NONLINEAR ANALYSIS OF PLATE DEFLECTION ACCOUNTING FOR BOTH INELASTIC FRACTURE AND PLASTIC YIELDING IN REINFORCED CONCRETE SLABS

THEORETICAL FOUNDATIONS AND USER'S MANUAL

Kolbjorn Saether

Kolbjorn Saether & Associates
1062 Chicago Avenue
Chicago, IL 60606

1.0 Theoretical Formulation

The solution algorithm developed herein has been specifically formulated to model the inelastic fracture and plastic yielding exhibited by steel reinforced concrete slabs with applied normal pressure loads to simulate body forces generated by self weight and by assumed service loading. The slab domain is further assumed to be contained within a rectangular region that contains a variety of column support configurations. The Finite Difference Method was selected to solve the governing field equations for slab deflection in favor of using the Finite Element Method primarily because of the regularity of the slab geometry. For regular domains in which difference methods can be applied, the analysis involves the direct solution to the governing equilibrium equations which in a point-wise fashion over the domain. In contrast, the finite element method is ideal for complicated geometric configuration but has the computational disadvantage that the basic element formulation is based on minimizing the energy of a functional which is an integral measure of the elastic energy stored within an element assemblage. The convergence of the finite element method is thus based on integral measures whereas the finite difference method permits a point-wise satisfaction of the direct equilibrium equations.

A regular nine-column slab configuration is selected for the present analysis. A schematic of this configuration is depicted below.

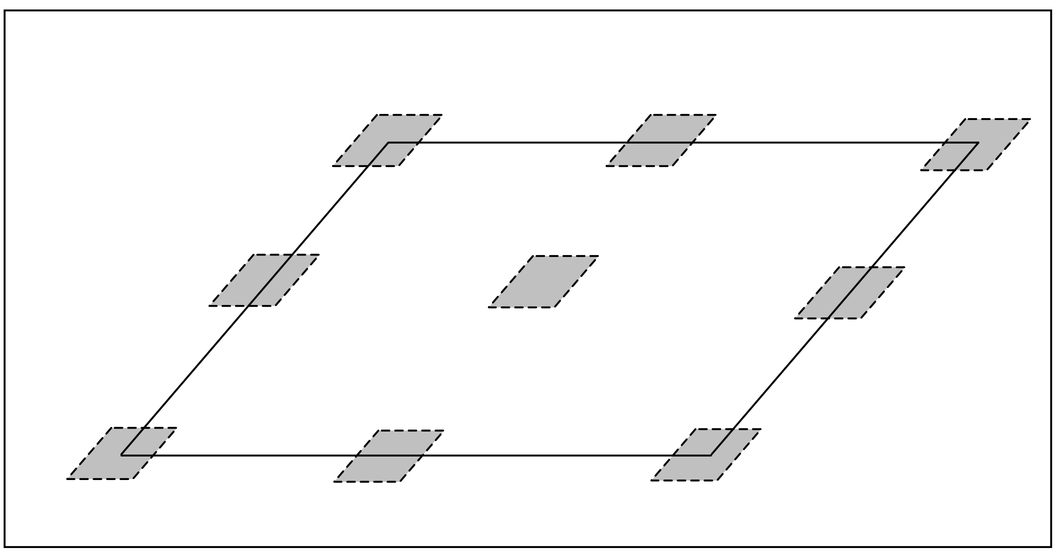


Figure 1. Nine column slab configuration.

The analysis developed herein is based on a Kirchoff plate formulation which has been enhanced to account for embedded reinforcing steel using classical lamination theory. The governing plate equilibrium equations are shown below in equations 1.

$$\frac{\partial N_{XX}}{\partial x} + \frac{\partial N_{XY}}{\partial y} = 0$$

$$\frac{\partial N_{XY}}{\partial x} + \frac{\partial N_{Y}}{\partial y} = 0 \qquad (1)$$

$$\frac{\partial^2 M_{XX}}{\partial x^2} + 2\frac{\partial^2 M_{XY}}{\partial x\,\partial y} + \frac{\partial^2 M_{YY}}{\partial y^2} = -P_{ZZ}$$

Applying classical lamination theory yields the following equilibrium equations governing the inplane stretching and normal deflection of a plate with a nonuniform distribution of material properties through the plate thickness.

$$A_{11}\frac{\partial^2 u}{\partial x^2} + 2A_{16}\frac{\partial^2 u}{\partial x \partial y} + A_{66}\frac{\partial^2 u}{\partial y^2} + A_{16}\frac{\partial^2 v}{\partial x^2} + \left(A_{12} + A_{66}\right)\frac{\partial^2 v}{\partial x \partial y} + A_{26}\frac{\partial^2 v}{\partial y^2} - B_{11}\frac{\partial^3 w}{\partial x^3} -$$
$$3B_{16}\frac{\partial^3 w}{\partial x^2 \partial y} - \left(B_{12} + 2B_{66}\right)\frac{\partial^3 w}{\partial x \partial y^2} - B_{26}\frac{\partial^3 w}{\partial y^3} = 0$$

$$A_{16}\frac{\partial^2 u}{\partial x^2} + \left(A_{12} + A_{66}\right)\frac{\partial^2 u}{\partial x \partial y} + A_{26}\frac{\partial^2 u}{\partial y^2} + A_{66}\frac{\partial^2 v}{\partial x^2} + 2A_{26}\frac{\partial^2 v}{\partial x \partial y} + A_{22}\frac{\partial^2 v}{\partial y^2} - B_{16}\frac{\partial^3 w}{\partial x^3} -$$
$$\left(B_{12} + 2B_{66}\right)\frac{\partial^3 w}{\partial x^2 \partial y} - 3B_{26}\frac{\partial^3 w}{\partial x \partial y^2} - B_{22}\frac{\partial^3 w}{\partial y^3} = 0 \qquad (2)$$

$$D_{11}\frac{\partial^4 w}{\partial x^4} + 4D_{16}\frac{\partial^4 w}{\partial x^3 \partial y} + 2\left(D_{12} + 2D_{66}\right)\frac{\partial^4 w}{\partial x^2 \partial y^2} + 4D_{26}\frac{\partial^4 w}{\partial x \partial y^3} + D_{22}\frac{\partial^4 w}{\partial y^4} - B_{11}\frac{\partial^3 u}{\partial x^3} - 3B_{16}\frac{\partial^3 u}{\partial x^2 \partial y} -$$
$$\left(B_{12} + 2B_{66}\right)\frac{\partial^3 u}{\partial x \partial y^2} - B_{26}\frac{\partial^3 u}{\partial y^3} - B_{16}\frac{\partial^3 v}{\partial x^3} - \left(B_{12} + 2B_{66}\right)\frac{\partial^3 v}{\partial x^2 \partial y} - 3B_{26}\frac{\partial^3 v}{\partial x \partial y^2} - B_{22}\frac{\partial^3 v}{\partial y^3} = 0$$

The slab cross-section is assumed to composed of discrete layers which contain different material and stiffness. This is depicted below in Figure 3.

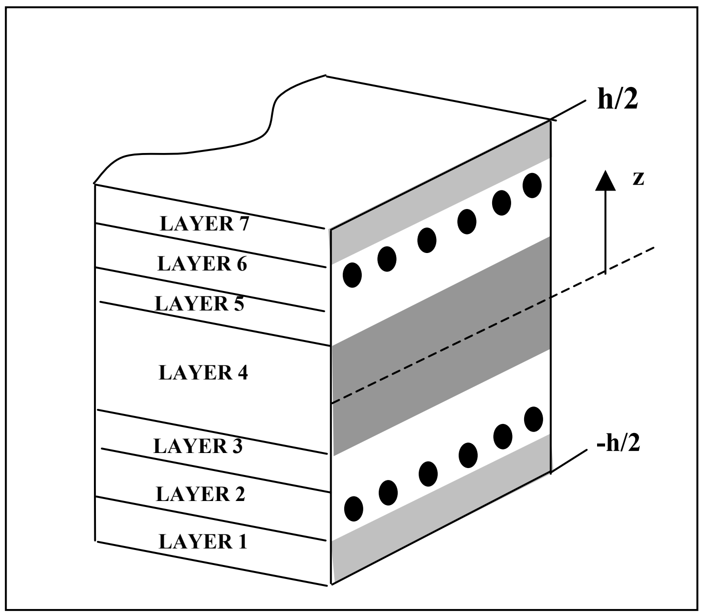


Figure 3. Layered slab cross section.

Classical lamination theory provides measures for inplane, membrane-bending coupling, and bending stiffness terms given by:

$$A_{ij} = \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k - z_{k-1}\right)$$

$$B_{ij} = \frac{1}{2}\sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k^2 - z_{k-1}^2\right) \qquad (3)$$

$$D_{ij} = \frac{1}{3}\sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k^3 - z_{k-1}^3\right)$$

where Q_{IJ} is equal to the reduced material stiffnesses for the k^{th} layer and z_k is the distance from the plate neutral surface to the k^{th} layer interface.

1.1 Nonlinear Material Behavior

The nonlinear behavior of the slab constituent materials is shown below in Figure 4. Concrete is assumed to exhibit simple inelastic fracture in which, after a critical load is reached, all load carrying capability is lost. This is shown below for the load-deflection response of a single layer and the response of the plate at some point due to distributed failures of multiple concrete layers. During load redistribution caused by failure, no regain in concrete properties is possible due to unloading. Reinforcing steel is assumed to fail in plastic yielding. The plasticity model assumed for steel is that of an elastic-perfectly plastic material such that after the yield point is reached, no further load may be carried by the steel. During unloading, the steel regains its initial elastic modulus but exhibits permanent plastic deformation. This behavior of steel, for an isolated reinforcing bar and for distributed yielding is also shown in Figure 4.

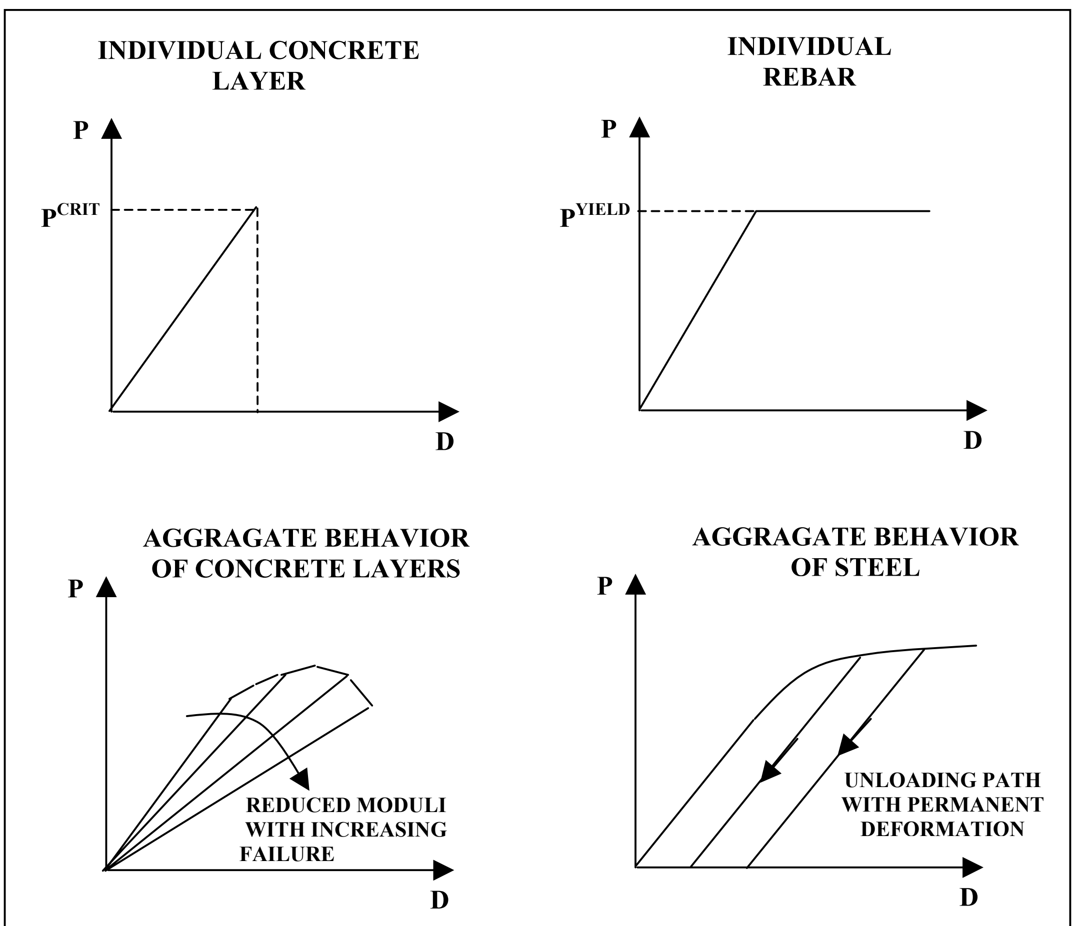


Figure 4. Assumed nonlinear behavior of concrete and steel material.

1.2 Nonlinear Solution Algorithm

In order to account for both inelastic fracture caused by rupture of the concrete and yielding of the reinforcing steel, the nonlinear solution algorithm is divided into two stages. The first stage performs a simple iteration to converge the majority of concrete layer fractures. During this stage no yielding of steel is calculated. The iterations are performed by applying the full input load, assessing layer ruptures, and repeating the analysis using the full load to account for additional layer cracking due to load redistribution. The next phase combines an iterative/incremental solution procedure in which the load is applied in increments and both yielding and cracking are accounted. Additional concrete cracking is expected to occur due to load redistribution of the reinforcing steel. If the additional layer cracking exceeds an input tolerance, say 1% of the total layers, the solution is deemed to have drifted far enough off the true equilibrium path and a new iteration is performed by restarting the incremental load solution from zero. A final set of single-pass solutions are obtained to separate out the influence of cracking and yielding on the final solution. Prior to beginning the nonlinear solution sequence a single pass is performed to obtain the elastic deflection solution, U_{ELAS} . The final nonlinear solution accounting for both cracking and yielding is stored as U_{TOTAL}. A final solution is calculated by considering only the reduced material properties due to cracking to obtain U_{FRACT} . The individual contributions are then given by

Component due to elastic deflection: $U_{ED} = U_{ELAS}$
Component due to concrete cracking: $U_{CRACK} = U_{FRACT} - U_{ELAS}$
Component due to rebar yielding: $U_{YIELD} = U_{TOTAL} - U_{CRACK} - U_{ED}$

2.0 User's Manual for Program *Deflect.f*

The first consideration in utilizing the *deflect.f* code is fixing the size of the various arrays used in the program which are required by the Fortran 77 programming used to create the program. The parameter statements are set at the beginning of the program and are listed as

```
      PARAMETER ( MXD1 = 1 ) : set to a minimum value
      PARAMETER ( MXBD = 1 ) : set to a minimum value
      PARAMETER ( MXD2 = 3500 )
      PARAMETER ( MXD3 = 1000 )
      PARAMETER ( MXD4 = 550  )
      PARAMETER ( MXD5 = 20   )
C
      DIMENSION GBLSYS(MXD1,MXBD),RHS(MXD1),TMP1(MXD1),NDIAG(25)
      DIMENSION SOL(MXD1),INDEX(MXD1),NEQCL(MXD2,3),NEQSE(MXD2,3)
```

The DIMENSION statements following these parameter definitions show how they are used to fix the size of various arrays used in the program, the largest being the global stiffness matrix named GBLSYS(MXD1,MXBD). To minimize the size of the required arrays, the primary parameters, MXD1 and MXBD, can be set to a small number – such as unity – and the code compiled with any Fortran compiler and executed. The code will automatically determine the deficiency in matrix sizing and output suggested sizes in the output file. This is convenient to minimize the array sizes for optimal execution speed. For repeated use, reasonable sizes should be specified in the PARAMETER statements that only need to be changed in the case when a large number of finite difference points are to be specified. The *deflect.f* program will also detect insufficient sizing for the MXD2 parameter. The above will require that the user be familiar with editing the source code and executing a Fortran compiler to generate a new executable.

The initial execution of the *Deflect.f* program will prompt the user for a *filename* to be used for input and output. For the input file, the following commands are used to input geometry, material properties, loading, and solution control for performing an analysis of slab deflection.

Table 1. *Deflect.F* input statements.

STATEMENT	FUNCTION	PAGE
*SOLUTION *YIELDING SOLUTION *CRACKING SOLUTION *COUPLING *BOUNDARY DERIVATIVES *DISCRETIZATION PARAMETERS	SOLUTION CONTROL	
*SLAB DIMENSION *UNIFORM STRESS *COLUMN DIMENSIONS *BOUNDARY *NORMAL PRESSURE *DEFAULT PRESSURE *MATERIAL *DAMAGE LAW *LAYER *ZONE *ZONE OVERLAP	MODEL INPUT	
*FILE PRINT *GRAPHICS *Diagnostic	OUTPUT CONTROL	
*HEADING *STATUS *ENDDATA *DEBUG	MISCELLANIOUS	

2.1 Solution Control

Two basic types of analyses may be selected using the following command:

```
*SOLUTION,  STYPE = STRING
 ITLIMC, ITLIMY, NUMINC, TOLC
```

where **`STRING`** may be set to either '**`LINEAR STATIC`**' or '**`NONLINEAR STATIC`**' depending on whether a linear or nonlinear analysis is to be performed. For a linear analysis, the data line containing input parameters is currently not used. For nonlinear analysis, the solution will account for both inelastic fracture in concrete and plastic yielding in steel. The necessary parameters on the data line are **`ITLIMC`** which specifies the maximum number of iterations allowed to converge concrete cracking, **`ITLIMY`** specifies the maximum number of iterations to be performed while accounting for yielding in the reinforcing steel, **`NUMINC`** is the number of load increments used in the solution for plastic yielding, and **TOLC** is the convergence tolerance for concrete cracking based on the ratio of accumulated layer failures to the total number of layers in the slab. Note: due to internal processing of failed layers, in the case where **TOLC** is large and the slab exhibits significant concrete cracking without yielding of steel, the cracking solution can be output with greater maximum deflections than the output of the maximum deflection due to combined cracking and steel yielding. This can be minimized by decreasing the value of **TOLC.**

The solution may be modified to calculate only yielding in the steel with the following command:

```
*YIELDING SOLUTION
```

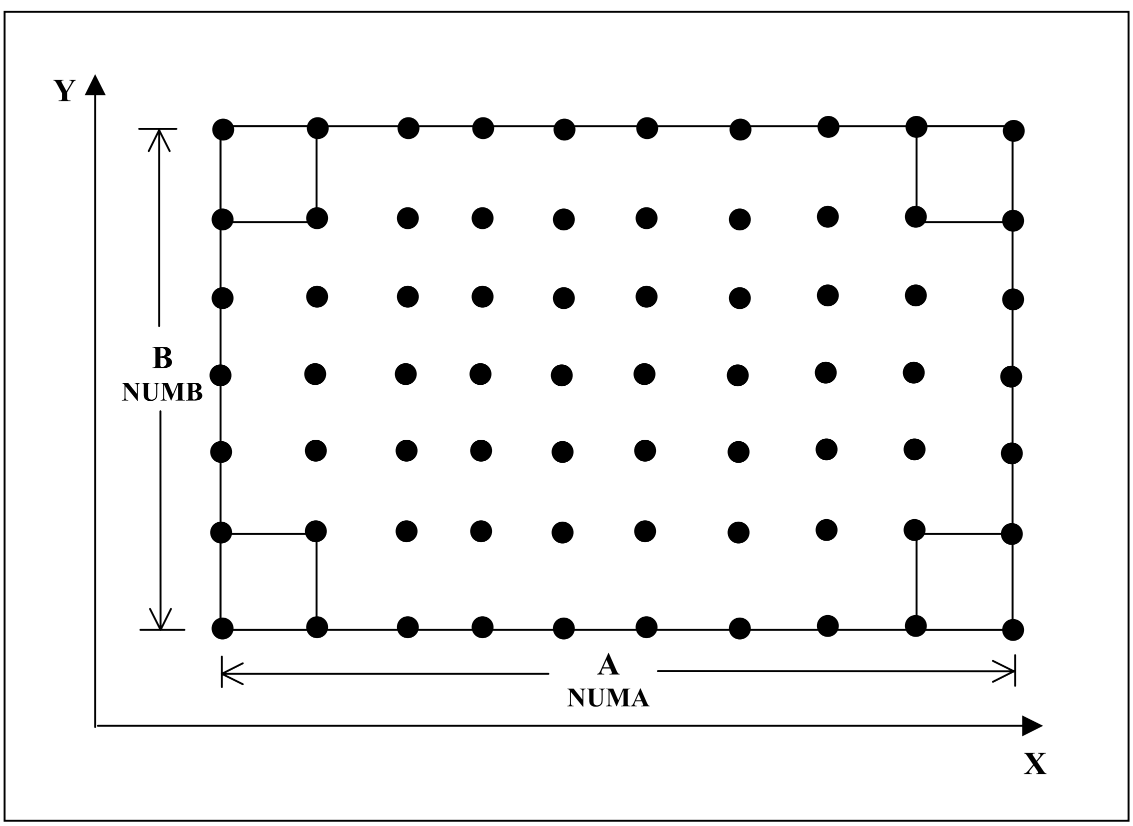
Or, one can specify that only concrete cracking be accounted for by inputting the following command:

```
*CRACKING SOLUTION
```

The resolution of the solution is controlled by specifying the number of finite difference points to be used in the global X and Y directions. Also, the number of difference lengths used to model a quarter column is input as a parameter. The following statement is used to specify the number of divisions to be made in the model

```
*DISCRETIZATION PARAMETERS
 NUMA, NUMB, ND
```

Where **NUMA** and **NUMB** are the number of difference points in the X and Y directions, respectively, and **ND** is the number of difference points used to model a side of a quarter column. This is shown in Figures 5 and 6.



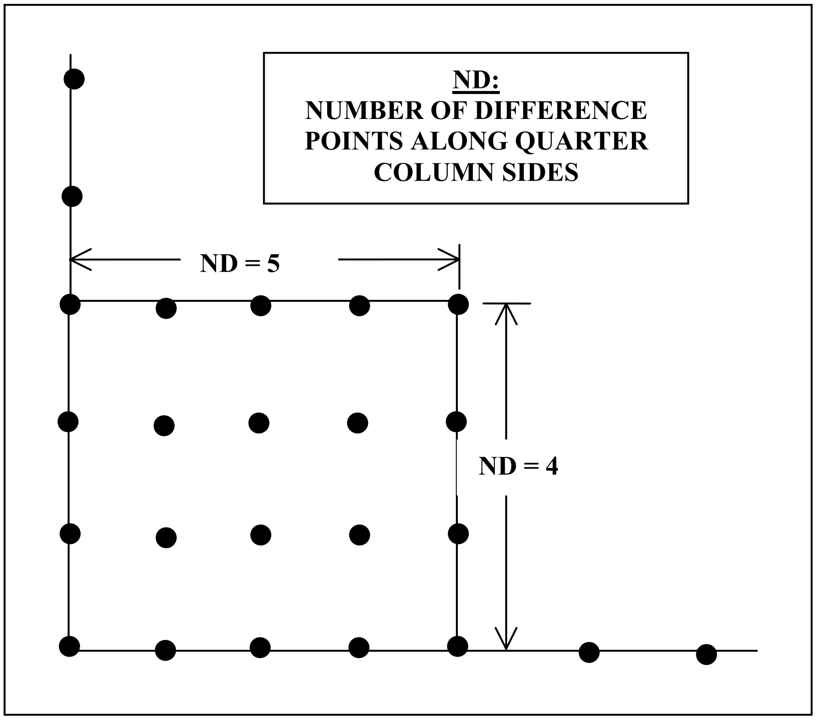


Figure 6. Definition of the **ND** parameter at column boundaries.

Thus, the ND parameter sets the size of the quarter columns. For example, in the X direction, if the length is 10 feet and the number of divisions is 20, then each spacing between finite difference points is 10/(20-1) or 0.526 feet. If ND is specified to be 2, then the quarter column dimension in the X-direction is equal to one spacing unit of 0.526 feet.

Note: If this restriction on column size is unacceptable, a more general column size and freedom in selecting the number of finite difference points across the slab can be made by not using the library boundary conditions and applying user-defined boundary conditions instead. Applying user-defined boundary conditions permits an arbitrary number of difference points in the x and y-directions to define clamped conditions representing the column cross-section.

The complete set of coupled plate equations describing U, V, and W displacements have been implemented into *Deflect.F* (old program version name). However, if the slab cross sections are relatively balanced, the coupling between lateral and in-plane deflections may be ignored. In this case only the uncoupled bending equation needs to be solved because the equations for in-plane stretching will only compute zero quantities. The following statement may be used to dictate whether coupling needs to be considered in calculating plate deflections

```
*COUPLYING = YES/NO
```

Specifying **coupling** = **no** causes only W-deflections to be calculated which reduced the size of the internal global array by two thirds and results in shorter run-times. Currently there is some indication that the membrane stiffness is *not* properly implemented and only W-deflections should be specified.

Specific treatments for calculating derivatives along the external boundary may be specified using the following statement:

```
*BOUNDARY DERIVATIVES, METH = KTYPE
```

where **METH** equals '**CENTRAL**' or '**MIXED**' to specify central differences or a mixture of forward, backward, and central differences to be used along the boundary. This treatment has demonstrated a large effect on recovered stresses and has not been completely certified as to which method is more accurate.

2.2 Model Description

The basic rectangular slab dimensions are input through the following statement:

```
*SLAB DIMENSIONS
  A, B
```

Where **A** and **B** are the slab length and width dimensions, respectively.

Column dimensions are determined by the input provided by the ***DISCRETIZATION PARAMETER** input statement or by the clamped conditions at finite difference points when using User-Defined boundary conditions. If this parameter is input using 'library' boundary conditions containing columns, an internal check is made regarding the accuracy of this specification using the actual column cross-section dimensions. These dimensions are input using the following statement:

```
*COLUMN DIMENSIONS
```

```
AC, BC
```

Where **AC** and **BC** are the column length and width. This statement currently has no effect on the solution; the column quarter dimensions are set by the inputted slab dimensions and the **ND** parameter in the ***DISCRETIZATION PARAMETERS** statement. It is advised that, in most cases, this statement should probably be neglected in the input; the actual effective column dimensions are written to the output file regardless of this statement.

The essential boundary conditions and basic slab layout are specified using the following input statement:

```
*BOUNDARY, TYPE = LIRBRARY or USER-DEFINED
 PARAMETER(S)
```

If **TYPE** is equal to '**LIBRARY**', the following slab boundary configurations have been built into the program and can be applied using the following character string identifiers on the **PARAMETER** line:

```
PARAMETER = '9-COLUMN'
            'CLAMPED'
            '4-COLUMN FREE
            '4-COLUMN SYMMETRY
            'SSCF'
            'SFSF'
            'SIMPLE'
```

A depiction of these archived boundary conditions is shown in Figure 7.

For greatest generality, **TYPE** can be set equal to '**USER-DEFINED**' in which completely arbitrary boundary conditions can be input. The required input on the parameter line, which is repeated to specify the complete set of boundary conditions, is given by:

```
PARAMETERS = I, J, BTPE, NCODE
```

Where **I**, **J** are the indices for the differnce point located **J** divisions in the X direction and **I** divisions in the Y direction with the point (1,1) assumed at the lower left hand corner of the model and **BTYPE** specifies the type of boundary condition given by:

BTYPE = **CL**: Clamped condition
SS: Simple Supports
FE: Free Edge
FS: Free Edge Symmetry

The parameter **NCODE** specifies which edge the conditions are being applies on. This code is given by:

NCODE = 1 for bottom boundary
2 for left boundary
3 for top boundary
4 for right boundary

A top view of a rectangular slab showing the **NCODE** is shown in Figure 8.

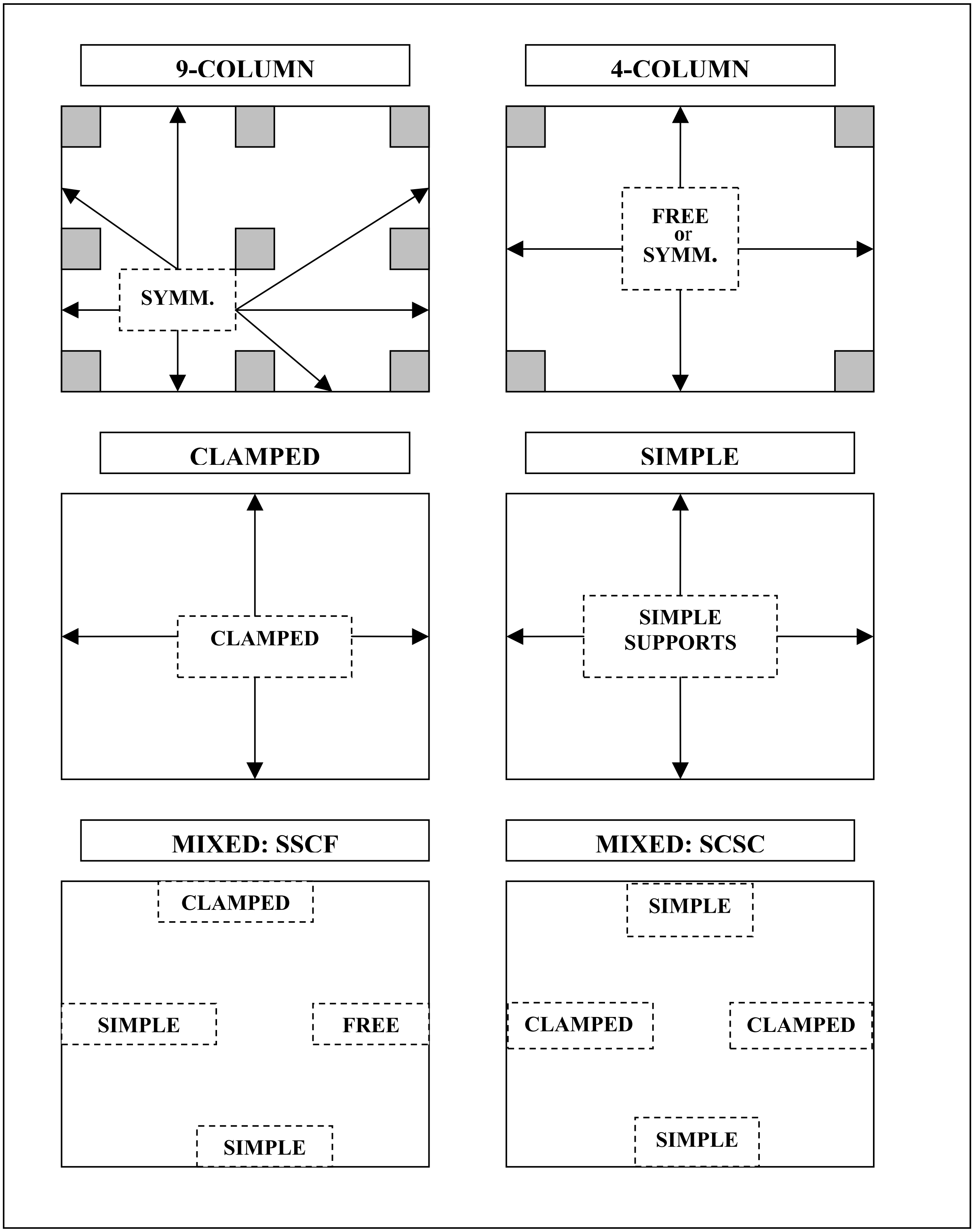


Figure 7. Library of slab configurations.

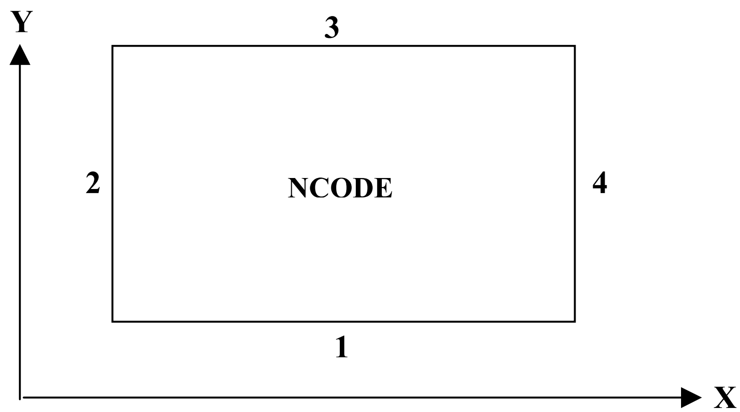


Figure 8. Definition of code for boundary designation

Loads are applied in the form of normal pressures applied to the slab upper surface. Several options are available in specifying applied loads. The general form of the input statement is given by

```
*NORMAL PRESSURE, NLOAD = N
 LOAD CASE, LOAD TYPE, MAGNITUDE, ZONE
```

The parameter **NLOAD** is set to the total number of different applied load conditions. For linear analysis multiple load cases are treated as independent load sets and are run sequentially. In nonlinear analysis, each load case is treated as a load step and results are added during execution. The first entry on the data line, **LOAD CASE**, is simply set to the number of the load case being input. Because multiple entries will normally be required to specify the load pattern across the slab, this entry is used to group slab loads into separate load sets. Several different types of loads may be specified using the **LOAD TYPE** parameter. This entry may assume the values:

```
LOAD TYPE = 'UNIFORM'
            'LIVE LOAD'
            'DEAD LOAD'
            'USER-DEFINED'
```

The load type **UNIFORM** specifies that the inputted load is to be applied uniformly over the entire slab. This avoids internal processing to determine which zone this load is associated with; a nonzero entry for the zone identification number must be entered but this number can be arbitrary as it is not used internally. The load types **LIVE LOAD** and **DEAD LOAD** are currently treated identically, they may be used to add readability to the input file to show which loads are associated with material self weight and those with service loads. The **USER-DEFINED** option directs DEFLECT.F to call to a subroutine supplied by the user that is accessed with an X and Y coordinate and returns a load magnitude. This gives the user the capability to code any possible type of applied loading such as ramped, sinusoidal, or discontinuous pressure distributions. An example of this subroutine is shown below.

```
      SUBROUTINE UPRESS ( SA,SB,XI,YI,PI,AMAG,NZONE,PX,PY,PZ )
C
C     ***********************************************
C     **                                           **
C     **  USER-DEFINED NORMAL PRESSURE DISTRIBUTION  **
C     **                                           **
C     ***********************************************
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C <<< SET EDGE PRESSURES EQUAL TO ZERO >>>
C
      PX = 0.0
      PY = 0.0
C
C <<< EXAMPLE 1: SINUSOIDAL VARYING NORMAL PRESSURE >>>
C
C      PZ = AMAG*SIN(XI*PI/SA)*SIN(YI*PI/SB)
C
C <<< EXAMPLE 2: DIFFERENT UNIFORM LOADS IN SPECIFIED REGIONS >>>
C
C      IF ( XI .LT. 5.0 .AND. YI .LT. 5.0 ) PZ = 5.0*AMAG
C      IF ( XI .GE. 5.0 .AND. YI .LT. 5.0 ) PZ = 10.0*AMAG
C      IF ( XI .LT. 5.0 .AND. YI .GE. 5.0 ) PZ = 17.0*AMAG
C      IF ( XI .GE. 5.0 .AND. YI .GE. 5.0 ) PZ = 35.0*AMAG
C
C <<< EXAMPLE 3: SET NORMAL PRESSURE TO A CONSTANT VALUE >>>
C
      PZ = AMAG
C
      RETURN
      END
```

Figure 9. Sample user-defined subroutine for applied pressure.

The **MAGNITUDE** entry is the magnitude of the applied pressure. Finally, the **ZONE** entry indicates the zone identification number over which the pressure magnitude is to be applied. This entry is only used with the **DEAD LOAD** and **LIVE LOAD** load types and is a highly effective format for inputting different uniform loads over various zones within a slab bay.

A default pressure can be specified to be used if a zone is not designated with a specific ***NORMAL PRESSURE** statement. This statement uses the following syntax:

```
*DEFAULT PRESSURE
  MAGNITUDE
```

Another feature that has been incorporated into the analysis program is the capability to input a pre-existing membrane tension or compression using the statement

***UNIFORM STRESS**
σ_{xx}, σ_{yy}

where σ_{xx} and σ_{yy} are the slab tension (positive)/compressive (negative) stress in the x or y directions, respectively. Assuming plane stress, the input stresses are converted to equivalent strains as

$$\bar{\varepsilon}_{xx} = \frac{1}{E}\left(\sigma_{xx} - \nu\sigma_{yy}\right)$$
$$\bar{\varepsilon}_{yy} = \frac{1}{E}\left(\sigma_{yy} - \nu\sigma_{xx}\right)$$

which are then added to the strains due to bending deflection to yield effective total strains as

$$\varepsilon_{xx}^{eff} = \varepsilon_{xx} + \bar{\varepsilon}_{xx}$$
$$\varepsilon_{yy}^{eff} = \varepsilon_{yy} + \bar{\varepsilon}_{yy}$$

Materials are input using the following input statement:

```
*MATERIAL, TYPE = STRING, MATID = N
 MATERIAL PROPERTY VALUES
```

The **TYPE** parameter may be set equal to '**STEEL**' or '**CONCRETE**'. The **MATID** parameter is set equal to an integer number to identify different concrete or steel materials. For steel materials, the required property values in the data line are given by:

```
DBAR, SPCNG, E, AMU, Y
```

Where

DBAR = Bar diameter
SPCNG = Bar spacing
E = Young's modulus
AMU = Poisson's ratio
Y = Yield strength

For the input of concrete properties, the following quantities are required on the data line

E, AMU, XT, XC, Y

Where

E = Young's modulus
AMU = Poisson's ratio
XT = Tensile strength
XC = Compressive strength
Y = Shear strength

The failure response of the different input materials are specified in the form of damage laws. These laws used to reduce the material properties after failure are input using the following statement:

```
*DAMAGE LAW, DAMAGID = DID, NTLAW
 TYPE
 FORMULA
 PARAMETERS_1
 PARAMETERS_2
```

The parameter **DMAGID** gives the integer identification number of the damage law while the **TYPE** parameter on the first data line is set equal to '**CONCRETE**' or '**STEEL**' to indicate which basic material type is being defined for damage properties. **NTLAW** is set to **STREE-BASED** or **STRAIN-BASED** depending on which quantity is to be used in the smeared crack model for computing the damage modulus.

The **TYPE** entry is a string set equal to **CONCRETE CRACKING MODEL, STEEL YIELDING MODEL, OR COMBINED MODEL** to specify if the damage law is for steel only, concrete only, or for a layer with combined concrete and steel.

The **FORMULA** entry is set either to the string **CONSTANT POST-FAILURE MODULUS** or **PROGRESSIVE** depending on how the behavior of post-failed concrete is assumed

For steel materials, the parameter on the third data line is simply given by:

```
PARAMETERS_1 = EY
```

Where **EY** is the minimum Young's modulus of the steel exhibited after the yield strength has been exceeded. For instantaneous failure, this value is used immediately following predicted failure. For concrete materials, the parameter list on the fourth data line is given by:

```
PARAMETERS_1 = EC, ET, EY
```

Where **EC, ET,** and **EY** are the minimum reduced Young's moduli to be applied after concrete failure is predicted in compression, tension, and shear, respectively. This value is used immediately after failure is predicted for instantaneous failure, or is used as the minimum value in applying a progressive smeared crack model for reduced moduli.

The fifth line designated by **PARMETERS_2** contains the damage law parameters A, B, C, and D in the following crack model:

$$E_{RED} = \left[A + B(r-1) + C(r^2-1) + D(r^3-1)\right]E_{INITIAL}$$

Different layers present in the slab such as top concrete cover, inlaid top and bottom steel, and internal concrete layers are described using the following statement:

```
*LAYER, LAYID = N
 STID, CCID, DLAW, THK, THETA
```

The parameter **LAYID** is used to give each layer a unique identification number. The entries in the data line are defined as follows:

STID = Identification number of steel material
CCID = Identification number of concrete material
DLAW = Identification of damage law to be used with this layer

THK = Thickness of the layer
THETA = Orientation of the layer

This statement is general to permit the entry of both concrete and steel reinforcing bars in the same layer; if the layer is defined to contain exclusively concrete or steel, the **STID** or **CCID** entries are input as zero ('0') to exclude the other material type.

Zones are defined across the slab in order to specify different applied loads and different cross sections. These are input by using the following statement:

```
*ZONE, ZID = N
 A, B, XO, YO
 LAYER_1,LAYER_2,...,LAYER_N
```

Where **ZID** is the identification number of the zone and the entries on the following two data lines are defined as:

A, B = Length and width of zone
XO, YO = Coordinates of the zone center
LAYER_1 = Bottom layer identification number
LAYER_N = Top layer identification number

Zones may be defined with an allowed overlap. This removes the sometime difficulty of precisely setting the bounding coordinates for each zone. This overlap is inputted using the following statement:

```
*ZONE OVERLAP
 AZLAP (RATIO OF OVERLAP)
```

2.3 Output Control

Basic output to the output file *filename.out* is specified using the following statement:

```
*FILE PRINT
 STRING
```

Where **STRING** is a string of characters that may be concatenated in any order to specify specific output. These identifiers are given by:

D = Displacements
+S = Top fiber stresses
-S = Bottom fiber stresses
+E = Top fiber strains
-E = Bottom fiber strains
FI = Layer failure ratios

An example may be given by **STRING = FI-S+ED, -S+EDFI,** or **D,-S,+E,FI** to request displacements, upper fiber stresses, lower fiber strains, and failure ratios to be output in tabular format.

(Note: the following is an old attempt at implementing a graphics capability and should currently be neglected as a supported feature) A rudimentary graphics capability is invoked by specifying the following statement:

```
*GRAPHICS = EXCEL or TECPLOT
      STRING
```

Where **STRING** has the same interpretation as the **FILE PRINT** statement given above. This command generates external files given the names *graph1.plt, graph2,plt,… ,graph8.plt.* The data contained in these files is contained in the external file *contents.dat*.

These files were designed for use with the *GNUPLOT* shareware code that has been incorporated into a *JAVA* script distribution containing the *DEFLECT.EXE* executable which yields an automated package to execute the code with a menu driven display of graphical results.

```
C     <<< *GRAPHICS = excel or tecplot       >>>
C     <<<  STRING = (EXAMPLE) D+S-S+E-EFI    >>>
C     <<<                                    >>>
C     <<<  WHERE: D = DISPLACEMENTS          >>>
C     <<<        +S = TOP FIBER STRESSES     >>>
C     <<<        -S = BOTTOM FIBER STRESSES  >>>
C     <<<        +E = TOP FIBER STRAINS      >>>
C     <<<        -E = BOTTOM FIBER STRAINS   >>>
C     <<<        FI = LAYER FAILURE RATIOS   >>>
```

A limited set of diagnostic output can be selected using the following statement

```
*DIAGNOSTIC
 ND1, ND2
```

where **ND1** and **ND2** can be assigned the values of '1' or '2' – or zero. Setting the a value equal to '1' in the argument line causes all concrete cracking due to compression failure to be identified in the output. Setting a value equal to '2' causes the mode of all concrete cracking to be identified in the output. These modes are listed as TENSION, COMMPRESSION, and SHEAR failure for each layer at each difference point exhibiting failure.

2.4 Miscellaneous

Comments are introduced anywhere in the input by placing '**' in the first two columns of the input file. For example,

```
**  This is a comment
```

A job heading can be entered using the following command:

```
*HEADING
 STRING
```

Where **STRING** is a heading up to 80 characters in length. This heading is printed to the beginning of the output file *filname.out.*

During execution, the progress of the job can be displayed to the screen by including the following command anywhere in the input file *filename.inp*.

```
*STATUS
```

This directive causes intermittent information to be printed to the screen indicating which part of the overall solution process is currently being processed.

The mandatory statement indicates the end of the input file:

***ENDDATA -OR- *END STEP**

This statement causes the input file parser to terminate reading from the input file and proceed to processing the inputted model and executing the selected solution algorithm.

An input statement used to give some information on input errors is provided by the following statement:

***DEBUG**

This statement causes additional output to be output to the *filename.out* file that may help in discovering input errors. The current debug output is given by:

1) A listing of input statements read in. If a problem is encountered in the syntax of the input, the last input statement will be printed out which indicates that the particular input statement being processed is at the beginning of the problem involved in the input file stream. Without this information, an input error such as accidentally inputting an ‘o’ instead of a zero, ‘0’, will cause the Fortran read statement to terminate execution without any information as to where the substitution has been made. Knowing which block of input data is being processed encountering the error is extremely helpful in localizing where the syntax error is located in the input file.

2) The total slab thickness of each zone is output. This can capture errors in the definition of layers in defining the region of a particular zone. For example, if a uniform slab thickness is desired, an incorrect definition of layers in a zone will yield a zone thickness that differs from others, thereby localizing the error.

Appendix A

Sample Input File *Deflect.inp*

```
**
**   INPUT FILE FOR A SQUARE COLUMN SUPPORTED SLAB
**   WITH SYMMETRY CONDITIONS ALONG THE FREE EDGES
**
**   FCPRIM=4000PSI,   SL=156 psf
**
**   MODEL CORRESPONDS TO THE BASIC DESIGN AND MEASURED
**   DEVIATIONS OF SLAB GEOMETRY USED IN THE CHICAGO
**   POLICE BUILDING AT <LOCATION>.
**
*HEADING
 Park Place  (KSA Job No.3347) Garage Ramp Center Bay.
**
** select linear/nonlinear solution
**
**SOLUTION, STYPE = LINEAR
** 1.0
*SOLUTION, STYPE = NONLINEAR
 25, 15, 15, 0.05
**
** select section property calculation option
**
**NEUTRAL AXIS = GEOMETRIC
*NEUTRAL AXIS = WEIGHTED SECTION
**
** select whether solution is coupled in the inplane
** displacements and lateral displacements
**
*COUPLING = NO
**
** select which approach will be used for computing derivatives
** along boundary.
**
**BOUNDARY DERIVATIVES, METH = CENTRAL
**BOUNDARY DERIVATIVES, METH = MIXED
```

```
*BOUNDARY DERIVATIVES, METH = EXCLUDED
**
** request intermediate output during
** program execution
**
*STATUS
**
**
** input number of difference points used in the
** x and y directions, and to specify the extent
** of the column cross section
**
*DISCRETIZATION PARAMETERS
 10  10  1
**
** Establish default uniform slab
** Surface pessure load
**
*DEFAULT PRESSURE
 555.
** 156.
**     define slab loading as surface pressures
**
** describe higher loads over slab areas containing
** the drop panels with lower loads interior. these
** are dead loads associated with material self-weight.
**
**
**  slab layout is taken from library for a 4-column
**  corner supported slab with symmetry conditions
**  applied along free edges.
**
*BOUNDARY, TYPE = LIBRARY
 4-COLUMN, SYMMETRIC
**
** ENTER MATERIAL PROPERTIES IN LB AND FT
** SUCH THAT MODULI ARE GIVEN IN LB/FT**2 (PSF)
**
**   top steel with 6" spacing
**
*MATERIAL, TYPE = STEEL, MATID = 1
 .0625, 0.58, 4.176E9, 0.3, 8.64E6
**
**   bottom steel with 14" spacing
**
*MATERIAL, TYPE = STEEL, MATID = 2
 .052, 0.58, 4.176E9, 0.3, 8.64E6
**
*MATERIAL, TYPE = STEEL, MATID = 3
 .042, 0.50, 4.176E9, 0.3, 8.64E6
**
*MATERIAL, TYPE = STEEL, MATID = 4
 .042, 0.67, 4.176E9, 0.3, 8.64E6
**
*MATERIAL, TYPE = STEEL, MATID = 5
 .042, 0.83, 4.176E9, 0.3, 8.64E6
**
```

```
**  concrete material
**
*MATERIAL, TYPE = CONCRETE, MATID = 11
 5.191E8, 0.3, 6.831E4, 5.760E5, 3.64E4
** 5.191E8, 0.3, 6.831E4, 5.760E5, 3.64E4
** 4.4956E8,0.3, 5.915E4, 4.32E5, 3.155E4
** 3.670E8, 0.3, 3.416E4, 2.880E5, 2.576E4
**
**         DEFINE LAYERS:
**
*LAYER, LAYID = 7
 0, 11, 2, 0.042, 0.0
**
*LAYER, LAYID = 8
 0, 11, 2, 0.073, 0.0
**
*LAYER, LAYID = 9
 0, 11, 2, 0.042, 0.0
**
*LAYER, LAYID = 10
 0, 11, 2, 0.070, 0.0
**
*LAYER, LAYID = 11
 0, 11, 2, 0.078, 0.0
**
**
*LAYER, LAYID = 1
 1, 11, 3, 0.063, 0.0
**
*LAYER, LAYID = 2
 2, 11, 3, 0.052, 90.0
**
*LAYER, LAYID = 3
 3, 11, 3, 0.042, 0.0
**
*LAYER, LAYID = 4
 4, 11, 3, 0.042, 90.0
**
*LAYER, LAYID = 5
 5, 11, 3, 0.042, 0.0
**
*LAYER, LAYID = 6
 5, 11, 3, 0.042, 90.0
**
**   DEFINE ZONES OVER SLAB WHICH CONTAIN
**   DIFFERENT CROSS-SECTIONS
**
** zones about column centers containing
** the drop panels; zones 1 -> 4
**
*ZONE,  ZID = 1
 9.42, 6.50, 0.0, 0.0
 7, 7, 3, 5, 10, 10, 10, 10, 2, 1, 8, 8
*ZONE,  ZID = 2
 9.42, 6.50, 28.25, 0.0
 7, 7, 3, 5, 10, 10, 10, 10, 2, 1, 8, 8
*ZONE,  ZID = 3
```

```
 9.42, 6.50,  0.0, 19.5
 7, 7, 3, 5, 10, 10, 10, 10, 2, 1, 8, 8
*ZONE,  ZID = 4
 9.42, 6.50, 28.25, 19.5
 7, 7, 3, 5, 10, 10, 10, 10, 2, 1, 8, 8
**
** zones of top steel extending out from areas
** containing the drop panels; zones 5 -> 12
**
*ZONE,  ZID = 5
 4.71, 13.00, 7.06, 0.0
 7, 7, 3, 4, 11, 11, 11, 11, 4, 9, 8, 8
*ZONE,  ZID = 6
 18.83, 3.25, 0.0, 4.88
 7, 7, 3, 5, 11, 11, 11, 11, 9, 5, 8, 8
*ZONE,  ZID = 7
 18.83, 3.25, 0.0, 14.63
 7, 7, 3, 5, 11, 11, 11, 11, 9, 5, 8, 8
*ZONE,  ZID = 8
 4.71, 13.00, 7.06, 19.5
 7, 7, 3, 4, 11, 11, 11, 11, 4, 9, 8, 8
*ZONE,  ZID = 9
 4.71, 13.00, 21.19, 19.5
 7, 7, 3, 4, 11, 11, 11, 11, 4, 9, 8, 8
*ZONE,  ZID = 10
 18.83, 3.25, 28.25, 14.63
 7, 7, 3, 5, 11, 11, 11, 11, 9, 5, 8, 8
*ZONE,  ZID = 11
 4.71, 13.00, 21.19, 0.0
 7, 7, 3, 4, 11, 11, 11, 11, 4, 9, 8, 8
*ZONE,  ZID = 12
 18.83, 3.25, 28.25, 4.88
 7, 7, 3, 5, 11, 11, 11, 11, 9, 5, 8, 8
**
** zones describing interior slab regions
** which do not contain top steel
**
*ZONE,  ZID = 13
 9.42, 19.50, 14.125, 9.75
 7, 7, 3, 4, 11, 11, 11, 11, 9, 9, 8, 8
*ZONE,  ZID = 14
 28.25, 6.50, 14.125, 9.75
 7, 7, 3, 4, 11, 11, 11, 11, 9, 9, 8, 8
**
**               ENTER DAMAGE LAWS
** POST-FAILURE MODULI OPTIONS:
**    "CONSTANT" = CONSTANT POST-FAILURE MODULUS" IN WHICH
**                 THE DEFAULT VALUES WILL BE USED DIRECTLY
**                 AFTER FAILURE IS PREDICTED
** "PROGRESSIVE" = PROGRESSIVELY REDUCED POST-FAILURE MODULUS
**                 WHICH WILL APPLY AN EQUATION TO APPROXIMATE
**                 A REDUCTION IN MODULI DUE TO ASSUMED INCREASE
**                 IN CRACK DENSITY. THIS EQUATION IS OF THE
**                 FORM:
**                 E' = [A + B(N-1) + C(N-1)^2 + D(N-1)^3]Eo
**
**       WHERE N = SIGMA_RECOVERED/SIGMA_ULTIMATE     (STRESS-BASED)
```

```
**                 = EPSILON_RECOVERED/EPSILON_ULTIMATE (STRAIN-BASED)
**
*DAMAGE LAW, DID = 1
 STEEL
 PROGRESSIVE
 4.17E7
 0.01, -0.001, 0.0, 0.0
*DAMAGE LAW, DID = 2, STRAIN-BASED
 CONCRETE
 PROGRESSIVE
 5.19E6, 5.19E6, 5.19E6
 0.01, -0.001, 0.0, 0.0
*DAMAGE LAW, DID = 3, STRAIN-BASED
 COMBINED
 PROGRESSIVE
 4.17E7, 5.19E6, 5.19E6, 5.19E6
 0.01, -0.001, 0.0, 0.0
**
** select output quantities
**
**file print, option = max values only
**  d, -s, +e, -e, fi
*file print, option = max values only
  d
**
** invoke output graphics
**
*GRAPHICS = EXCEL
 d
**
*ENDDATA
```

The graphical output of defined zones can be graphed using Microsoft EXCEL. The following explains the procedure.

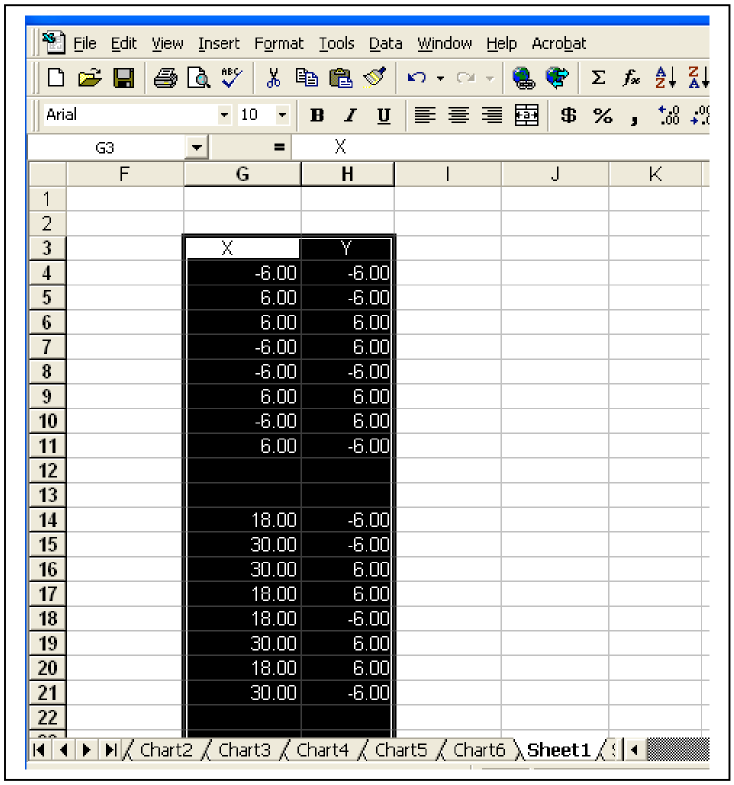
Steps to obtain zone graphics in EXCEL

1) Run def-pc.exe

2) Open *filename.zon* . You will see a column of numbers:

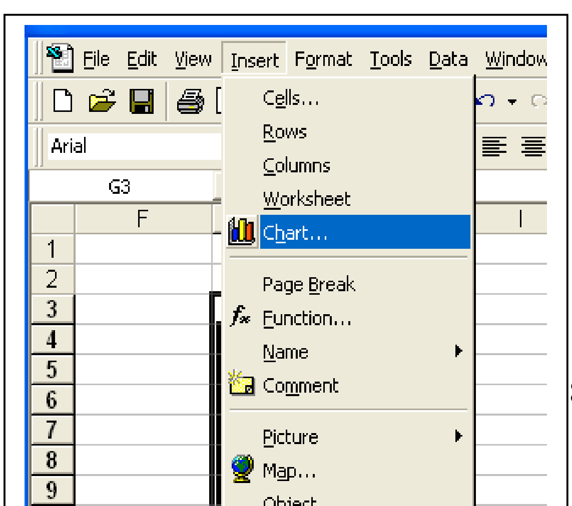
```
   X
-6.000
 6.000
 6.000
-6.000
-6.000
 6.000
-6.000
 6.000

 5.995
18.005
 5.995
18.005
    .
    .
    .


    Y
-6.000
-6.000
 6.000
 6.000
-6.000
 6.000
 6.000
-6.000


-6.000
-6.000
 6.000
   .
   .
   .
```

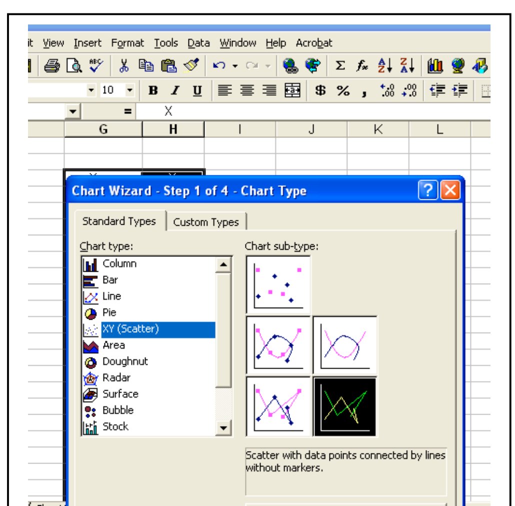
3) Cut the top half of the column labeled ‘X’ down to where the line ‘Y’ is. Paste this into a column in and EXCEL worksheet. Then cut and paste the remaining bottom half of the column into the adjacent column in EXCEL. Your EXCELL worksheet should look like:



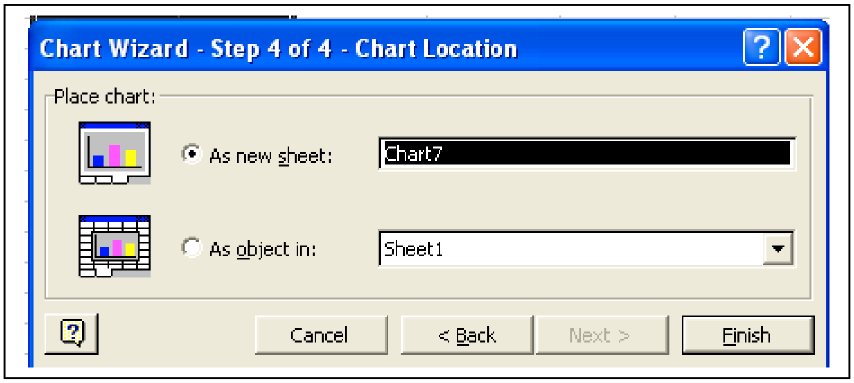
4) Highlight the ‘X’ and ‘Y’ columns down to the last number in the columns. This has already been done in the figure above.

5) Click on the INSERT menu and select CHART. This is shown in the left figure below. Under CHART you get a selection of choices, choose the XY-SCATTER plot and select the type of scatter plot as highlighted in black. This is shown in the figure on the right side below.



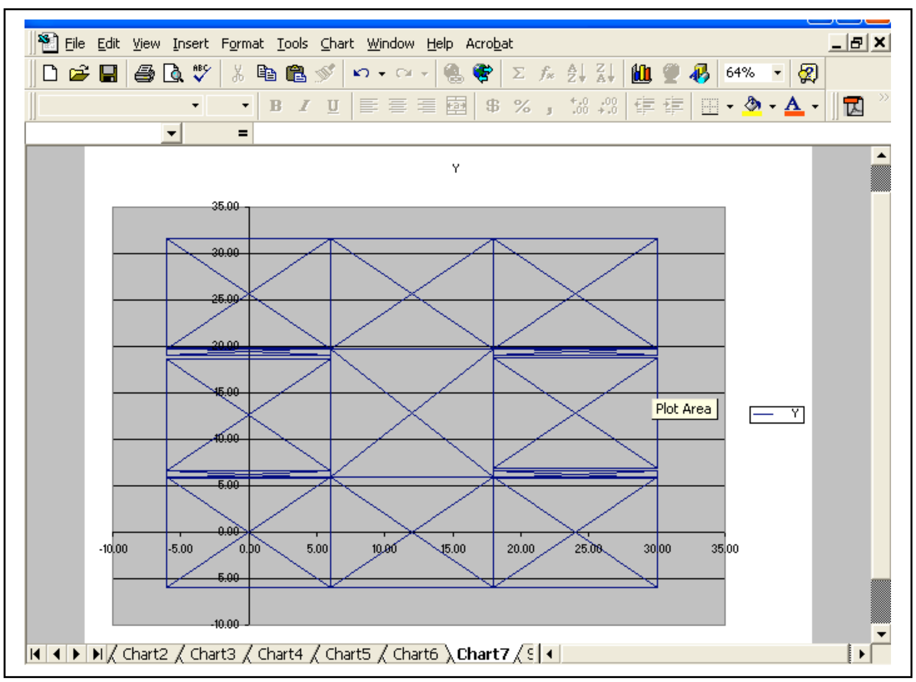


6) Select the NEXT button 3 times to get to this menu:

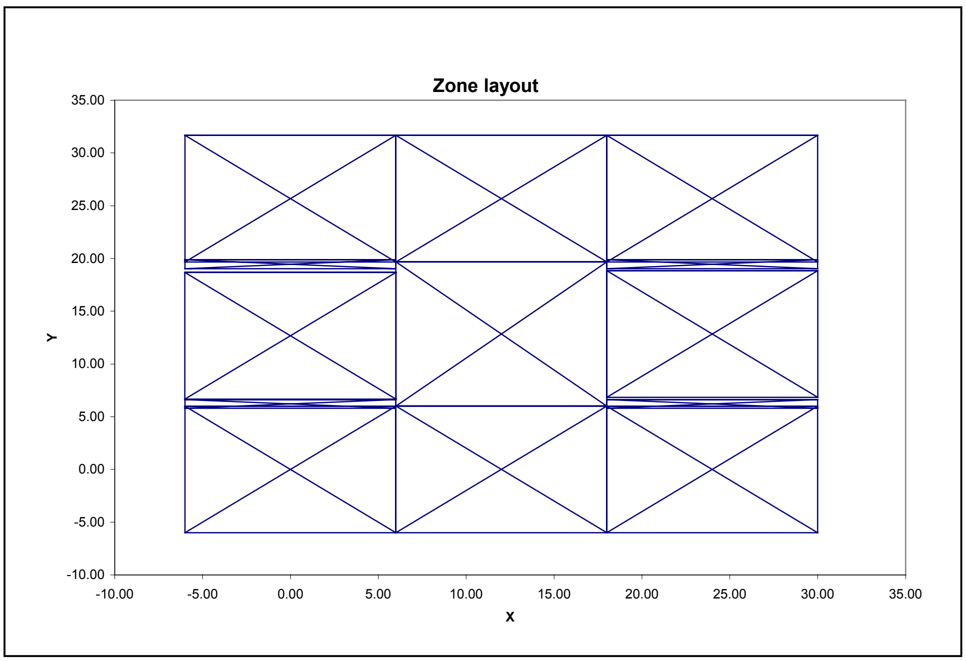


Select to have graphic placed on a new sheet and then click on FINISH.

7) This is what you'll see:



A pretty poor looking plot! But, it can be cleaned up by removing grid lines, setting the background to white, and moving the axes to look a bit clearer as shown in the figure below. Even without the clean-up, the EXCEL plot will still show problem areas in defining the zones.



Another feature which has been added to *def109.f* is the output of the displacement field in TEXPLOT format for graphical display of the deformation field. These are contained in three files denoted *filename.dp1, filename.dp2,* and *filename.dp3* which contain the displacement fields for the linear solution, cracking solution, and combined concrete cracking and steel yielding solutions, respectively. An example deformation field is presented below. These results correspond to the use of user-defined boundary conditions which specify a 4-column supported slab with the left bay free, and all other bays between columns with symmetry conditions.