

巻末資料 A 引き戻し・応答スペクトル法の計算プログラム

巻末CDには、本資料5章に示した引き戻し・応答スペクトル法の計算プログラム [decrespec] を収録している。ただし、収録しているプログラムは5章の解析に使用したものと同一ではなく、1次モードに限定した計算を行うように機能を制限したものである。以下、このプログラムの使用方法等について説明する。

なお、本プログラムの作成にあたって使用した言語はFortran90であり、使用したコンパイラはGfortran (開発元：GNU プロジェクト、windows-20140629 版) である。

A1 使用許諾条件

本資料のプログラムまたはソースコード (以下、「本著作物」という。) は、以下の事項に同意の上で利用することができる。

1. 利用者は、本著作物を利用して得られた計算結果を、著作権者への断りなく公表することができる。ただし、その場合は本著作物を利用した旨を表示しなければならない。
2. 利用者は、著作権者への断りなく、本著作物を第三者に配布することができる。
3. 利用者は、著作権者への断りなく本著作物を改変し、二次著作物を作成することができる。ただし、二次著作物を公表・配布する場合は、原著物である本著作物の著作権が国立研究開発法人土木研究所にある旨を表示しなければならない。
4. 利用者は、著作権者への断りなく、本著作物または二次著作物を営利目的で利用することができる。ただし、その場合は原著物である本著作物の著作権が国立研究開発法人土木研究所にある旨を表示しなければならない。
5. 本著作物または二次著作物の利用によって生じるあらゆる損害に対して、著作権者は一切の責任を負わない。

A2 プログラムの実行の方法

実行ファイルの名称は [decrespec.exe] であり、その保存場所に関する指定は特にない。実行の手順は次のとおりである。

- (1) 実行ファイルと同じディレクトリに下記3種類の入力データファイルを保存する。
 - 1) layer.csv
 - 2) S_AG.csv
 - 3) depth.csv
- (2) 実行ファイル [decrespec.exe] を起動することで、計算が実行される。
- (3) 入力データファイルに不適切な点がある場合はエラーストップする。問題なく計算が完了した場合は、計算結果として以下の出力データファイルが生成される。
 - 4) mode.csv
 - 5) response.csv

A3 入力データファイルの作成方法

ここでは、上述した3つの入力データファイルの作成方法を示す。

ファイル形式はいずれもテキスト形式 (カンマ区切り、文字コード: Shift-JIS、改行コード: CR+LF) としている。この形式のファイルは、Microsoft Excel によって容易に編集することができることから、以降に示すデータフォーマットも Microsoft Excel による編集を想定し、表形式で示している。なお、文字コードが UTF-8 である場合など、ファイル形式、文字コード、改行コードが上記と異なる csv ファイルは、本プログラムで正しく読み込むことができないので注意を要する。

1) layer.csv

地盤条件を指定するためのファイルである。地層構成とパラメータの名称を図-A3.1 に示す。

このファイルには、地表 ($j=1$) から基盤 ($j = nl+1$) までの各地層の深度、物性値を入力する。プログラムでは、第 $nl+1$ 層の上面以深に第 $nl+1$ 層と同一の物性値を有する地層が無限に連続するものとして扱う。設計時と完成時で盛土の有無が異なる場合は、完成時の地盤条件を入力する。

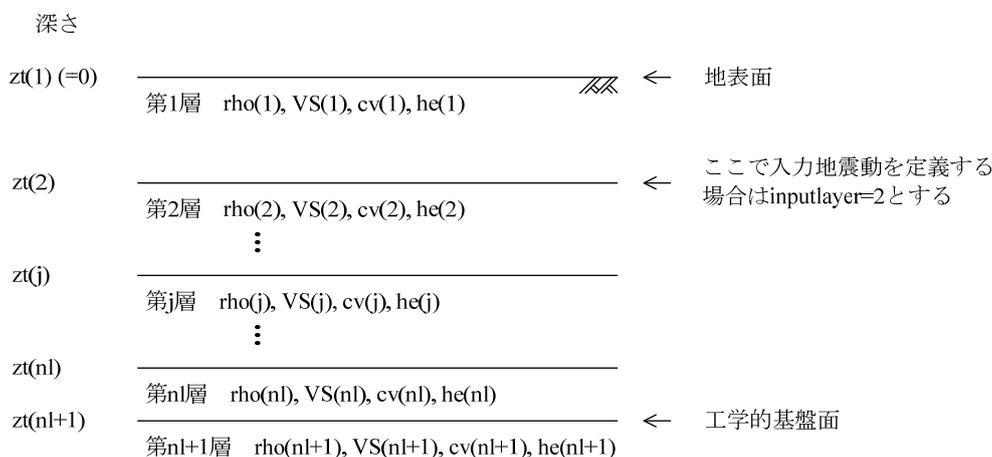


図-A3.1 地層構成と物性値の記号

このファイルのフォーマットを以下に示す。

	1 列目	2 列目	3 列目	4 列目	5 列目
1 行目	zt(1)	rho(1)	VS(1)	cv(1)	he(1)
2 行目	zt(2)	rho(2)	VS(2)	cv(2)	he(2)
...

ここに、

- zt(j) : 地表から j 番目の地層上面までの深度 [m]
- rho(j) : 地表から j 番目の地層の密度 [t/m^3]
- VS(j) : 地表から j 番目の地層の S 波速度 (微小ひずみ域) [m/s]
- cv(j) : 地表から j 番目の地層の S 波速度比。表-4.2 のうち Case2 の値を推奨する。
- he(j) : 地表から j 番目の地層の要素減衰定数。式(2.87)において最大減衰定数 $h_{max}=0.20$ として算出される値を推奨する。

各地層上面の深度 zt(j)の入力における留意事項を以下に示す。

- 第 1 層目の上面の深度 zt(1) = 0 とし、鉛直下向きを正とする。

- $z_t(j+1) - z_t(j)$ として算出した j 番目の地層の層厚が零または負の場合は、エラーストップする。このため、各地層上面の深度を地表面側から下方の順に指定しなければならない。
- 最下層を工学的基盤として扱う。地層の総数 (データ行数) の制限はない。

2) S_AG.csv

入力地震動に関するファイルであり、離散的な加速度応答スペクトルとして入力する。プログラムでは、ユーザーが入力した固有周期 T 、加速度応答スペクトル S_AG の離散値を、図-A3.2のように補間して使用する。すなわち、

- 1 点目 ($T(1), S_AG(1)$) より短周期側では、 $S_AG(1)$ を一定として外挿する。
- nsp 点目 ($T(nsp), S_AG(nsp)$) より長周期側では、両対数軸上で第 $nsp-1$ 点目と第 nsp 点目を結んだ直線によって外挿する。ここで、 nsp はユーザーが入力した加速度応答スペクトルの点数である。
- 上記以外の周期帯では、ユーザーが入力した (T, S_AG) の離散点を両対数軸上で線形補間して使用する。

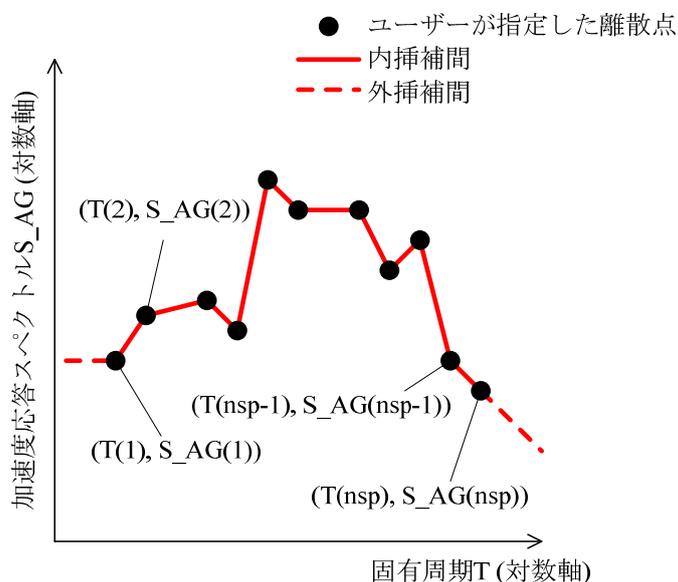


図-A3.2 加速度応答スペクトルの入力データと補間方法

このファイルのフォーマットを以下に示す。

	1 列目	2 列目
1 行目	cz	inputlayer
2 行目	T(1)	S_AG(1)
3 行目	T(2)	S_AG(2)
...

ここに、

- cz** : 加速度応答スペクトルの振幅調整係数。地域別補正係数の入力を想定したものであるが、その他にも、単位換算 ($gal \rightarrow m/s^2$) などに使用することもできる。
- inputlayer** : 入力地震動の定義位置 (耐震設計上の地盤面) を上面とする地層の番号。inputlayer ≥ 1 の場合は、与えられた加速度応答スペクトル S_AG を inputlayer の上面における 2E 成分の値であるとみなして計算する。inputlayer = 0 の場合は、 S_AG をモデル最下面 ($n+1$ 層上面) における E+F 成分の値であるとみなして計算する。

T(j) : 短周期側から j 点目の固有周期 [s]
 S_AG(j) : 短周期側から j 点目の加速度応答スペクトル [m/s²]

このファイルの作成における留意事項を以下に示す。

- 入力する加速度応答スペクトル S_AG は、減衰定数 $h=5\%$ として算出されたものでなければならない。
- 加速度応答スペクトル S_AG が 2E 境界で定義された値の場合は inputlayer ≥ 1 としなければならない。また、加速度応答スペクトル S_AG が工学的基盤面における E+F 境界で定義された値の場合は inputlayer=0 としなければならない。
- ユーザーが入力した加速度応答スペクトル S_AG に対し、プログラムでは振幅調整係数 cz を乗じて使用する。したがって、地域別補正係数として 1.0 以外の値を設定する場合は、S_AG として加速度応答スペクトルの標準値 (地域別補正係数を乗じる前の値) を指定しなければならない。
- 固有周期 T は、短周期側から順に入力しなければならない。T の入力順序が逆転している箇所がある場合は、エラーストップする。
- 加速度応答スペクトルの点数 nsp (データ行数) の制限はない。

以下、パラメータ inputlayer を設定する際の参考情報として、入力地震動の E+F 成分と 2E 成分の違いについて説明しておく。

- 地中で観測される地震動波形は、下方から伝播する上昇成分 (E) と上方から伝播する下降成分 (F) の和として表される。したがって、地中で観測される地震動波形は全て E+F 成分である。上昇成分 E と下降成分 F の振幅比は地盤条件に固有なものとして決まるため、E+F 成分の地震動を地盤条件とは無関係に地中から入力することは、「上部地層の振動が当該地層の振動に全く影響しないとみなせるほどに当該地層が大きな剛性を有するものと仮定して地震動を入力する」ことを意味する。このため、地震動の E+F 成分を地中に入力する方法は剛基盤入力と呼ばれることもある。時間領域の地震応答解析では、モデル下面の変位を固定して地震動を入力する方法が剛基盤入力に相当する。
- 地表面はせん断応力が零の境界であるため、下方から伝播した上昇波 (E) が全反射して下降波 (F) に転ずる。したがって、地表で観測される地震動波形は全て 2E 成分である。地中の地震動波形を重複反射理論によって上昇成分 E と下降成分 F に分離し、上昇成分 E の振幅に 2 を乗じることで、2E 成分を得ることができる。2E 成分の地震動波形を地中から入力することは、「仮に上部地層が存在しなかった場合に当該地層上面で観測されるであろう波形を、当該地盤の速度構造を考慮した上で入力する」ことを意味する。このため、2E 境界で定義された地震動を地中に入力することは弾性基盤入力と呼ばれることもある。時間領域の地震応答解析では、粘性境界を介してモデル下面に地震動を入力する方法が弾性基盤入力に相当する。

3) depth.csv

計算結果を出力する深度を指定するためのファイルである。このファイルのフォーマットを以下に示す。

	1 列目
1 行目	zo(1)
2 行目	zo(2)
...	...

ここに、

zo(i) : j 番目の出力深度 [m]

出力深度 zo(j)の入力における留意事項を以下に示す。

- ・ ユーザーが指定した出力深度 zo(j)が負の場合、または z(nl+1)より大きい場合は、その深度を読み飛ばして処理を行う。なお、z(nl+1)はモデル最下面の深度である。
- ・ 出力深度の点数 (データ行数) の制限はない。指定順序も任意。

A4 出力データファイル

計算が正常に完了すると、プログラムは計算結果として下記 4)、5) のファイルを出力する。これらの出力データファイルの形式も、テキスト形式 (カンマ区切り、文字コード : Shift-JIS、改行コード : CR+LF) である。

4) mode.csv

1 次振動モードに関する情報。フォーマットは次のとおり。なお、下線部は定型的に出力する文字列である。

	1 列目	2 列目		1 列目	2 列目
1 行目	<u>mode</u>	<u>1</u>	11 行目	<u>zo</u>	<u>phi(1)</u>
2 行目	<u>fs</u>	fs(1)	12 行目	zo(1)	phi(1,1)
3 行目	<u>Ts</u>	1/fs(1)	13 行目	zo(2)	phi(2,1)
4 行目	<u>h</u>	hm(1)
5 行目	<u>cD</u>	cD(1)	(11+no)列目	zo(no)	phi(no,1)
6 行目	<u>beta</u>	beta(1)	(12+no)列目	<u>zo</u>	<u>dphi(1)</u>
7 行目	<u>rMe</u>	rMe(1)	(13+no)列目	zo(1)	dphi(1,1)
8 行目	<u>absH</u>	absH(1)	(14+no)列目	zo(2)	dphi(2,1)
9 行目	<u>S_AG</u>	S_AG(1)
10 行目	<u>S_DB</u>	S_DB(1)	(11+2no)列目	zo(no)	dphi(no,1)

ここに、

- fs(1) : 1 次固有振動数 [Hz]
- hm(1) : 1 次モード減衰定数
- cD(1) : 1 次モード減衰定数 h(1)に対する減衰定数別補正係数
- beta(1) : 1 次モードの刺激係数
- rMe(1) : 1 次モードの有効質量比
- absH(1) : 1 次固有振動数における伝達関数の絶対値
- S_AG(1) : 入力地震動の定義位置 inputlayer における 1 次固有周期の加速度応答スペクトル (減衰定数 $h=5\%$) [m/s²]. ユーザーが [S_AG.csv] で指定した値の補間により算出したもの。
- S_DB(1) : 工学的基盤面における 1 次固有周期の変位応答スペクトル (減衰定数 $h_1=hm(1)$) [m].
 - zo(j) : j 番目の出力深度
 - phi(j,1) : 深度 zo(j)における 1 次のモード分布 ϕ
 - dphi(j,1) : 深度 zo(j)における 1 次のモード分布の空間微分 $d\phi/dz$

5) response.csv

1 次振動モードの最大応答値分布。フォーマットは次のとおり。なお、下線部は定型的に出力する文字列である。

	1 列目	2 列目	3 列目	4 列目	5 列目	6 列目
1 行目	<u>z</u>	<u>Amax</u>	<u>Vmax</u>	<u>Dmax</u>	<u>gmamax</u>	<u>taumax</u>
2 行目	zo(1)	Amax(1)	Vmax(1)	Dmax(1)	gmamax(1)	taumax(1)
3 行目	zo(2)	Amax(2)	Vmax(2)	Dmax(2)	gmamax(2)	taumax(2)
...

ここに、

zo(j) : j 番目の出力深度 [m]

Amax(j) : 出力深度 zo(j)における最大絶対加速度 [m/s²]

Vmax(j) : 出力深度 zo(j)における最大相対速度 [m/s]

Dmax(j) : 出力深度 zo(j)における最大相対変位 [m]

gmamax(j) : 出力深度 zo(j)における最大せん断ひずみ

taumax(j) : 出力深度 zo(j)における最大せん断応力 [kN/m²]

A5 ソースコード

```
!   引き戻し・応答スペクトル法による表層地盤の地震応答計算プログラム
!   decrespec ver.220428
!   著作権者：国立研究開発法人土木研究所
!*****
!   module commonparameters
!*****
!   integer*4 nl, nm
!   real*8,dimension(:),allocatable::zt, th, rho, VS, cv, VSD, he
!   real*8,dimension(:),allocatable::R
!
!   real*8,dimension(:),allocatable::A0, B0
!   real*8,dimension(:,:),allocatable::A, B
!   real*8,dimension(:,:),allocatable::phi, dphi, Gdphi
!   real*8,dimension(:),allocatable::fn, hm, cD, beta, rMe
!
!   integer*4 nsp
!   real*8 cz
!   real*8,dimension(:,:),allocatable::S_AG0
!   real*8,dimension(:),allocatable::S_AG, S_DB, absH
!
!   end module
!*****
!   program decrespec
!*****
!   use commonparameters
!   implicit none
!   temporary
!   real*8 Ts, df
!   real*8 tmp, tmp1, tmp2, tmp3, tmp4
!   real*8 fL, fC, fU
!   real*8 UbL, UbC, UbU
!   real*8 C, S, CC, CS, SS
!   integer*4 ndf
!   real*8 wn
!   real*8 erra
!   real*8 Mtotal
!   integer*4 no
!   real*8,dimension(:),allocatable::zo
!   real*8 zl
!   real*8 Amax, Vmax, Dmax, gmamax, taumax
!   integer*4 inputlayer
!   character*3 ctmp
!   counter
```

```

integer*4 i, j, k
! subfunction
integer*4 countrows
real*8 pi, Ub, S_AGint, xfer2

!-----

write(*,*)
write(*,*)
write(*,*)
write(*,"(--- 引き戻し・応答スペクトル法による表層地盤の地震応答計算プログラム)")
write(*,"(--- decrespec ver.220428)")
write(*,"(--- 著作権者：国立研究開発法人土木研究所)")
write(*,*)

!-----

! 1. データ読み込み

!-----

nm = 1 ! モード次数を1次に限定
ndf = 50
erra = 1.d-12
allocate(fn(0:nm))
allocate(hm(nm))
allocate(cD(nm))
allocate(beta(nm))
allocate(rMe(nm))
allocate(S_AG(nm))
allocate(S_DB(nm))
allocate(absH(nm))

!
! 地盤条件
! - 読み込み
open(10,file='layer.csv')
nl = countrows(10) - 1
allocate(zt (nl+1))
allocate(th (nl))
allocate(rho(nl+1))
allocate(VS (nl+1))
allocate(cv (nl+1))
allocate(VSD(nl+1))
allocate(he (nl+1))
allocate(A0 (nl+1))
allocate(B0 (nl+1))
allocate(A (nl+1,nm))
allocate(B (nl+1,nm))
allocate(R (nl))
rewind 10
do j = 1, nl+1

```

```

    read(10,*) zt(j), rho(j), VS(j), cv(j), he(j)
    VSD(j) = cv(j) * VS(j)
end do
close(10)
! - エラーチェック
if(zt(1) /= 0.d0) then
    write(*, "('エラー : 1 層目の上面深度が 0 になっていません')")
    stop
end if
do j = 1, nl
    if(zt(j+1) - zt(j) <= 0.d0) then
        write(*, "('エラー : 層厚が零または負になっています')")
        write(*,*) '地層番号 j=', j
        stop
    end if
end do
! - 計算準備
Ts = 0.d0
Mtotal = 0.d0
do j = 1, nl
    th(j) = zt(j+1) - zt(j)
    Mtotal = Mtotal + rho(j) * th(j)
    R(j) = rho(j) * VSD(j) / rho(j+1) / VSD(j+1)
    Ts = Ts + 4.d0 * th(j) / VSD(j)
end do
!
! 加速度応答スペクトル
! - 読み込み
open(10, file='S_AG.csv')
nsp = countrows(10) - 1
allocate(S_AG0(nsp,2))
rewind 10
read(10,*) cz, inputlayer
do i = 1, nsp
    read(10,*) (S_AG0(i,j), j = 1, 2)
end do
S_AG0(:,2) = cz * S_AG0(:,2)
close(10)
! - エラーチェック
if((inputlayer < 0) .or. (nl+1 < inputlayer)) then
    write(*,*) 'fatal error!: inputlayer の設定が不適切です'
    write(*,*) 'inputlayer =', inputlayer
    stop
end if
do i = 1, nsp - 1

```

```

if(S_AG0(i,1) >= S_AG0(i+1,1)) then
  write(*, "('エラー：固有周期が短周期側から順に入力されていません')")
  write(*, ', ', i, ', ', i+1, '点目'
  stop
end if
end do

```

!

! 出力深度

! - 読み込み

```

open(10, file='depth.csv')
no = countrows(10)
allocate(zo(no))
rewind 10
j = 0
do i = 1, no
  read(10, *) tmp
  if(tmp < 0.d0) cycle
  if(tmp > zt(nl+1)) cycle
  j = j + 1
  zo(j) = tmp
end do
no = j

```

! do i = 1, no

! read(10, *) zo(i)

! end do

```

close(10)

```

!-----

! 2. 固有振動数の計算

!-----

```

df = 1.d0 / Ts / dfloat(ndf)
fL = 0.d0 ; UbL = Ub(0.d0)

```

!

```

do k = 1, nm

```

!

! 挟み撃ち法 (step1:はさむ)

```

  fL = fn(k-1) + df ; UbL = Ub(fL)
  do
    fU = fL + df ; UbU = Ub(fU)
    if(UbL * UbU <= 0.d0) exit
    fL = fU ; UbL = UbU
  end do

```

!

! 挟み撃ち法 (step2:狭める)

```

  do
    fC = (fL + fU) / 2.d0 ; UbC = Ub(fC)

```

```

if(abs(UbL) <= erra) then
  fn(k) = fL
  fL = fn(k) + df
  fU = fL + df
  exit
end if
if(abs(UbU) <= erra) then
  fn(k) = fU
  fL = fn(k) + df
  fU = fL + df
  exit
end if
if(abs(UbC) <= erra) then
  fn(k) = fC
  fL = fn(k) + df
  fU = fL + df
  exit
end if
if(UbL * UbC < 0.d0) then
  fU = fC ; UbU = UbC
else
  fL = fC ; UbL = UbC
end if
end do
!
end do

```

!-----

! 3. モード分布の計算

!-----

```

do k = 1, nm
  call calcAB0(fn(k))
  A(:,k) = A0(:)
  B(:,k) = B0(:)
end do
!
allocate(phi(no,nm))
allocate(dphi(no,nm))
allocate(Gdphi(no,nm))
do i = 1, no
  call localz(zo(i), zl, j)
do k = 1, nm
  wn = 2.d0 * pi() * fn(k)
  phi(i,k) = A(j,k) * cos(wn * zl / VSD(j)) + B(j,k) * sin(wn * zl / VSD(j))
  dphi(i,k) = wn / VSD(j) * (- A(j,k) * sin(wn * zl / VSD(j)) + B(j,k) * cos(wn * zl / VSD(j)))
  Gdphi(i,k) = (rho(j) * VSD(j) ** 2.d0) * dphi(i,k)

```

```
end do
end do
```

```
! 4. モード減衰, 刺激係数, 有効質量比
```

```
do k = 1, nm
  tmp1 = 0.d0 ; tmp2 = 0.d0 ; tmp3 = 0.d0 ; tmp4 = 0.d0
  wn = 2.d0 * pi() * fn(k)
  do j = 1, nl
    tmp = wn * th(j) / VSD(j)
    C = VSD(j) / wn * sin(tmp)
    S = VSD(j) / wn * (1.d0 - cos(tmp))
    CC = (2.d0 * th(j) + VSD(j) / wn * sin(2.d0 * tmp)) / 4.d0
    CS = VSD(j) / wn / 4.d0 * (1.d0 - cos(2.d0 * tmp))
    SS = (2.d0 * th(j) - VSD(j) / wn * sin(2.d0 * tmp)) / 4.d0
    tmp3 = tmp3 + he(j) * rho(j) * &
    & (wn ** 2.d0) * ((A(j,k) ** 2.d0) * SS - 2.d0 * A(j,k) * B(j,k) * CS + (B(j,k) **2.d0) * CC)
    tmp4 = tmp4 + rho(j) * (wn ** 2.d0) * &
    & ((A(j,k) ** 2.d0) * SS - 2.d0 * A(j,k) * B(j,k) * CS + (B(j,k) **2.d0) * CC)
    tmp1 = tmp1 + rho(j) * (A(j,k) * C + B(j,k) * S)
    tmp2 = tmp2 + rho(j) * ((A(j,k) ** 2.d0) * CC + 2.d0 * A(j,k) * B(j,k) * CS + (B(j,k) **2.d0) * SS)
  end do
  hm(k) = tmp3 / tmp4
  beta(k) = tmp1 / tmp2
  rMe(k) = (tmp1 ** 2.d0) / tmp2
end do

!
cD(:) = 1.5d0 / (40.d0 * hm(:) + 1.d0) + 0.5d0
rMe(:) = rMe(:) / Mtotal
```

```
! 5. 基盤地震動(E+F)
```

```
do k = 1, nm
  if(inputlayer == 0) then
    absH(k) = 1.d0
  else
    absH(k) = xfer2(1.d0 / fn(k), inputlayer)
  end if
  S_AG(k) = S_AGint(1.d0 / fn(k))
  S_DB(k) = cD(k) * S_AG(k) / absH(k) / (2.d0 * pi() * fn(k)) ** 2.d0
end do
```

```
! 6. モード分解の結果出力
```

```
open(30,file='mode.csv')
```

```

write(30,("mode,'*(i3,')")) (k, k = 1, nm)
write(30,("fs,'*(e13.6,')")) (fn(k), k = 1, nm)
write(30,("Ts,'*(e13.6,')")) (1.d0 / fn(k), k = 1, nm)
write(30,("h,'*(e13.6,')")) (hm(k), k = 1, nm)
write(30,("cD,'*(e13.6,')")) (cD(k), k = 1, nm)
write(30,("beta,'*(e13.6,')")) (beta(k), k = 1, nm)
write(30,("rMe,'*(e13.6,')")) (rMe(k), k = 1, nm)
write(30,("absH,'*(e13.6,')")) (absH(k), k = 1, nm)
write(30,("S_AG,'*(e13.6,')")) (S_AG(k), k = 1, nm)
write(30,("S_DB,'*(e13.6,')")) (S_DB(k), k = 1, nm)

```

!

```

write(ctmp,"(i3.3)") nm
write(30,("zo,'" // trim(ctmp) // "(phi('i3,')")) (k, k = 1, nm)
do i = 1, no
  write(30,("e13.7,';*(e13.6,')")) zo(i), (phi(i, k), k = 1, nm)
end do

```

!

```

write(ctmp,"(i3.3)") nm
write(30,("zo,'" // trim(ctmp) // "(dphi('i3,')")) (k, k = 1, nm)
do i = 1, no
  write(30,("e13.7,';*(e13.6,')")) zo(i), (dphi(i, k), k = 1, nm)
end do

```

!

```

close(30)

```

!-----

! 7. 最大応答値の計算, 出力

!-----

```

open(30,file='response.csv')
write(30,("z,Amax,Vmax,Dmax,gmamax,taumax'))"
do i = 1, no
  Amax = 0.d0
  Vmax = 0.d0
  Dmax = 0.d0
  gmamax = 0.d0
  taumax = 0.d0
do k = 1, nm
  Amax = Amax + (S_DB(k) * ((2.d0 * pi() * fn(k)) ** 2.d0) * beta(k) * phi(i,k)) ** 2.d0
  Vmax = Vmax + (S_DB(k) * (2.d0 * pi() * fn(k)) * beta(k) * phi(i,k)) ** 2.d0
  Dmax = Dmax + (S_DB(k) * beta(k) * phi(i,k)) ** 2.d0
  gmamax = gmamax + (S_DB(k) * beta(k) * dphi(i,k)) ** 2.d0
  taumax = taumax + (S_DB(k) * beta(k) * Gdphi(i,k)) ** 2.d0
end do
  Amax = sqrt(Amax)
  Vmax = sqrt(Vmax)
  Dmax = sqrt(Dmax)

```

```

    gmamax = sqrt(gmamax)
    taumax = sqrt(taumax)
    write(30,"(6(e13.6,','))")zo(i), Amax, Vmax, Dmax, gmamax, taumax
end do
close(30)
!
write(*,"(--- 計算完了)")
write(*,*)
write(*,*)
write(*,*)
!
end program
!*****
!   円周率
    real*8 function pi()
!*****
    implicit none
!-----
    pi = 3.14159265358979d0
    end function
!*****
!   ファイルの行数を数える
    integer*4 function countrows(ifile)
!*****
    implicit none
    integer*4 ifile
!-----
    countrows = 0
    rewind ifile
!
100  read(ifile,*,end=999)
    countrows = countrows + 1
    goto 100
!
999  continue
    end function
!*****
!   与えられた周波数 f に対応するモード分布(A0, B0)の計算
    subroutine calcAB0(f)
!*****
    use commonparameters
    implicit none
    real*8 f, w, tmp
    integer*4 j
    real*8 pi

```

```

!-----
A0(1) = 1.d0 ; B0(1) = 0.d0
w = 2.d0 * pi() * f
!
do j = 1, nl
  tmp = w * th(j) / VSD(j)
  A0(j+1) = A0(j) * cos(tmp) + B0(j) * sin(tmp)
  B0(j+1) = - R(j) * A0(j) * sin(tmp) + R(j) * B0(j) * cos(tmp)
end do
!
end subroutine
!*****
! モデル下面の水平変位 Ub の計算
real*8 function Ub(f)
!*****
  use commonparameters
  implicit none
  real*8 f
!-----
  call calcAB0(f)
  Ub = A0(nl+1)
!
end function
!*****
! 地表からの深度 z を含む地層の番号 j を検索し、j 層上面からの深度 zl を求める
subroutine localz(zs, zl, j)
!*****
  use commonparameters
  implicit none
  real*8 zs ! 地表からの深度 (ユーザー入力)
  real*8 zl ! 層上面からの深度
  integer*4 i, j
!-----
  do i = 1, nl
    if(zs <= zt(i+1)) then
      j = i
      zl = zs - zt(i)
      exit
    end if
  end do
!
end subroutine
!*****
! 加速度応答スペクトルの補間
real*8 function S_AGint(T)

```

```

!*****
use commonparameters
implicit none
real*8 T
integer*4 i

!-----

S_AGint = S_AG0(1,2)

!
do i = 1, nsp
if(T <= S_AG0(i,1)) then
if(i == 1) exit
S_AGint = (log(T) - log(S_AG0(i-1,1))) * &
& (log(S_AG0(i,2)) - log(S_AG0(i-1,2))) / (log(S_AG0(i,1)) - log(S_AG0(i-1,1))) &
& + log(S_AG0(i-1,2))
S_AGint = exp(S_AGint)
exit
end if
end do

!
end function
!*****
! 重複反射理論による伝達関数 (第 k 層上面 2E / 基盤 E+F) の計算
real*8 function xfer2(T, k)
!*****

use commonparameters
implicit none
real*8 w, T
complex*16,dimension(:),allocatable::E, F, Vsc
integer*4 j, k
complex*16 i, R2
real*8 pi

!-----

i = dcmplx(0.d0, 1.d0)
w = 2.d0 * pi() / T
allocate(E(nl+1))
allocate(F(nl+1))
allocate(Vsc(nl+1))
Vsc(:) = VSD(:) * cdsqrt(dcmplx(1.d0, 2.d0 * he(:)))

!
E(1) = dcmplx(1.d0, 0.d0) ; F(1) = dcmplx(1.d0, 0.d0)
do j = 1, nl
R2 = rho(j) * Vsc(j) / rho(j+1) / Vsc(j+1)
E(j+1) = + (1.d0 + R2) / 2.d0 * E(j) * cdexp(+dcmplx(0.d0, 1.d0) * w * th(j) / Vsc(j)) &
& + (1.d0 - R2) / 2.d0 * F(j) * cdexp(-dcmplx(0.d0, 1.d0) * w * th(j) / Vsc(j))
F(j+1) = + (1.d0 - R2) / 2.d0 * E(j) * cdexp(+dcmplx(0.d0, 1.d0) * w * th(j) / Vsc(j)) &

```

```
      &      + (1.d0 + R2) / 2.d0 * F(j) * cdexp(-dcmplx(0.d0, 1.d0) * w * th(j) / Vsc(j))
end do
!
xfer2 = cdabs(2.d0 * E(k) / (E(nl+1) + F(nl+1)))
!
end function
```