN=0.0
YMAX=DATAPRA(4)/RAMUDA
BRANCH1=DATAPRA(9)
BRANCH2=DATAPRA(10)
LENGTH=DATAPRA(11)
T=DATAPRA(12)
ANG1=BRANCH1*3.14159/180.0
ANG2=BRANCH2*3.14159/180.0
MICRON=1.0E-6
C
X=XMIN
XSTEP=(XMAX-XMIN)/IA
DO I= 1,IA
  XDAT(I)=X
  X=X+XSTEP
ENDDO
C
DO J= 1,IBE
  YDAT(J)=0.0
  ZZ=(J-1)*IE*YSTEP
  DO I= 1,IA
    READ(50,*) GRDAT(I,J)
  ENDDO
C ****
C Simple waveguide
C X1=0.0
C X1=ZZ*TAN(ANG1)

  IF(ZZ.LT.500.0*ZSTEP) THEN
    X1=0.0
  ELSE IF((ZZ.GE.500.0*ZSTEP).AND.(ZZ.LT.1500.0*ZSTEP)) THEN
    X1=(ZZ-500.0*ZSTEP)*TAN(ANG1)
  ELSE
    X1=1000.0*ZSTEP*TAN(ANG1)
  END IF

```

```

IF(ABS(XDAT(I)-X1).LT.(T/2.0)) THEN
C ****
C      Directional coupler
C      X1=0.0E-6
C      X2=9.0E-6
C      IF((ABS(XDAT(I)-X1).LT.(T/2.0)).OR.
C &      (ABS(XDAT(I)-X2).LT.(T/2.0))) THEN
C ****
C      3-Branching waveguide
C      X1=ZZ*TAN(ANG1)
C      X2=0.0
C      X3=ZZ*TAN(ANG2)
C      IF((ABS(XDAT(I)-X1).LT.(T/2.0)).OR.(ABS(XDAT(I)-X2).LT.(T/2.0))
C &      .OR.(ABS(XDAT(I)-X3).LT.(T/2.0))) THEN
C ****
C ****
C      Symmetrical 3-branch waveguide
C      X1=ZZ*TAN(ANG1)
C      X2=0.0
C      X3=ZZ*TAN(ANG2)
C      IF(ZZ.LT.LENGTH*ZSTEP) THEN
C          IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X3).LT.(T/2.0))) THEN
C              END IF
C          ELSE
C              IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
C &              .OR.(ABS(X-X3).LT.(T/2.0))) THEN
C                  END IF
C ****
C      YDAT(J)=GRDAT(I,J)*GRDAT(I,J)*XSTEP+YDAT(J)
END IF
ENDDO
IF(YDAT(1).LT.2.8393570E-06) THEN
    YDAT(1)=2.8393570E-06
END IF
YMAG(J)=ABS(YDAT(J)/YDAT(1))
YDB(J)=10.0*LOG10(YMAG(J))
WRITIE(6,*) J,ZZ,YDAT(J),YDB(J),YMAG(J)
ENDDO
C
IWIDTH=INT(0.5*T*(IA-1)/(XMAX-XMIN))+1
DO J= 2,IWE
    K=0
    DO I=2,IA-1,3
        GR1=(GRDAT(I-2,J)+GRDAT(I-3,J)+GRDAT(I-4,J))/3.0
        GR2=(GRDAT(I+2,J)+GRDAT(I+3,J)+GRDAT(I+4,J))/3.0
        GR3=(GRDAT(I,J)+GRDAT(I-1,J)+GRDAT(I+1,J))/3.0
        GR4=0.10
        IF((GR3.GT.GR1).AND.(GR3.GT.GR2))

```

```

#      .AND. (GRDAT(I,J).GT.GR4)) THEN
      K=K+1
      PEAK(K,J)=GRDAT(I,J)
      IPOS(K,J)=I
   END IF
   KK=K
ENDDO
ENDDO

C
DO J= 2,IBE
  DO K= 1,KK
    POWER(K,J)=0.0
    DO I= 1,IA
      IP1=IPOS(K,J)-IWIDH
      IP2=IPOS(K,J)+IWIDH
      IF(I.GT.IP1.AND.I.LT.IP2) THEN
        POWER(K,J)=GRDAT(I,J)*GRDAT(I,J)*XSTEP+POWER(K,J)
      END IF
    ENDDO
    POWERMAG(K,J)=ABS(POWER(K,J)/YDAT(1))
    POWERDB(K,J)=10.0*LOG10(POWERMAG(K,J))
    WRITE(6,*) J,K,POWERDB(K,J),POWERMAG(K,J),IPOS(K,J)
  ENDDO
  POWERMAG2(J)=POWERMAG(1,J)+POWERMAG(2,J)+POWERMAG(3,J)
  POWERDB2(J)=10.0*LOG10(POWERMAG2(J))
  WRITE(6,*) J,K,POWERDB2(J),POWERMAG2(J)
ENDDO

C
OPEN (60, FILE='fort.60', ACCESS="APPEND")
RATIO=YMAG(IBE)/POWERMAG2(IBE)
POWERR1=POWERMAG(1,IBE)*RATIO
POWERR2=POWERMAG(2,IBE)*RATIO
POWERR3=POWERMAG(3,IBE)*RATIO

C
WRITE(60,*) BRANCH,LENGTH,IPOS(1,IBE),IPOS(2,IBE),IPOS(3,IBE)
WRITE(60,*) POWERMAG2(IBE),YMAG(IBE),
#POWERMAG(1,IBE),POWERMAG(2,IBE),POWERMAG(3,IBE)
WRITE(60,*) RATIO,POWERR1,POWERR2,POWERR3
WRITE(60,*)
WRITE(60,*)

C
CLOSE(60)
CLOSE(50)
RETURN
END

```

```

C ****
C Beam propagation metod
C BPM-loss bpmloss.f
C S.Banba 1992.12.18
C ****
C SUBROUTINE ANALYSIS(IAA,IBB,GRDAT,XDAT,YDAT,YDB,YMAG,PEAK,POWER,
#IPOS,POWERDB,POWERMAG,POWERDB2,POWERMAG2,IC,DATAPRA,ICC)
C
C IMPLICIT INTEGER(I-K),DOUBLE PRECISION(A-H,L-Z)
IMPLICIT INTEGER(I-K),REAL(A-H,L-Z)
C PARAMETER (IAA=1024,IBB=200,ICC=5,IC=20)
DIMENSION GRDAT(IAA,IBB),XDAT(IAA),YDAT(IBB),YDB(IBB),YMAG(IBB)
DIMENSION PEAK(ICC,IBB),POWER(ICC,IBB),IPOS(ICC,IBB),
#POWERDB(ICC,IBB),POWERMAG(ICC,IBB),POWERDB2(IBB),POWERMAG2(IBB)
DIMENSION DATAPRA(IC)
OPEN (50, FILE='fort.50', STATUS="OLD")
DO K=1,12
    READ(50,*) DATAPRA(K)
ENDDO
C
XMIN=DATAPRA(1)
XMAX=DATAPRA(2)
YSTEP=DATAPRA(3)
RAMUDA=DATAPRA(5)
IA=INT(DATAPRA(6))
IB=INT(DATAPRA(7))
IE=INT(DATAPRA(8))
IBE=INT(DATAPRA(7)/DATAPRA(8))+1
YMIN=0.0
YMAX=DATAPRA(4)/RAMUDA
BRANCH1=DATAPRA(9)
BRANCH2=DATAPRA(10)
LENGTH=DATAPRA(11)
T=DATAPRA(12)
ANG1=BRANCH1*3.14159/180.0
ANG2=BRANCH2*3.14159/180.0
MICRON=1.0E-6
C
X=XMIN
XSTEP=(XMAX-XMIN)/IA
DO I= 1,IA
    XDAT(I)=X
    X=X+XSTEP
ENDDO
C
DO J= 1,IBE
    YDAT(J)=0.0
    ZZ=(J-1)*IE*YSTEP
    DO I= 1,IA
        READ(50,*) GRDAT(I,J)
    ENDDO
C ****
C Symmetrical 4-branch waveguide
X1=-ZZ*TAN(ANG1)
X2=-ZZ*TAN(ANG2)
X3=ZZ*TAN(ANG2)
X4=ZZ*TAN(ANG1)

IF(ZZ.LT.LENGTH*ZSTEP) THEN
    IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X4).LT.(T/2.0))) THEN
        END IF

```

```

    ELSE
      IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
      & .OR.(ABS(X-X3).LT.(T/2.0)))
      & .OR.(ABS(X-X4).LT.(T/2.0))) THEN
    END IF

C ****
C YDAT(J)=GRDAT(I,J)*GRDAT(I,J)*XSTEP+YDAT(J)

      END IF

      ENDDO
C IF(YDAT(1).LT.2.8393570E-06) THEN
C   YDAT(1)=2.8393570E-06
C   END IF
      YMAG(J)=ABS(YDAT(J)/YDAT(1))
      YDB(J)=10.0*LOG10(YMAG(J))
C   WRITE(6,*) J,ZZ,YDAT(J),YDB(J),YMAG(J)
      ENDDO
C
      IWIDIH=INT(0.5*T*(IA-1)/(XMAX-XMIN))+1
      DO J= 2,IBE
        K=0
        DO I=2,IA-1,3
          GR1=(GRDAT(I-2,J)+GRDAT(I-3,J)+GRDAT(I-4,J))/3.0
          GR2=(GRDAT(I+2,J)+GRDAT(I+3,J)+GRDAT(I+4,J))/3.0
          GR3=(GRDAT(I,J)+GRDAT(I-1,J)+GRDAT(I+1,J))/3.0
          GR4=0.10
          IF((GR3.GT.GR1).AND.(GR3.GT.GR2)
#           .AND.(GRDAT(I,J).GT.GR4)) THEN
            K=K+1
            PEAK(K,J)=GRDAT(I,J)
            IPOS(K,J)=I
          END IF
          KK=K
        ENDDO
      ENDDO
C
      DO J= 2,IBE
        DO K= 1,KK
          POWER(K,J)=0.0
          DO I= 1,IA
            IP1=IPOS(K,J)-IWIDIH
            IP2=IPOS(K,J)+IWIDIH
            IF(I.GT.IP1.AND.I.LT.IP2) THEN
              POWER(K,J)=GRDAT(I,J)*GRDAT(I,J)*XSTEP+POWER(K,J)
            END IF
          ENDDO
          POWERMAG(K,J)=ABS(POWER(K,J)/YDAT(1))
          POWERDB(K,J)=10.0*LOG10(POWERMAG(K,J))
C         WRITE(6,*) J,K,POWERDB(K,J),POWERMAG(K,J),IPOS(K,J)
        ENDDO
        POWERMAG2(J)=POWERMAG(1,J)+POWERMAG(2,J)
#         +POWERMAG(3,J)+POWERMAG(4,J)
        POWERDB2(J)=10.0*LOG10(POWERMAG2(J))
C         WRITE(6,*) J,K,POWERDB2(J),POWERMAG2(J)
      ENDDO
C
      OPEN (60, FILE='fort.60', ACCESS="APPEND")
      RATIO=YMAG(IBE)/POWERMAG2(IBE)
      POWERR1=POWERMAG(1,IBE)*RATIO
      POWERR2=POWERMAG(2,IBE)*RATIO
      POWERR3=POWERMAG(3,IBE)*RATIO

```

POWER4=POWERMAG(4,IBE)*RATIO
C
WRITE(60,*) BRANCH1,LENGIH,IPOS(1,IBE),IPOS(2,IBE),
#IPOS(3,IBE),IPOS(4,IBE)
WRITE(60,*) POWERMAG2(IBE),YMAG(IBE),
#POWERMAG(1,IBE),POWERMAG(2,IBE),POWERMAG(3,IBE),POWERMAG(4,IBE)
WRITE(60,*) RATIO,POWERR1,POWERR2,POWERR3,POWERR4
WRITE(60,*)
WRITE(60,*)

C
CLOSE(60)
CLOSE(50)
RETURN
END

```

C ****
C     Beam propagation metod
C     BPM-loss bpmloss.f
C     S.Banba 1991.07.26
C ****
C     SUBROUTINE ANALYSIS(IAA,IBB,GRDAT,XDAT,YDAT,YDB,YMAG,PEAK,POWER,
C     #IPOS,POWERDB,POWERMAG,POWERDB2,POWERMAG2,IC,DATAPRA,ICC)
C
C     IMPLICIT INTEGER(I-K),DOUBLE PRECISION(A-H,I-Z)
C     IMPLICIT INTEGER(I-K),REAL(A-H,I-Z)
C     PARAMETER (IAA=1024,IBB=200,ICC=5,IC=20)
C     DIMENSION GRDAT(IAA,IBB),XDAT(IAA),YDAT(IBB),YDB(IBB),YMAG(IBB)
C     DIMENSION PEAK(ICC,IBB),POWER(ICC,IBB),IPOS(ICC,IBB),
C     #POWERDB(ICC,IBB),POWERMAG(ICC,IBB),POWERDB2(IBB),POWERMAG2(IBB)
C     DIMENSION DATAPRA(IC)
C     OPEN (50, FILE='fort.50', STATUS="OLD")
C     DO K=1,12
C        READ(50,*) DATAPRA(K)
C     ENDDO
C
C     XMIN=DATAPRA(1)
C     XMAX=DATAPRA(2)
C     YSTEP=DATAPRA(3)
C     RAMUDA=DATAPRA(5)
C     IA=INT(DATAPRA(6))
C     IB=INT(DATAPRA(7))
C     IE=INT(DATAPRA(8))
C     IBE=INT(DATAPRA(7)/DATAPRA(8))+1
C     YMIN=0.0
C     YMAX=DATAPRA(4)/RAMUDA
C     BRANCH1=DATAPRA(9)
C     BRANCH2=DATAPRA(10)
C     LENGTH=DATAPRA(11)
C     T=DATAPRA(12)
C     ANG1=BRANCH1*3.14159/180.0
C     ANG2=BRANCH2*3.14159/180.0
C     MICRON=1.0E-6
C
C     X=XMIN
C     XSTEP=(XMAX-XMIN)/IA
C     DO I= 1,IA
C        XDAT(I)=X
C        X=X+XSTEP
C     ENDDO
C
C     DO J= 1,IBE
C        YDAT(J)=0.0
C        ZZ=(J-1)*IE*YSTEP
C        DO I= 1,IA
C           READ(50,*) GRDAT(I,J)
C
C ****
C     Simple waveguide
C     X1=0.0
C     X1=ZZ*TAN(ANG1)
C
C     IF(ZZ.LT.500.0*ZSTEP) THEN
C        X1=-50.0E-6
C     ELSE IF((ZZ.GE.500.0*ZSTEP).AND.(ZZ.LT.1500.0*ZSTEP)) THEN
C        X1=(ZZ-500.0*ZSTEP)*TAN(ANG1)-50.0E-6
C     ELSE
C        X1=1000.0*ZSTEP*TAN(ANG1)-50.0E-6
C     END IF

```

```

C           IF(ABS(XDAT(I)-X1).LT.(T/2.0)) THEN
C ****
C     Directional coupler
C
C     X1=0.0E-6
C     X2=9.0E-6
C
C     IF((ABS(XDAT(I)-X1).LT.(T/2.0)).OR.
C     &      (ABS(XDAT(I)-X2).LT.(T/2.0))) THEN
C ****
C     3-Branching waveguide
C
C     X1=ZZ*TAN(ANG1)
C     X2=0.0
C     X3=ZZ*TAN(ANG2)
C
C     IF((ABS(XDAT(I)-X1).LT.(T/2.0)).OR.(ABS(XDAT(I)-X2).LT.(T/2.0))
C     &      .OR.(ABS(XDAT(I)-X3).LT.(T/2.0))) THEN
C ****
C ****
C     Symmetrical 3-branch waveguide
C
C     X1=ZZ*TAN(ANG1)
C     X2=0.0
C     X3=ZZ*TAN(ANG2)
C
C     IF(ZZ.LT.LENGTH*ZSTEP) THEN
C       IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X3).LT.(T/2.0))) THEN
C         END IF
C
C       ELSE
C         IF((ABS(X-X1).LT.(T/2.0)).OR.(ABS(X-X2).LT.(T/2.0))
C         &          .OR.(ABS(X-X3).LT.(T/2.0))) THEN
C           END IF
C
C ****
C
C     YDAT(J)=GRDAT(I,J)*GRDAT(I,J)*XSTEP+YDAT(J)
C
C     END IF
C
C     ENDDO
C     IF(YDAT(1).LT.2.8393570E-06) THEN
C       YDAT(1)=2.8393570E-06
C     END IF
C     YMAG(J)=ABS(YDAT(J)/YDAT(1))
C     YDB(J)=10.0*LOG10(YMAG(J))
C     WRITE(6,*) J,ZZ,YDAT(J),YDB(J),YMAG(J)
C   ENDDO
C
C   IWIDIH=INT(0.5*T*(IA-1)/(XMAX-XMIN))+1
C   DO J= 2,IBE
C     K=0
C     DO I=2,IA-1,3
C       GR1=(GRDAT(I-2,J)+GRDAT(I-3,J)+GRDAT(I-4,J))/3.0
C       GR2=(GRDAT(I+2,J)+GRDAT(I+3,J)+GRDAT(I+4,J))/3.0
C       GR3=(GRDAT(I,J)+GRDAT(I-1,J)+GRDAT(I+1,J))/3.0
C       GR4=0.10
C       IF((GR3.GT.GR1).AND.(GR3.GT.GR2))

```

```

#      .AND. (GRDAT(I,J).GT.GR4)) THEN
      K=K+1
      PEAK(K,J)=GRDAT(I,J)
      IPOS(K,J)=I
   END IF
   KK=K
ENDDO
ENDDO

C
DO J= 2,IBE
  DO K= 1,KK
    POWER(K,J)=0.0
    DO I= 1,IA
      IP1=IPOS(K,J)-IWIDIH
      IP2=IPOS(K,J)+IWIDIH
      IF(I.GT.IP1.AND.I.LT.IP2) THEN
        POWER(K,J)=GRDAT(I,J)*GRDAT(I,J)*XSTEP+POWER(K,J)
      END IF
    ENDDO
    POWERMAG(K,J)=ABS(POWER(K,J)/YDAT(1))
    POWERDB(K,J)=10.0*LOG10(POWERMAG(K,J))
    WRITE(6,*) J,K,POWERDB(K,J),POWERMAG(K,J),IPOS(K,J)
  ENDDO
  POWERMAG2(J)=POWERMAG(1,J)+POWERMAG(2,J)+POWERMAG(3,J)
  POWERDB2(J)=10.0*LOG10(POWERMAG2(J))
  WRITE(6,*) J,K,POWERDB2(J),POWERMAG2(J)
ENDDO

C
OPEN (60, FILE='fort.60', ACCESS="APPEND")
RATIO=YMAG(IBE)/POWERMAG2(IBE)
POWERR1=POWERMAG(1,IBE)*RATIO
POWERR2=POWERMAG(2,IBE)*RATIO
POWERR3=POWERMAG(3,IBE)*RATIO

C
WRITE(60,*) BRANCH,LENGTH,IPOS(1,IBE),IPOS(2,IBE),IPOS(3,IBE)
WRITE(60,*) POWERMAG2(IBE),YMAG(IBE),
#POWERMAG(1,IBE),POWERMAG(2,IBE),POWERMAG(3,IBE)
WRITE(60,*) RATIO,POWERR1,POWERR2,POWERR3
WRITE(60,*) 
WRITE(60,*)

C
CLOSE(60)
CLOSE(50)
RETURN
END

```

```

PROGRAM kazukochan2
C ****
C Beam propagation method
C BPM 2D-profiler show2d.f
C S.Banba 1991.07.26
C ****
IMPLICIT INTEGER(I-N), REAL(A-H,O-Z)
PARAMETER (IAA=1024,IBB=200) ! maximum number of data points
REAL GRDAT(IAA,IBB),XDAT(IAA),YDAT(IAA,IBB)
DIMENSION DATAPRA(20)

C * Read graph parameters
C OPEN (20, FILE='graph2d.dat', STATUS="OLD")
C CLOSE(20)

C * Read parameters (BPM profile)
OPEN (50, FILE='/usr2/convex/home/banba/bpm_atr/fort.50'
#, STATUS="OLD")

DO K=1,12
    READ(50,*) DATAPRA(K)
ENDDO
XMIN=DATAPRA(1)*1.E6
XMAX=DATAPRA(2)*1.E6
YSTEP=DATAPRA(3)
RAMUDA=DATAPRA(5)
IA=INT(DATAPRA(6))
IB=INT(DATAPRA(7))
IE=INT(DATAPRA(8))
IBE=INT(DATAPRA(7)/DATAPRA(8))+1
YMIN=0.0
YMAX=DATAPRA(4)/RAMUDA
C
X=XMIN
XSTEP=(XMAX-XMIN)/IA
DO I= 1,IA
    XDAT(I)=X
    X=X+XSTEP
ENDDO
C
DO J= 1,IBE
    DO I= 1,IA
        READ(50,*) GRDAT(I,J)
        YDAT(I,J)=GRDAT(I,J)*100.0+(J-1)*FLOAT(IE)*YSTEP/RAMUDA ! 200
    ENDDO
    WRITE(6,*) J,YDAT(1,J)
ENDDO
CLOSE(50)

C *** Draw a graph
CALL DEVIATESUN(0)
CALL PAGE (8.0, 12.0)
CALL AREA2D(6.0, 9.0)
C CALL FRAME

C * Axes etc.
XINC= (XMAX-XMIN)/4.
YINC= (YMAX-YMIN)/4.
CALL HEIGHT(0.25)
CALL SIMPLX
CALL XNAME('Window [ um ]',100)
CALL YNAME('Propagation distance [ ramuda ]',100)
CALL GRAF( XMIN,XINC,XMAX, YMIN,YINC,YMAX+100.)

```

```
C *** Draw a curve
DO J=0,IBE
  CALL CURVE(XDAT,YDAT(I,J),IA,0)
ENDDO
CALL ENDPL(0)
CALL DONEPL
STOP
END
```

```

PROGRAM kazukochan
C ****
C Beam propagation method
C BPM 3D-profiler show3d.f
C S.Banba 1991.07.26
C ****
IMPLICIT INTEGER(I-N), REAL(A-H,O-Z)
PARAMETER (IAA=1024, IBB=200) ! maximum number of data points
REAL GRDAT(IAA,IBB)
DIMENSION DATAPRA(20)
C * Read graph parameters
OPEN (20, FILE='graph3d.dat', STATUS="OLD")
READ (20,*) WBOXLX,WBOXLY,WBOXLZ
READ (20,*) VPHI,VIHETA,VRADI
READ (20,*) ZMIN,ZMAX
CLOSE(20)

C * Read parameters (BPM profile)
OPEN (50, FILE='fort.50', STATUS="OLD")
OPEN (50, FILE='/usr2/convex/home/banba/bpm_atr/fort.50'
#, STATUS="OLD")
DO K=1,12
    READ(50,*) DATAPRA(K)
ENDDO
XMIN=DATAPRA(1)*1.E6
XMAX=DATAPRA(2)*1.E6
YSTEP=DATAPRA(3)
RAMUDA=DATAPRA(5)
IA=INT(DATAPRA(6))
IB=INT(DATAPRA(7))
IE=INT(DATAPRA(8))
IBE=INT(DATAPRA(7)/DATAPRA(8))+1
YMIN=0.0
YMAX=DATAPRA(4)/RAMUDA
C
DO J= 1,IBE
    DO I= 1,IA
        READ(50,*) GRDAT(I,J)
    ENDDO
ENDDO
CLOSE(50)

C *** Draw a graph
CALL DEVIATESUN(0)
CALL PAGE (11., 9.0)
CALL AREA2D(9.0, 6.0)
CALL HEIGHT(0.2)
CALL SIMPLX
CALL FRAME

C ** Text
CALL VOLM3D(WBOXLX,WBOXLY,WBOXLZ)
CALL VUANGL(VPHI,VIHETA,VRADI)

C * Axes etc.
XINC= (XMAX-XMIN)/4.
YINC= (YMAX-YMIN)/4.
ZINC= (ZMAX-ZMIN)/2.
CALL X3NAME('Window [ um ]',100)
CALL Y3NAME('Propagation distance [ um ]',100)
CALL Z3NAME('Field amplitude',15)
CALL GRAF3D( XMIN,XINC,XMAX, YMIN,YINC,YMAX, ZMIN,ZINC,ZMAX )
C CALL BOX3D

```

```
C *** Draw a surface
CALL SURVIS('TOP')
CALL SURSIZE(XMIN,XMAX,YMIN,YMAX)
CALL SURMAT(GRDAT,IA-1,IA,1,IBE,0)
CALL ENDPL(0)
CALL DONEPL
STOP
END
```

"graph3d.dat"

20 30 5
-75 30 100
0.0 1.0

```

PROGRAM kazukochan4
C ****
C 2D-profiler gr1_1.f
C S.Banba 1992.12.16
C ****
      IMPLICIT INTEGER(I-N), REAL(A-H,O-Z)
      PARAMETER (IA=51,IB=14) ! maximum number of data points
      DIMENSION DAT(IA,IB), XDAT(IA)
      CHARACTER LXNAME*40, LYNAME*40

C * Read parameters
      OPEN (10, FILE='fort.10', STATUS="OLD")

      READ(10,*) XMIN,XMAX,XINC
      READ(10,*) YMIN,YMAX,YINC
      READ(10,101) LXNAME
      READ(10,101) LYNAME
      READ(10,*) XAREA,YAREA,HEI,THK
101    FORMAT(A40)

      READ(10,*) IAE,IBE
      CLOSE(10)
C
      OPEN (60, FILE='/usr2/convex/home/banba/bpm_atr/fort.60'
#, STATUS="OLD")

      DO I= 1,IAE
C       DO J= 1,IBE
          READ(60,*) DAT(I,1),DAT(I,2),DAT(I,3),DAT(I,4),DAT(I,5)
          READ(60,*) DAT(I,6),DAT(I,7),DAT(I,8),DAT(I,9),DAT(I,10)
          READ(60,*) DAT(I,11),DAT(I,12),DAT(I,13),DAT(I,14)
          XDAT(I)=DAT(I,2)
          WRITE(6,*) I,J,XDAT(I),DAT(I,6),DAT(I,8),DAT(I,9),DAT(I,10)

      OPEN (70, FILE='fort.70', STATUS="APPEND")
      WRITE(70,*) XDAT(I),DAT(I,6),DAT(I,8),DAT(I,9),DAT(I,10)
701    FORMAT(5F4.4)

C       ENDDO
C       ENDDO
      CLOSE(70)
      CLOSE(60)

C *** Draw a graph
      CALL DEVITAKESUN(0)
      CALL NOBRDR
      CALL PAGE (12.0, 12.0)
      CALL AREA2D (XAREA,YAREA)
C       CALL FRAME

C * Axes etc.
      CALL HEIGHT(HEI)
      CALL SIMPLX
      CALL XNAME(LXNAME,100)
      CALL YNAME(LYNAME,100)
      CALL GRAF(XMIN,XINC,XMAX,YMIN,YINC,YMAX)

C *** Draw a curve
      CALL SPLINE
      CALL THKCRV(THK)
C       DO I=1,ICE
          CALL CURVE(XDAT,DAT(I,5),IAE,0) ! DAT(I,6)
          CALL CURVE(XDAT,DAT(I,7),IAE,0) ! DAT(I,8)

```

```
CALL CURVE(XDAT,DAT(I,8),IAE,0) ! DAT(I,9)
```

```
C     CALL CURVE(XDAT,YDAT,IBE,0)
```

```
C ENDDO
```

```
CALL ENDPL(0)
```

```
CALL DONEPL
```

```
STOP
```

```
END
```