USER GUIDE

Research group of Prof. Xinzheng Lu
Tsinghua University

1. Installation

The installation requirement is the same with the SPEED program. Please refer to the following link for more detailed information.
http://speed.mox.polimi.it/DOWNLOAD/DOXYGEN_NEW/html/page1.html

2. Soil model

Please prepare soil models following the user guide of SPEED in the following link:
http://speed.mox.polimi.it/DOWNLOAD/DOXYGEN_NEW/html/page2.html.
The command of the program is basically the same with SPEED program ***except for the following 4 points***.

1. Two additional function types are defined

(1) FUNC NF 777 1 1 1

NF: ID number of the function used for distinguishing different functions
TYPE: 777
Dummy: 1 1 1
When this function is adopted, the reaction force from buildings will be used for corresponding commands. (See the following command for NEUX and PLOX)

(2) FUNC NF 773 Nt Filename

NF: ID number of the function used for distinguishing different functions
TYPE: 773
Filename: The name of the file containing records.
Nt: The size of records written in "Filename".
This type is very similar to the existing type "3". Users need to provide time series in a file named "Filename". The file must be formatted in the following form:
Nt
Val1
Val2
...
ValNt

Attention: The **Val*n*** in the file will be adopted directly at the ***n***th time step. No interpolation procedure will be adopted when this type of function is used, so that the computing time will be greatly reduced. But the users have to make sure that the data in the file have been well prepared.

For example, if the total time is 2 s and the time step for analysis is 0.01, then ***Nt*** should be larger than 2/0.01+1=201, and the first 201 data will be used directly in this analysis.
Also, when the PLAX/PLAY/PLAZ command is adopted, the input records in the file should be the velocity time series.

2. LS.INPUT

Same with the monitor file in SPEED, this file contains a list of monitored points for which the solution is written in output. But the difference is that, in this program, the building positions must be written in this file firstly.
The list of monitored points must be written in the following format:

Nmonitor
1 BX(1) BY(1) BZ(1)
...
Nbuilding BX(Nbuilding) BY(Nbuilding) BZ(Nbuilding)
Nbuilding+1 X Y Z
...

Nmonitor: Total number of monitored points.
Nbuilding: Total number of buildings.
BX/BY/BZ(*i*): The building position coordinate of the ***i***th Building

3. NEUX/ NEUY/ NEUZ/ NEUN

NEUX BLOCKID NF VAL1 VAL2 VAL3 VAL4 BID BDir
(1) This command assigns Neumann boundary conditions to a group (BLOCKID) of surface (quad) elements by using the reaction force from buildings. The load may be applied in the ***X***, ***Y***, ***Z*** or ***Normal*** direction. If the type of function ***NF*** is 777, the load is specified as the product of a distributed load factor varying from ***VAL1*** to ***VAL4***, times the reaction force obtained from the building with ID of ***BID*** along the direction of ***BDir***. Otherwise, if the type of function ***NF*** is not 777, the load is specified as the product of a time function of index ***NF***, times a distributed load varying from ***VAL1 to VAL4***, and also please define ***BID*** and ***BDir*** as 0 in this condition.
(2) **BLOCKID**: block id number defined and exported from the mesh generator
(3) **NF**: number of function associated with this load
(4) **VALi**: value of Neumann load at vertex i of the element
(5) **BID**: ID of the building where the reaction force is obtained
(6) **BDir**: The direction of the reaction force. Use 1 for *X*, or 2 for *Y*

4. PLOX/PLOY/PLOZ

PLOX BLOCKID NF X Y Z VAL BID BDir
(1) This command assigns Neumann type point load to the spectral node nearest to a specified ***X***, ***Y***, ***Z*** position by using the reaction force from buildings. The load may be applied in the ***X***, ***Y***, or ***Z***

direction. If the type of function ***NF*** is 777, the load is specified as the product of a load factor ***VAL***, times the reaction force obtained from the building with ID of ***BID*** along the direction of ***BDir***. Otherwise, if the type of function ***NF*** is not 777, the load is specified as the product of a time function of index ***NF***, times a point load ***VAL***, and also please define ***BID*** and ***BDir*** as 0 in this condition.

(2) **BLOCKID**: block id number defined and exported from the mesh generator

(3) **NF**: number of function associated with this load

(4) **VALi**: value of Neumann load at vertex i of the element

(5) **BID**: ID of the building where the reaction force is obtained

(6) **BDir**: The direction of the reaction force. Use 1 for *X*, or 2 for *Y*

Attention1: If more than 1 point load along the same direction are applied to the same point, please use different ***NF***. For example, the following commands are correct:

PLOX ***1*** *785.2755 -625.6388 525.4000 0.666667E-01 1 1*

PLOX ***2*** *785.2755 -625.6388 525.4000 0.666667E-01 5 1*

The force cannot be applied together if the commands are used as follows:

PLOX ***1*** *785.2755 -625.6388 525.4000 0.666667E-01 1 1*

PLOX ***1*** *785.2755 -625.6388 525.4000 0.666667E-01 5 1*

Attention2: If no PLOX/Y/Z or NEUX/NEUY/NEUZ is defined, then the soil model is calculated as a free field condition, since no reaction force from buildings is imposed to the soil model. Thus, no SCI effects would be considered during the computation. Also, in this condition, the building behavior is calculated based on corresponding free field motions obtained from the building positions defined in **LS.INPUT**.

3. Config file

A file named “Config.txt” is in need, and the file should be formatted in the following form:

Np
Damping, Massarea
DampingType

Np: The number of total MPI cores. This should be the same with the value adopted to run SPEED-MDOF.exe

Damping: The damping ratio for building models.

Massarea: The mass per area of each story for building models

DampingType: If this value is 0, the damping type is only determined by the command of “**DAMPING**” in “**SPEED.input**”. If this value is 1, a stiffness damping will be added together with the mass damping defined in “**SPEED.input**”. Here, we strongly recommend the users to define “**DAMPING 1**” in “**SPEED.input**”.

4. Building model

The building information should be contained in a file named “BuildingInfo.txt”. The file should be

formatted in the following form:

NB
ID1 TYPE NST SH AREA
Props(1)
...
Props(NST)
ID2 TYPE NST SH AREA
....
IDn TYPE NST SH AREA
Props(1)
...
Props(NST)
ID1
DS(1)
...
DS(NST)
ID2
...
IDn
DS(1)
...
DS(NST)

NB: Total number of buildings.
IDi: Building ID
TYPE: Building type. Dummy integer.
NST: Story number.
SH: Story height.
AREA: Story area.
Props: This array contains 10 parameters determining the IDR-force relations, including:

Props(1): Initial tangent stiffness.
Props(2): Initial yield stress.
Props(3): Hardening stiffness ratio.
Props(4): Parameter for cumulative damage. (No cumulative damage is considered when this value is positive infinity)
Props(5): Pinching factor. (0 for very severe pinching; 1 for no pinching)
Props(6): Softening stiffness ratio
Props(7): The ratio of peak strength over yielding strength
Props(8): The ratio of the yield strength between the negative and positive loading direction
Props(9): Parameter for unloading stiffness. (0 if the unloading stiffness equals initial stiffness; 1 if unloading path pointing to 0)
Props(10): Crack closure position.
We recommend users to use **createSAMSCI.exe**, provided in the following part, to generate these parameters

DS：This array contains 4 parameters determining thresholds for slight, moderate, severe and complete damage.

Buildings model can be generated using the following program:

(1) createSAMSCI.exe

This program is modified based on the SimCenter Workflow to generate *BuildingInfo.txt* file.

The input files must be named in this format: *1-BIM.json, 2-BIM.json, , , i-BIM.json*

Note: The *data* folder is necessary to determine the parameters of building. In addition, the applications require the jansson lib (*libjansson-4.dll*, *libwinpthread-1.dll*) be installed.

(2) BuildingPreShear.exe

Put this program together with *BuildingInfo.txt*. Run this program to generate the required files for building to run the SPEED-MDOF.

5. Computation

1. Please use the BRCtrl.txt in the ***example file*** for each case, so that useful results can be output.
2. Please use ***run.sh*** in the ***example file*** to run SPEED-MDOF.exe. In this file, users need to input the number of total MPI cores, which should be the same with the ***NP*** in ***Config.txt***.

6. Post-process of results

To adopt the results of SPEED-MDOF in the SimCenter workflow, the users need to use PostprocessSpeed_MDOF.exe to convert the result files.

Put this program in the same directory of the *SPEED-MDOF.exe* and double click this program, the EDP file of each building can be obtained.

Note: the applications require the jansson lib (*libjansson-4.dll*, *libwinpthread-1.dll*) be installed.

Note: The source code of above programs can be found in *SRC* folder.