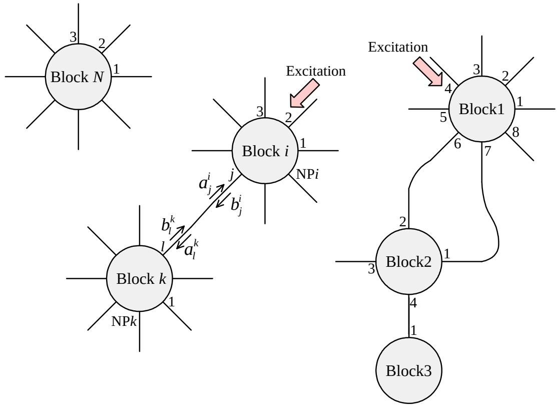
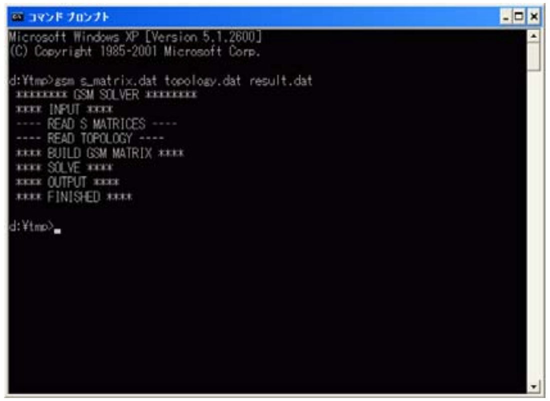
# GSM Solver

## - Development of Free Software to Solve Arbitrarily-Connected Scattering Matrix Network -



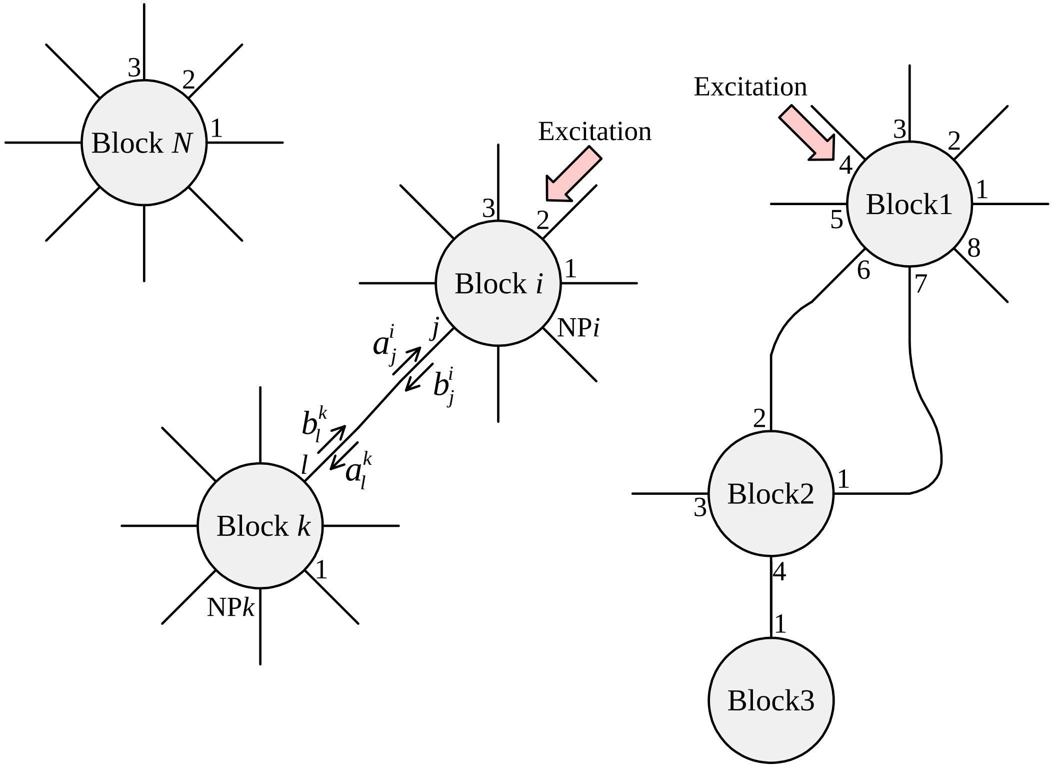


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

d:¥tmp>gsm s_matrix.dat topology.dat result.dat
 ******** GSM SOLVER ********
 **** INPUT ****
 ---- READ S MATRICES ----
 ---- READ TOPOLOGY ----
 **** BUILD GSM MATRIX ****
 **** SOLVE ****
 **** OUTPUT ****
 **** FINISHED ****

d:¥tmp>
```

Takuichi Hirano

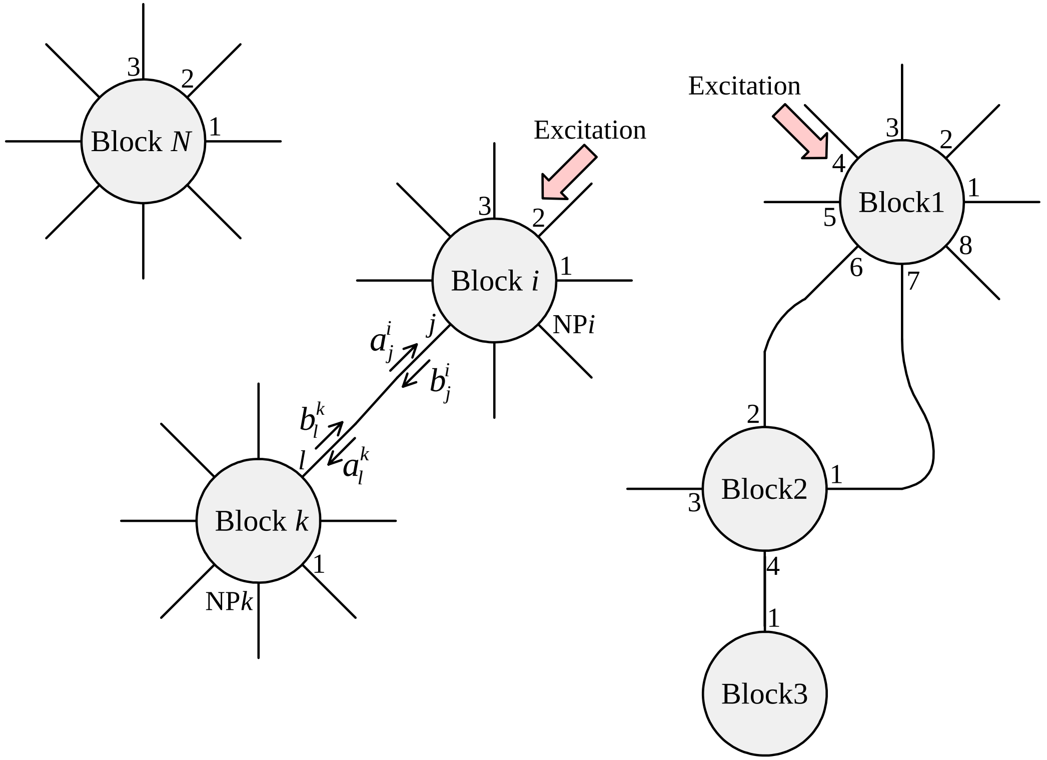
Tokyo Institute of Technology

# Objective


Block $N$
3
2
1
Excitation
Block $i$
3
2
1
$j$
NP$i$
$a_j^i$
$b_j^i$
$b_l^k$
$l$
$a_l^k$
Block $k$
1
NP$k$
Excitation
Block1
3
2
1
4
5
6
7
8
Block2
2
1
3
4
1
Block3

# To Solve S-matrix Network

Generalized Scattering Matrix (GSM) Solver is a free software to solve connected scattering matrix network. User can specify arbitrarily-connected scattering matrix network by using two input files. One input file specifies scattering matrices of each block, another one describes connection network, excitation and matched-load terminal condition.

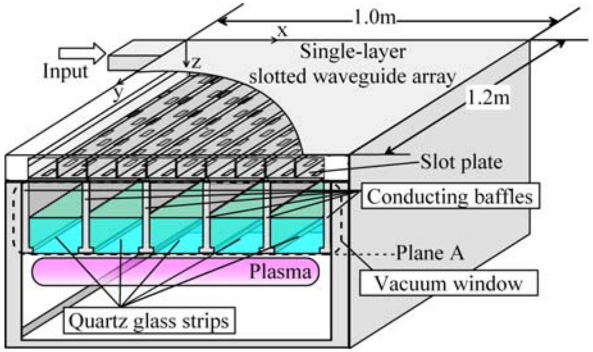


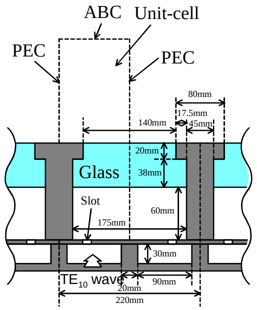
# Motivation of Development (1)

## Personal Point of View

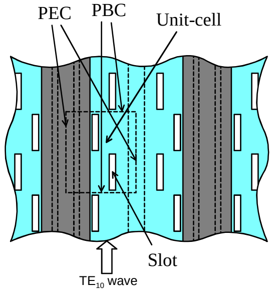
GSM-MoM analysis for a unit-cell slotted waveguide arrays for plasma excitation.

(… because MoM analysis failed. Green's function for a waveguide cavity with the length on the order of a half guide wavelength [e.g) Block2, Block5] has singularity.)

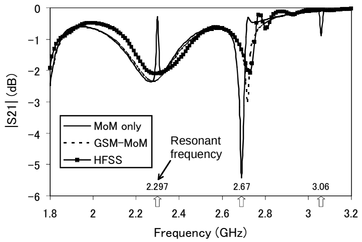


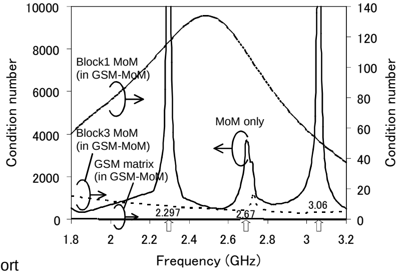


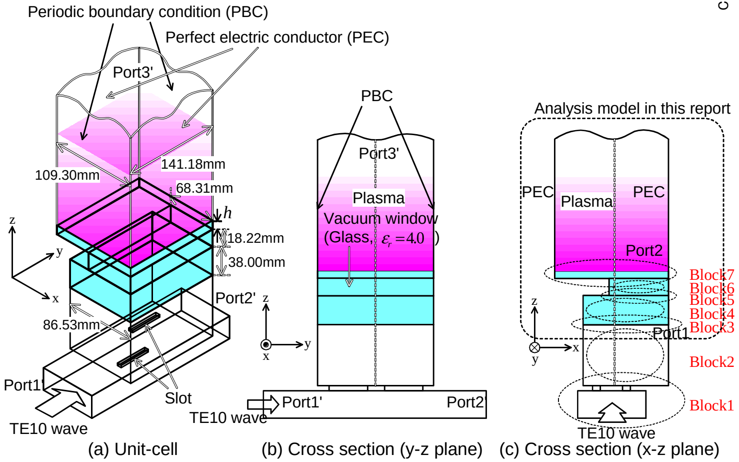
(a) Cross section



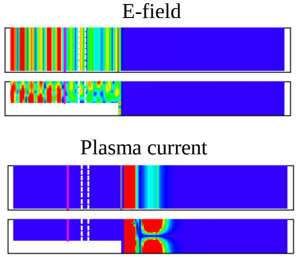
(b) Top view







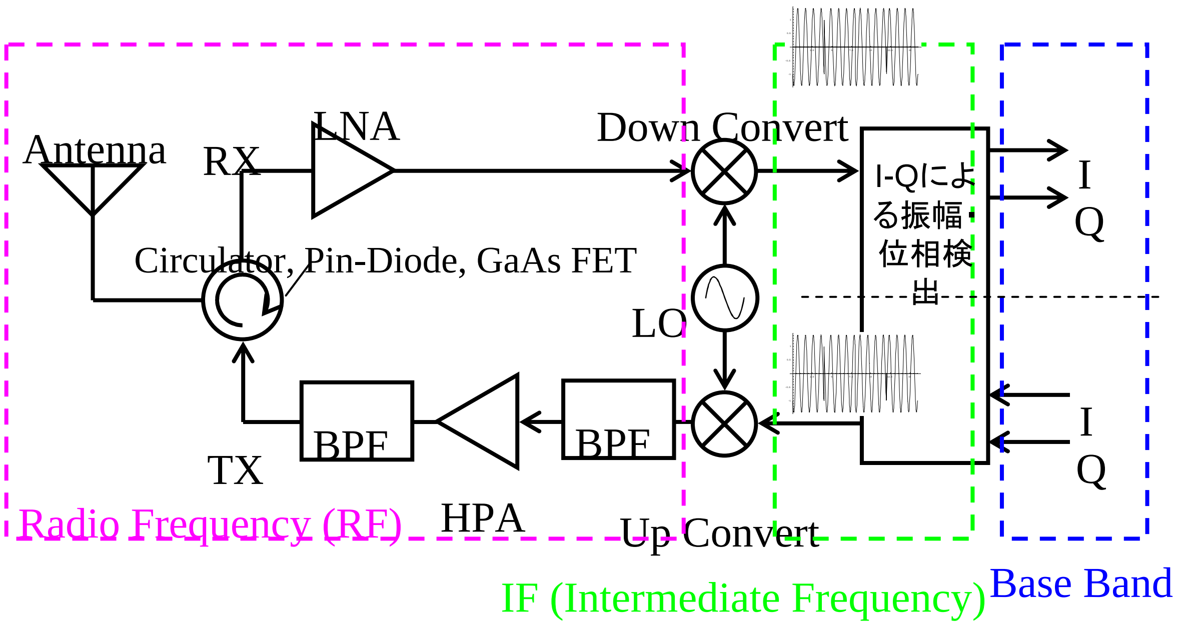
(a) Unit-cell (b) Cross section (y-z plane) (c) Cross section (x-z plane)



# Motivation of Development (2)

## Prospective Point of View

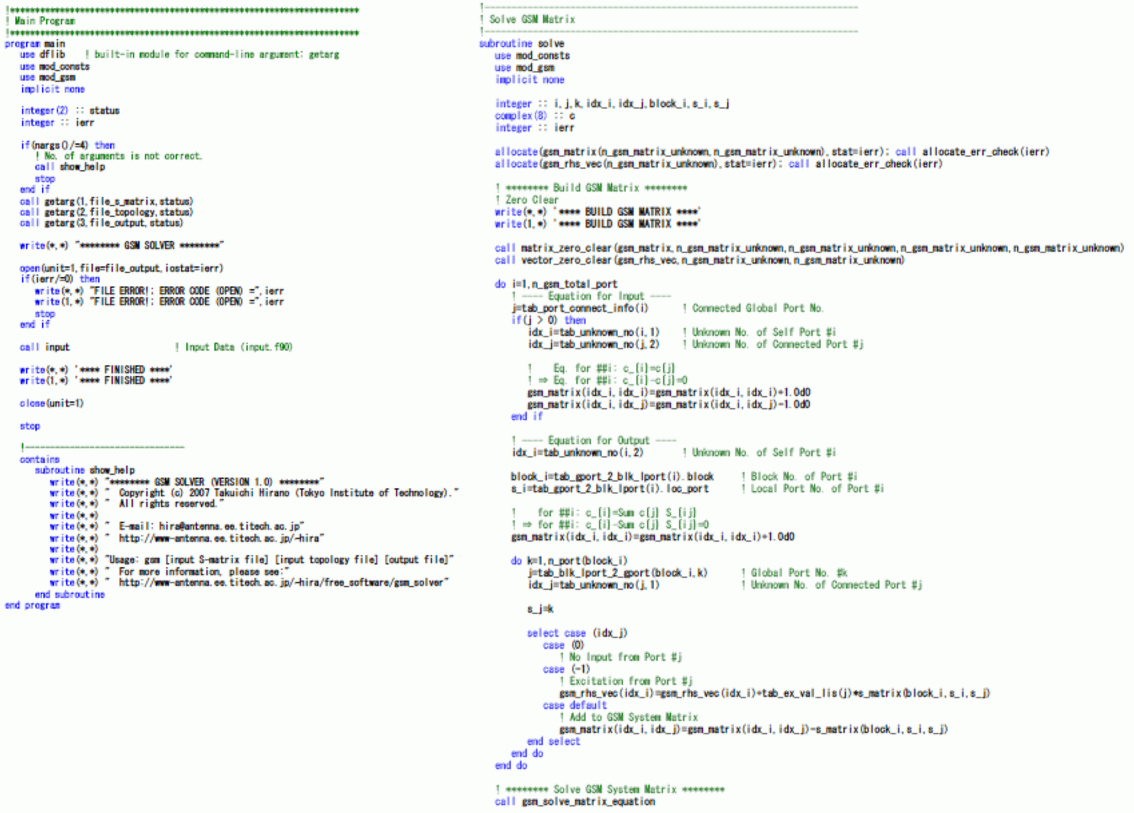
- Useful for many kinds of microwave system analysis and design.
- Development of millimeter wave system.



- ➢Full-wave analysis (solving Maxwell’s equation) is not practical.
- ➢S-matrix connection is practical.
- ➢If higher-order mode couplings are considered, the accuracy is the same as that of full-wave analysis.

LNA: Low Noise Amplifier
HPF: High Power Amplifier

# Algorithm

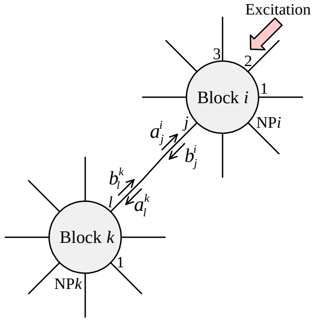


```
!*********************************************************************
! Main Program
!*********************************************************************
program main
   use dflib    ! built-in module for command-line argument: getarg
   use mod_consts
   use mod_gsm
   implicit none

   integer(2) :: status
   integer :: ierr

   if(nargs()/=4) then
      ! No. of arguments is not correct.
      call show_help
      stop
   end if
   call getarg(1,file_s_matrix,status)
   call getarg(2,file_topology,status)
   call getarg(3,file_output,status)

   write(*,*) "******** GSM SOLVER ********"

   open(unit=1,file=file_output,iostat=ierr)
   if(ierr/=0) then
      write(*,*) "FILE ERROR!: ERROR CODE (OPEN) =",ierr
      write(1,*) "FILE ERROR!: ERROR CODE (OPEN) =",ierr
      stop
   end if

   call input                        ! Input Data (input.f90)

   write(*,*) '**** FINISHED ****'
   write(1,*) '**** FINISHED ****'

   close(unit=1)

   stop

   !------------------------------
   contains
      subroutine show_help
         write(*,*) "******** GSM SOLVER (VERSION 1.0) ********"
         write(*,*) "  Copyright (c) 2007 Takuichi Hirano (Tokyo Institute of Technology)."
         write(*,*) "  All rights reserved."
         write(*,*)
         write(*,*) "  E-mail: hira@antenna.ee.titech.ac.jp"
         write(*,*) "  http://www-antenna.ee.titech.ac.jp/~hira"
         write(*,*)
         write(*,*) "Usage: gsm [input S-matrix file] [input topology file] [output file]"
         write(*,*) "  For more information, please see:"
         write(*,*) "  http://www-antenna.ee.titech.ac.jp/~hira/free_software/gsm_solver"
      end subroutine
end program
```

```
!--------------------------------------------------------------
! Solve GSM Matrix
!--------------------------------------------------------------
subroutine solve
   use mod_consts
   use mod_gsm
   implicit none

   integer :: i,j,k,idx_i,idx_j,block_i,s_i,s_j
   complex(8) :: c
   integer :: ierr

   allocate(gsm_matrix(n_gsm_matrix_unknown,n_gsm_matrix_unknown),stat=ierr); call allocate_err_check(ierr)
   allocate(gsm_rhs_vec(n_gsm_matrix_unknown),stat=ierr); call allocate_err_check(ierr)

   ! ******** Build GSM Matrix ********
   ! Zero Clear
   write(*,*) '**** BUILD GSM MATRIX ****'
   write(1,*) '**** BUILD GSM MATRIX ****'

   call matrix_zero_clear(gsm_matrix,n_gsm_matrix_unknown,n_gsm_matrix_unknown,n_gsm_matrix_unknown,n_gsm_matrix_unknown)
   call vector_zero_clear(gsm_rhs_vec,n_gsm_matrix_unknown,n_gsm_matrix_unknown)

   do i=1,n_gsm_total_port
      ! ---- Equation for Input ----
      j=tab_port_connect_info(i)      ! Connected Global Port No.
      if(j > 0) then
         idx_i=tab_unknown_no(i,1)    ! Unknown No. of Self Port #i
         idx_j=tab_unknown_no(j,2)    ! Unknown No. of Connected Port #j

         !    Eq. for ##i: c_[i]=c[j]
         ! => Eq. for ##i: c_[i]-c[j]=0
         gsm_matrix(idx_i,idx_i)=gsm_matrix(idx_i,idx_i)+1.0d0
         gsm_matrix(idx_i,idx_j)=gsm_matrix(idx_i,idx_j)-1.0d0
      end if

      ! ---- Equation for Output ----
      idx_i=tab_unknown_no(i,2)       ! Unknown No. of Self Port #i

      block_i=tab_gport_2_blk_lport(i).block     ! Block No. of Port #i
      s_i=tab_gport_2_blk_lport(i).loc_port      ! Local Port No. of Port #i

      !    for ##i: c_[i]=Sum c[j] S_[ij]
      ! => for ##i: c_[i]-Sum c[j] S_[ij]=0
      gsm_matrix(idx_i,idx_i)=gsm_matrix(idx_i,idx_i)+1.0d0

      do k=1,n_port(block_i)
         j=tab_blk_lport_2_gport(block_i,k)      ! Global Port No. #k
         idx_j=tab_unknown_no(j,1)               ! Unknown No. of Connected Port #j

         s_j=k

         select case (idx_j)
            case (0)
               ! No Input from Port #j
            case (-1)
               ! Excitation from Port #j
               gsm_rhs_vec(idx_i)=gsm_rhs_vec(idx_i)+tab_ex_val_lis(j)*s_matrix(block_i,s_i,s_j)
            case default
               ! Add to GSM System Matrix
               gsm_matrix(idx_i,idx_j)=gsm_matrix(idx_i,idx_j)-s_matrix(block_i,s_i,s_j)
         end select
      end do
   end do

   ! ******** Solve GSM System Matrix ********
   call gsm_solve_matrix_equation
```

# Building the System Matrix Equation

## Normal ports



$$\begin{cases} a_j^i = b_k^l & \text{(input)} \\ b_j^i = \sum_{n=1}^{NP_i} a_n^i S_{jn}^{(i)} & \text{(output)} \end{cases}$$

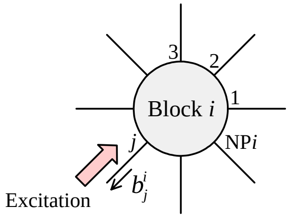
$$\Leftrightarrow \begin{cases} a_j^i = b_k^l & \text{(input)} \\ b_j^i = \sum_{n \in \text{Normal Port}} a_n^i S_{jn}^{(i)} + \sum_{n \in \text{Excitation}} c_n^i S_{jn}^{(i)} & \text{(output)} \end{cases}$$

$$\Leftrightarrow \begin{cases} a_j^i - b_k^l = 0 & \text{(input)} \\ b_j^i - \sum_{n \in \text{Normal Port}} a_n^i S_{jn}^{(i)} = \sum_{n \in \text{Excitation}} c_n^i S_{jn}^{(i)} & \text{(output)} \end{cases}$$

## Excitation or matched-loaded ports

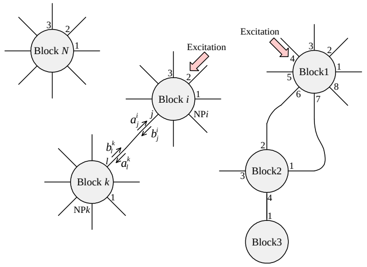
These ports are matched-terminated (no reflections)

$$\begin{cases} ---- & \text{(input)} \\ b_j^i = \sum_{n=1}^{NP_i} a_n^i S_{jn}^{(i)} & \text{(output)} \end{cases}$$



$$\Leftrightarrow \begin{cases} ---- & \text{(input)} \\ b_j^i = \sum_{n \in \text{Normal Port}} a_n^i S_{jn}^{(i)} + \sum_{n \in \text{Excitation}} c_n^i S_{jn}^{(i)} & \text{(output)} \end{cases}$$

$$\Leftrightarrow \begin{cases} ---- & \text{(input)} \\ b_j^i - \sum_{n \in \text{Normal Port}} a_n^i S_{jn}^{(i)} = \sum_{n \in \text{Excitation}} c_n^i S_{jn}^{(i)} & \text{(output)} \end{cases}$$

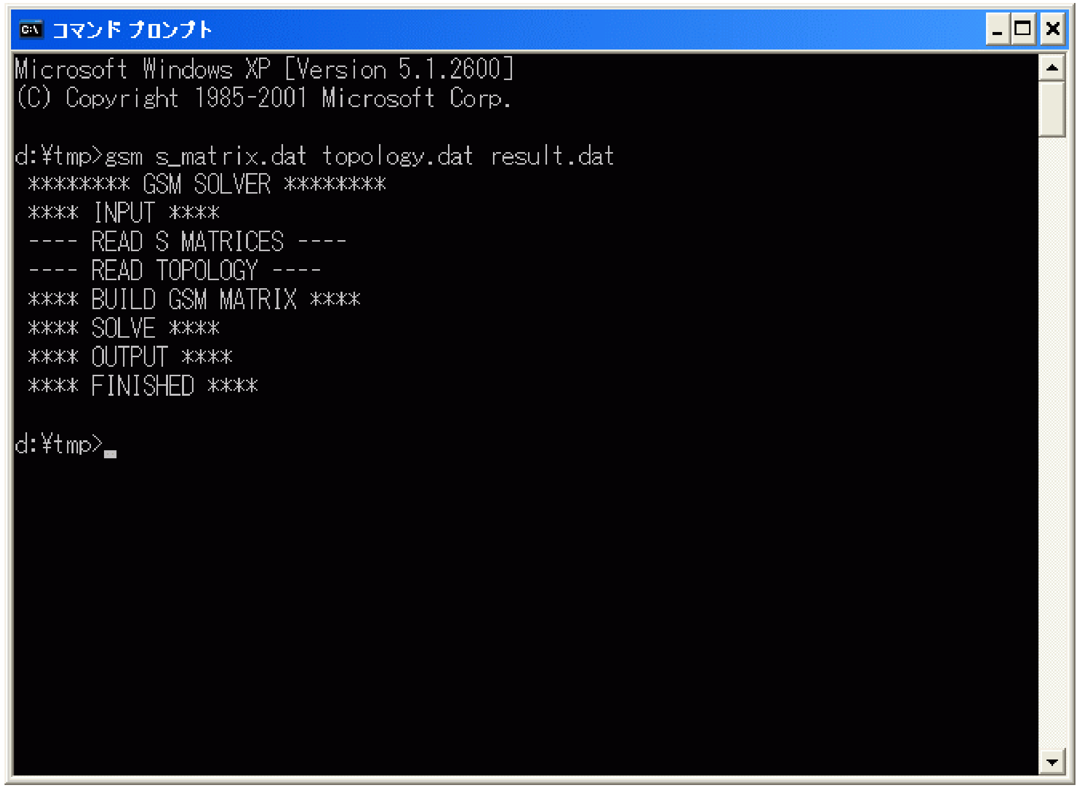


Linear equations with the same number as unknowns are build.

More details are written in technical notes:

http://www-antenna.ee.titech.ac.jp/~hira/free_software/gsm_solver/technical_notes/gsm_technical_notes.pdf

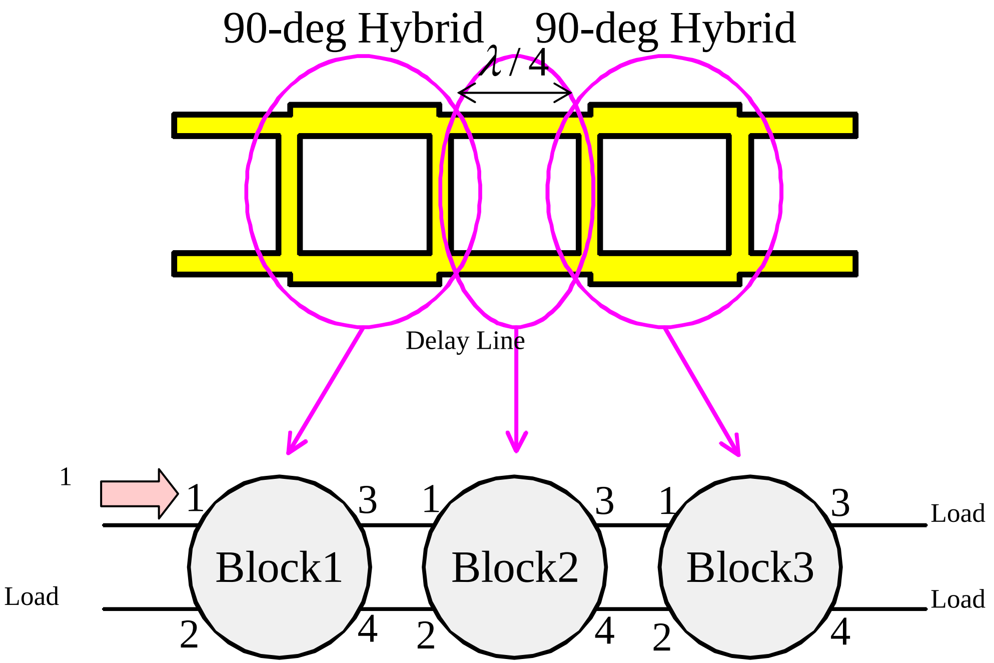
# Usage (Example)



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

d:¥tmp>gsm s_matrix.dat topology.dat result.dat
 ******** GSM SOLVER ********
 **** INPUT ****
 ---- READ S MATRICES ----
 ---- READ TOPOLOGY ----
 **** BUILD GSM MATRIX ****
 **** SOLVE ****
 **** OUTPUT ****
 **** FINISHED ****

d:¥tmp>
```

# Example (Two Hybrids)

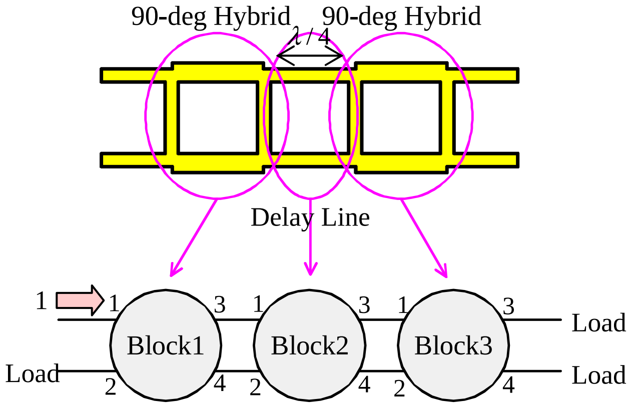
Cascade-connected two branch-line couplers (hybrids)



$$\begin{bmatrix} 0 & 0 & 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} \\ 0 & 0 & e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} & 0 & 0 \\ e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \end{bmatrix} \quad e^{-j\frac{\pi}{2}}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} \\ 0 & 0 & e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} & 0 & 0 \\ e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \end{bmatrix}$$

# Input file (1)

Needs two input files:
* Input file for S-matrix description
* Input file for topology description



$$\begin{bmatrix} 0 & 0 & 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} \\ 0 & 0 & e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} & 0 & 0 \\ e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \end{bmatrix} \quad e^{-j\frac{\pi}{2}}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} \\ 0 & 0 & e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} & 0 & 0 \\ e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \end{bmatrix}$$

Input file for S-matrix description
(s_matrix.dat)

```
"-------- GSM SOLVER (S-MATRIX INPUT FILE) --------"
"NO. OF BLOCKS", 3
"NO. OF PORTS IN BLOCK 1", 4
"S(1,1) [dB, deg]=", -100.0,    0.0
"S(1,2) [dB, deg]=", -100.0,    0.0
"S(1,3) [dB, deg]=", -3.0103,   0.0
"S(1,4) [dB, deg]=", -3.0103, -90.0
"S(2,1) [dB, deg]=", -100.0,    0.0
"S(2,2) [dB, deg]=", -100.0,    0.0
"S(2,3) [dB, deg]=", -3.0103, -90.0
"S(2,4) [dB, deg]=", -3.0103,   0.0
"S(3,1) [dB, deg]=", -3.0103,   0.0
"S(3,2) [dB, deg]=", -3.0103, -90.0
"S(3,3) [dB, deg]=", -100.0,    0.0
"S(3,4) [dB, deg]=", -100.0,    0.0
"S(4,1) [dB, deg]=", -3.0103, -90.0
"S(4,2) [dB, deg]=", -3.0103,   0.0
"S(4,3) [dB, deg]=", -100.0,    0.0
"S(4,4) [dB, deg]=", -100.0,    0.0
```

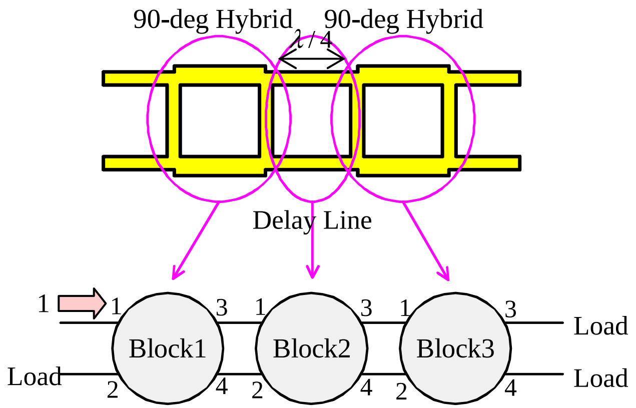
```
"NO. OF PORTS IN BLOCK 2", 4
"S(1,1) [dB, deg]=", -100.0,    0.0
"S(1,2) [dB, deg]=", -100.0,    0.0
"S(1,3) [dB, deg]=",    0.0,  -90.0
"S(1,4) [dB, deg]=", -100.0,    0.0
"S(2,1) [dB, deg]=", -100.0,    0.0
"S(2,2) [dB, deg]=", -100.0,    0.0
"S(2,3) [dB, deg]=", -100.0,    0.0
"S(2,4) [dB, deg]=",    0.0,  -90.0
"S(3,1) [dB, deg]=",    0.0,  -90.0
"S(3,2) [dB, deg]=", -100.0,    0.0
"S(3,3) [dB, deg]=", -100.0,    0.0
"S(3,4) [dB, deg]=", -100.0,    0.0
"S(4,1) [dB, deg]=", -100.0,    0.0
"S(4,2) [dB, deg]=",    0.0,  -90.0
"S(4,3) [dB, deg]=", -100.0,    0.0
"S(4,4) [dB, deg]=", -100.0,    0.0
```

```
"NO. OF PORTS IN BLOCK 3", 4
"S(1,1) [dB, deg]=", -100.0,    0.0
"S(1,2) [dB, deg]=", -100.0,    0.0
"S(1,3) [dB, deg]=", -3.0103,   0.0
"S(1,4) [dB, deg]=", -3.0103, -90.0
"S(2,1) [dB, deg]=", -100.0,    0.0
"S(2,2) [dB, deg]=", -100.0,    0.0
"S(2,3) [dB, deg]=", -3.0103, -90.0
"S(2,4) [dB, deg]=", -3.0103,   0.0
"S(3,1) [dB, deg]=", -3.0103,   0.0
"S(3,2) [dB, deg]=", -3.0103, -90.0
"S(3,3) [dB, deg]=", -100.0,    0.0
"S(3,4) [dB, deg]=", -100.0,    0.0
"S(4,1) [dB, deg]=", -3.0103, -90.0
"S(4,2) [dB, deg]=", -3.0103,   0.0
"S(4,3) [dB, deg]=", -100.0,    0.0
"S(4,4) [dB, deg]=", -100.0,    0.0
```

# Input file (2)

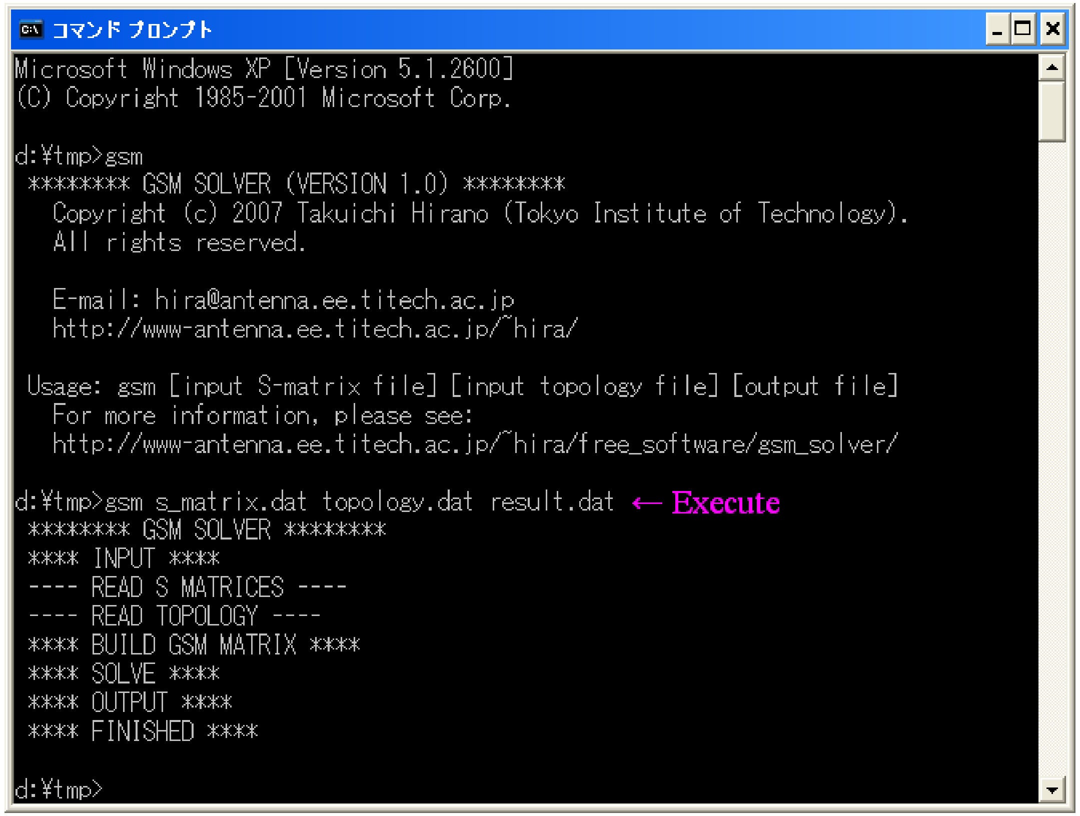
## Input file for topology description (topology.dat)

```
CM -------- GSM SOLVER (TOPOLOGY INPUT FILE) --------
CM CM; COMMENTS
CM CN BLOCKi PORTi BLOCKj PORTj; CONNECT
CM EX BLOCKi PORTi MAG[dB] PHA[deg]; EXCITE
CM LD BLOCKi PORTi; MATCHED LOAD
CM OP BLOCKi PORTi IN_OUT[1=IN, 2=OUT]; OUTPUT
CM ED; END
CN  1 3  2 1
CN  1 4  2 2
CN  2 3  3 1
CN  2 4  3 2
EX  1 1 0.0d0 0.0d0
LD 1 2
LD 3 3
LD 3 4
OP 1 1 2
OP 1 2 2
OP 3 3 2
OP 3 4 2
ED
```



$$\begin{bmatrix} 0 & 0 & 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} \\ 0 & 0 & e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} & 0 & 0 \\ e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \end{bmatrix} \quad e^{-j\frac{\pi}{2}}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} \\ 0 & 0 & e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & e^{-j\frac{\pi}{2}}/\sqrt{2} & 0 & 0 \\ e^{-j\frac{\pi}{2}}/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \end{bmatrix}$$

# Execute (Run)



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

d:¥tmp>gsm
 ******** GSM SOLVER (VERSION 1.0) ********
   Copyright (c) 2007 Takuichi Hirano (Tokyo Institute of Technology).
   All rights reserved.

   E-mail: hira@antenna.ee.titech.ac.jp
   http://www-antenna.ee.titech.ac.jp/~hira/

 Usage: gsm [input S-matrix file] [input topology file] [output file]
   For more information, please see:
   http://www-antenna.ee.titech.ac.jp/~hira/free_software/gsm_solver/

d:¥tmp>gsm s_matrix.dat topology.dat result.dat
 ******** GSM SOLVER ********
 **** INPUT ****
 ---- READ S MATRICES ----
 ---- READ TOPOLOGY ----
 **** BUILD GSM MATRIX ****
 **** SOLVE ****
 **** OUTPUT ****
 **** FINISHED ****

d:¥tmp>
```

← Execute

In console:

$$\text{gsm}\ \underbrace{\text{s\_matrix.dat}}_{\text{input file(1)}}\ \underbrace{\text{topology.dat}}_{\text{input file(2)}}\ \underbrace{\text{result.dat}}_{\text{output file}}$$

# Result (Output File)

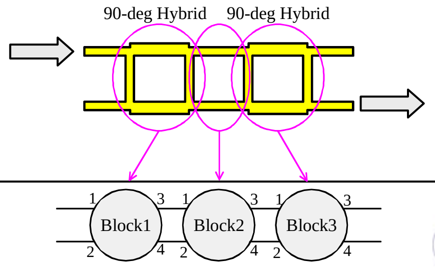
Output file (result.dat)

```
**** INPUT ****
---- READ S MATRICES ----
NO. OF TOTAL BLOCKS=          3
NO. OF TOTAL PORTS=          12
-- BLOCK           1 --
S(          1,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          3)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          4)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          2,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          3)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          2,          4)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          1)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          2)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          3,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          1)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          4,          2)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
```

```
-- BLOCK           2 --
S(          1,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          3)= 0.000000000000000E+000 [dB],
 -90.0000000000000       [deg]
S(          1,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          4)= 0.000000000000000E+000 [dB],
 -90.0000000000000       [deg]
S(          3,          1)= 0.000000000000000E+000 [dB],
 -90.0000000000000       [deg]
S(          3,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          2)= 0.000000000000000E+000 [dB],
 -90.0000000000000       [deg]
S(          4,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
```

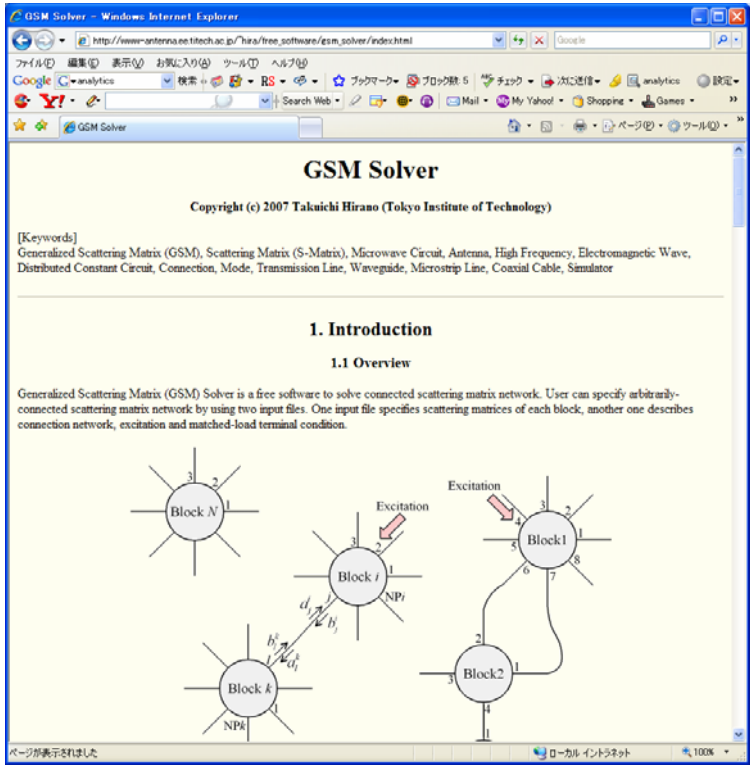
```
-- BLOCK           3 --
S(          1,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          3)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          1,          4)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          2,          1)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          2)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          2,          3)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          2,          4)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          1)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          2)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          3,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          3,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          1)= -3.01030000000000      [dB],
 -90.0000000000000       [deg]
S(          4,          2)= -3.01030000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          3)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
S(          4,          4)= -100.000000000000      [dB],
 0.000000000000000E+000 [deg]
---- END OF READ S MATRICES ----
```

```
---- READ TOPOLOGY ----
CM
CM
CM
CM
CM
CM
CN           1           3           2           1
CN           1           4           2           2
CN           2           3           3           1
CN           2           4           3           2
EX           1           1  0.000000000000000E+000
0.000000000000000E+000
LD           1           2
LD           3           3
LD           3           4
OP           1           1           2
OP           1           2           2
OP           3           3           2
OP           3           4           2
ED
---- END OF READ TOPOLOGY ----
NO. OF GSM UNKNOWNS=          20
**** BUILD GSM MATRIX ****
**** SOLVE ****
CONDITION NUMBER=   6.74204449940667
**** OUTPUT ****
(BLOCK, PORT, IN[1] OR OUT[2])=MAGNITUDE [dB],
PHASE [deg]
(           1,           1,           2)=
 -100.000173721609      [dB],   1.723036480736702E-008
[deg]
(           1,           2,           2)=
 -100.000173721464      [dB],   1.521479064812470E-008
[deg]
(           3,           3,           2)=
 -100.000000088057      [dB],  -90.0011459357343       [deg]
(           3,           4,           2)=
-8.498318098466007E-008 [dB],    180.000000000000
[deg]
**** FINISHED ****
```



Cascade-connected two branch-line couplers (hybrids) works as a cross-coupler.

# Distribution


GSM Solver - Windows Internet Explorer
http://www-antenna.ee.titech.ac.jp/~hira/free_software/gsm_solver/index.html
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Google analytics
Search Web
Mail
My Yahoo!
Shopping
Games
GSM Solver
ページ(P)
ツール(O)
ページが表示されました
ローカル イントラネット
100%

## GSM Solver

**Copyright (c) 2007 Takuichi Hirano (Tokyo Institute of Technology)**

[Keywords]
Generalized Scattering Matrix (GSM), Scattering Matrix (S-Matrix), Microwave Circuit, Antenna, High Frequency, Electromagnetic Wave, Distributed Constant Circuit, Connection, Mode, Transmission Line, Waveguide, Microstrip Line, Coaxial Cable, Simulator

## 1. Introduction

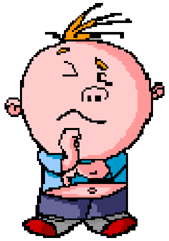
### 1.1 Overview

Generalized Scattering Matrix (GSM) Solver is a free software to solve connected scattering matrix network. User can specify arbitrarily-connected scattering matrix network by using two input files. One input file specifies scattering matrices of each block, another one describes connection network, excitation and matched-load terminal condition.

# Special Attentions for Release

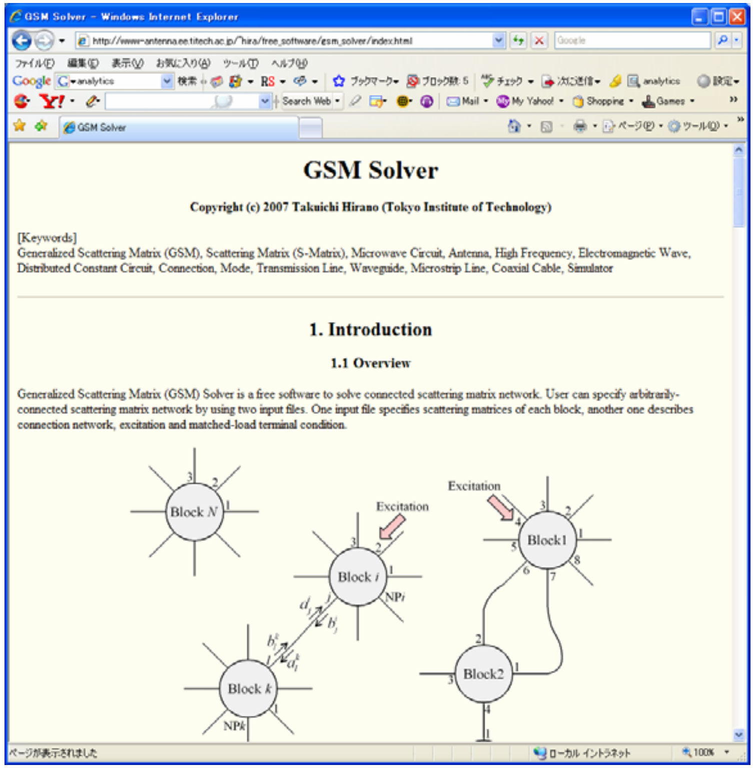
Fortran90

- Dynamic memory allocation (allocate, deallocate)
- Command-line parameter (nargs(), getarg)
- Error trap, protect buffer over flow (Protection from computer virus)
- Parameter check (User-friendly)
- Declaration of copyrights and agreement for use (Protection from lawsuit)



# URL

http://www-antenna.ee.titech.ac.jp/~hira/free_software/gsm_solver/


GSM Solver - Windows Internet Explorer
http://www-antenna.ee.titech.ac.jp/~hira/free_software/gsm_solver/index.html
Google
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Google analytics 検索 RS ブックマーク ブロック数: 5 チェック 次に送信 analytics 設定
Y! Search Web Mail My Yahoo! Shopping Games
GSM Solver
ページ(P) ツール(O)
ページが表示されました
ローカル イントラネット
100%

## GSM Solver

Copyright (c) 2007 Takuichi Hirano (Tokyo Institute of Technology)

[Keywords]
Generalized Scattering Matrix (GSM), Scattering Matrix (S-Matrix), Microwave Circuit, Antenna, High Frequency, Electromagnetic Wave, Distributed Constant Circuit, Connection, Mode, Transmission Line, Waveguide, Microstrip Line, Coaxial Cable, Simulator

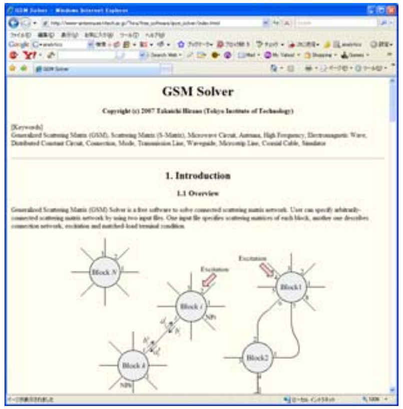
## 1. Introduction

### 1.1 Overview

Generalized Scattering Matrix (GSM) Solver is a free software to solve connected scattering matrix network. User can specify arbitrarily-connected scattering matrix network by using two input files. One input file specifies scattering matrices of each block, another one describes connection network, excitation and matched-load terminal condition.

- 1. Introduction
  - 1.1 Overview
- 2. Agreement on the Use of GSM Solver Software
- 3. Download (Free Software)
  - 3.1 Windows Console Application
  - 3.2 Source File (Fortran 90)
- 4 Manual
  - 4.1 Example
  - 4.2 Input Files
  - 4.3 Execute
  - 4.4 Other Samples
- 5. Technical Notes
- 6. Release Notes
- 7. Acknowledgement
- References

# Access Analysis



# GSM Solver

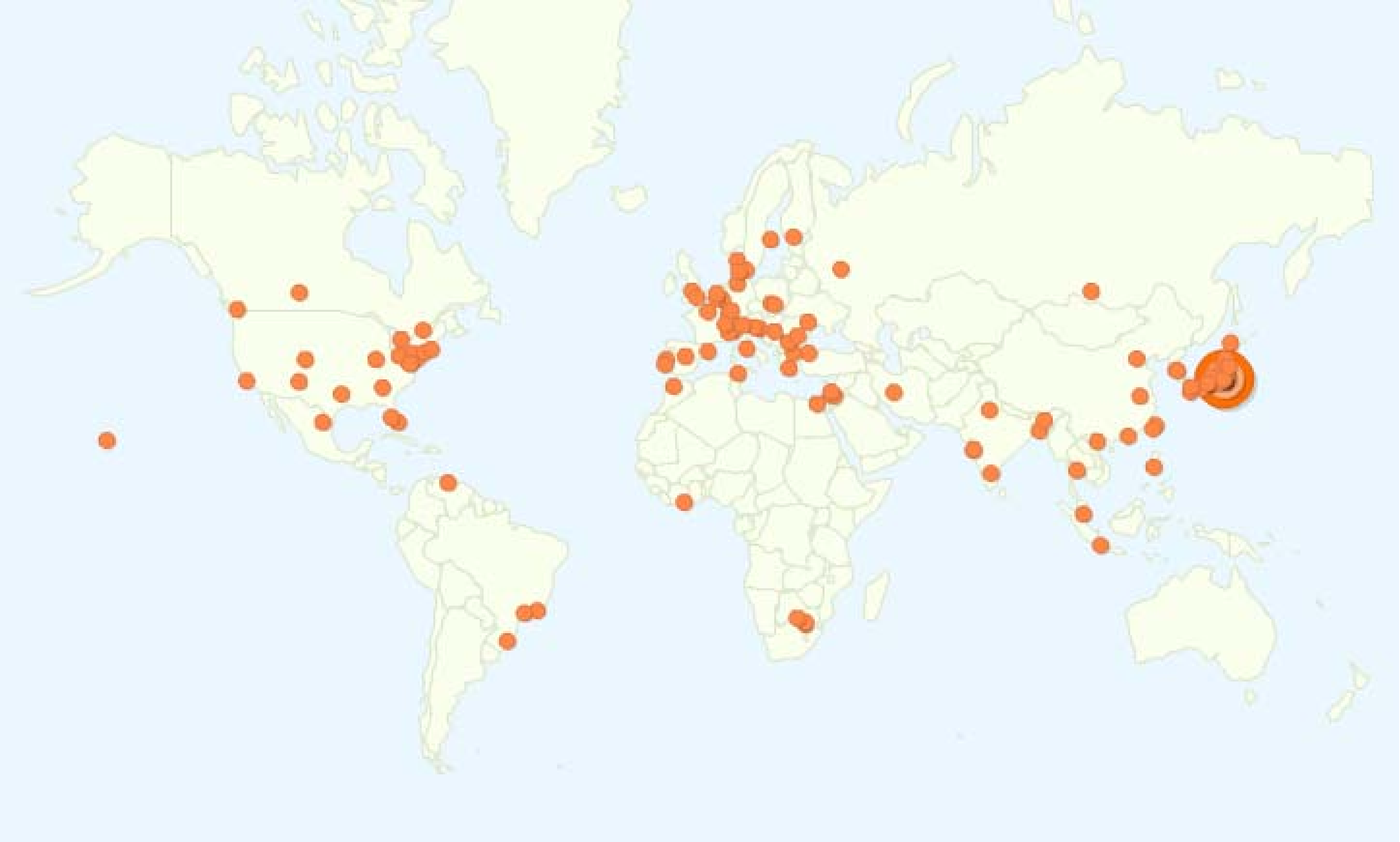
**Copyright (c) 2007 Takuichi Hirano (Tokyo Institute of Technology)**

[Keywords]
Generalized Scattering Matrix (GSM), Scattering Matrix (S-Matrix), Microwave Circuit, Antenna, High Frequency, Electromagnetic Wave, Distributed Constant Circuit, Connection, Mode, Transmission Line, Waveguide, Microstrip Line, Coaxial Cable, Simulator

## 1. Introduction

### 1.1 Overview

Generalized Scattering Matrix (GSM) Solver is a free software to solve connected scattering matrix network. User can specify arbitrarily-connected scattering matrix network by using two input files. One input file specifies scattering matrices of each block, another one describes connection network, excitation and matched-load terminal condition.



Powered By Google Analytics

2007.1.19-2007.6.27

# Summary

- Developed free software “GSM-Solver”.
- Special attention for release:
  - Program code (User friendly, Protection from computer virus)
  - Law (Provision for lawsuit)