E-20-F72
#1

Cracked Hole Finite Element Modeling

(E–20–F72)

Research Report Submitted to:
Lockheed Martin,

Program Manager:
Dr. Stephen P. Engelstad

Principal Investigator:
Dr. Rami M. Haj–Ali

School of Civil and Environmental Engineering
Georgia Institute of technology
Atlanta, GA 30332–0355

September 2001

Cracked Hole Finite Element Modeling

Table of Contents

Project Summary

The overall goal of this project is to develop a parametric finite element (FE) mesh generating software for creating models with cracks emanating from the edge of a circular hole in a notched plate. The term cracked hole mesh (CHM) will be used in this report to refer to the generated FE model. Also, the term parametric mesh generator (PMG) is used in this report. These terms are both used to describe the developed program, i.e CHM program or PMG program. The software input will be relatively small and include the geometry of the plate, the notched circular hole, and information about the crack systems that emanate for the circular edge. The geometry input is followed by mesh input in the form of number of elements at different major directions and areas of the model. The output of the program includes different files that define the geometry and FE mesh of the model. These files can be used as part of the input to a FE software, such as ABAQUS®. Furthermore, the output CHM files include PATRAN® neutral files that are generated to allow incorporating the CHM as part of a general global model, such as a global mesh for a lap splice joint structure.

Several examples were generated using the proposed cracked–hole mesh program. These demonstrate the general capability of the CHM automated modeling approach. A second post–processing program that is capable of using the ABAQUS output files and calculates the crack's strain energy release rate (ERR), using the virtual crack closure technique (VCCT), is also described. The program is called "aba_vcct". Different examples are generated and compared with ERR solution cases that are found in the literature. Finally, a case where the local CHM model is integrated in a global model is demonstrated.

1. General Overview and Objectives

Multi–site fatigue cracks typically initiate at the loaded fastener holes in skin splices. These multiple site damage (MSD) cracks can significantly reduce the residual strength of the structure. Current modeling approaches may not be sufficiently accurate to allow capturing the local deformation and damage evolution. This requires a detailed modeling approach that includes the local geometry, interacting damage systems, and account for the contact behavior of the fastener and the splice. Detailed finite element (FE) modeling of metallic fastened joints is crucial to accurately model existing and/or potential damage development. Refined shell element models can be used for this purpose. In addition, it is important to develop automated computer programs that are capable of generating a parametric and refined FE mesh.

A new "cracked hole" finite element modeling approach is developed in this study. The main goal of this project is to develop a computer program that can act as a parametric mesh generator (PMG) for cracked (or notched but uncracked) plate. Figure 1 shows different possible crack configurations that can be generated by the PMG. The program generates the meshes rapidly and efficiently. These can be used repetitively at numerous locations (e.g., in a skin splice), reducing the labor of re–creating this "mesh" numerous times. The mesh can be used as a substructure or can be condensed to a superelement by only retaining stiffness at the nodes on the interface boundaries of the substructure. The PMG output includes files that can be used in the input stage of the finite element software ABAQUS.

A post–processing program is also developed in this project. The purpose of this program is to read the ABAQUS binary result file "fil–files" and apply the virtual crack closure technique (VCCT) to obtain the energy release rate (ERR) at the crack tip. In this case, the PMG program has a parameter (IVCCT) that indicates the need to create a VCCT–crack–tip mesh area. Soft and stiff springs are located behind and in front of the crack tip, respectively. The PMG program creates an additional output file in case the VCCT mesh is needed. This file includes the tip nodal information and the spring elements. It is needed as an input for the "aba_vcct" post–processing program.

The capability of the PMG program to effectively generate complicated mesh models is demonstrated. Also, the VCCT modeling feature is applied on symmetrically emanating cracks and compared with an analytical solution available in the literature.

2. The CHM Program

This section describes the CHM or PMG program and the way it is applied. As previously mentioned, the possible general crack configurations that the program can generate FE models are shown in Fig. 1. The mesh is considered as a plane type mesh where plane–stress or plate type elements are used. The main features of the CHM code and the way it is activated are now described.

2.1 Program Arrays and Parameters

The file "Param_common" includes many parameters and array definitions. These parameters and the array sizes can be changed based on future changes that a user deems necessary. Also, the default array sizes can be increased to allow producing very detailed meshes. The following are major parameters and arrays:

max_num_cracks:	Maximum number of cracks allowed
max_num_theta:	Maximum number of angular intervals allowed
max_num_intrv_r:	Maximum number of intervals in the radial direction
max_el_blk:	Maximum number of elements in each block
max_nel_r:	Maximum number of radial elements in a block
max_nset:	Maximum number of nodes in a given nodal set
max_elset:	Maximum number of elements in a given element set
max_tip_nodes:	Maximum number of nodes in a detailed crack–tip area
max_tip_els:	Maximum number of elements in a detailed crack–tip area
max_rows:	Maximum number of rows within the transition zone
max_nd_row:	Maximum number of nodes in a row within the transition zone
Tol_theta:	Tolerance in the angular direction in terms of angles (degrees)
Tol_r_d:	Tolerance in the radial direction
Tol_eq:	Tolerance used for nodal equivalence
Tol_max_r:	Tolerance/fraction of the radial distance that defines the last circular perimeter that is connected to the outer rectangular perimeter.
idofs:	number of nodal degrees of freedom (2=plane–stress, 6 for plate or shell type elements)

The current default parameters allow the PMG to generate a wide range of applicable FE models with reasonable size of nodes and elements on the order of hundreds of elements. Having said that, the program is also capable of creating large scale meshes by increasing the "max_el_blk" and the "max_num_blks" parameters. Other parameters may also need to be changed depending on the desired procedures that are needed with the large–scale mesh.

2.2 Output Files

The output of the PMG program is in the form of several files that define the different part of the FE model and the way it is interfaced with the global model. Some of these files are not complete and/or part of them need to be redefined. For example, the boundary conditions may need to be redefined depending on the physical problem and loading conditions at hand. The name of an output file starts with a prefix string that is supplied by the user and followed by an attached suffix describing the type of the file. The following are the names of the output files:

"prefix"_abaqus.inp

This is the abaqus input file that includes the nodal and element definitions. In addition, the file may include nodes and elements in a transition area that connects with the global mesh. the interface with the global mesh can also be in the form of nodal constraints between the fine local mesh and the coarse global mesh. The transition area and global/local interface is an optional feature of the code. Alternatively, one can use the PMG program to define a local mesh, i.e. a rectangular plate with a notched circular hole with a number of cracks emanating from the notch. A fictitious static problem with elastic material definition, axial loading, and boundary conditions are added to the file in order to complete its definition. The user should be aware to modify and/or add the desired properties and procedural definitions that are appropriate to their model.

"prefix"_abaqus_sup.inp

This file is similar to the first abaqus file except it defines the CHM definition as a superelement. This is by using the "*SUPER,ID=Z1001, FILE=sup_lib1, RECOVERY=YES" option in Abaqus. This command line maybe modified in the file by the PMG users to reflect a proper element number and/or library filename etc. The retained DOF of the superelement are currently defined by the outer nodal perimeters that can interfaced with the global mesh. The procedure for defining the superelement ends with the Abaqus command line: "*END SUPER" followed by an incomplete global model. This model needs to be completed by the user in order to properly define the superelement within the global model. Furthermore, the material definitions in the superelemnt may need to be modified.

"prefix"_patran_p1.out "prefix"_patran_tr.out "prefix"_patran_all.out

These three files are PATRAN neutral files that include the definitions of the nodes and elements of the model. The first file will always be generated. The second and third files will be generated in case a triangular transition zone is requested from the PMG program. in this case, the second file contains the nodes and elements of the transition zone alone while the third file contains all nodal and element definitions.

"prefix"_vcct.inp

This file is generated in the case where the VCCT method is needed for evaluating the ERR at the different crack tips. The information about the crack tip nodes, and the soft and hard spring elements added in the model, is written in this file. This file is needed later as an input to the VCCT program in order to post–process the ERR from the analysis results.

"prefix"_ses

This file is generated to report the elements that are deleted in the mesh and replaced with a detailed crack–tip rings for J–integral calculations. It also may include warning or error messages.

nodal_constraints

This file is generated when the method for interfacing with the global mesh is nodal constraints. The nodal constraints are generated using quadratic shape functions between the global coarse nodal edges and the local fine mesh. The corner nodes are directly linked with each other. The file should be used with the "*EQUATION" option in Abaqus.

2.3 Program Input File

One input file is needed to generate a FE mesh. Each line of input that begins with the character "C" or "c" is ignored. Therefore, one can use this to write comment lines. The input parameters are described using schematic mesh illustrations.

1) Input Stage #1: Geometry parameters

The plate width, height, and diameter of the notched hole (a,b,D) need to be given in the first line of input. Figure 2 identifies these input parameters and describes the initial stage of the mesh and how it is partitioned to a default 8–blocks. The dotted square includes part of an input file illustrating the current line of input. Note that lines that begin with "C" or "c" are used as comment line. Also, the actual lines of input that are needed are marked with horizontal left arrow. Other lines are comment lines. Finally, the local hole–center coordinate is set to (0,0,0). This can be changed later allowing the nodes to be translated once the mesh is completed prior to creating the transition zone.

2) Input Stage #2: Optional Non–default Partition Lines

This option is in addition to the default lines that are shown as dashed lines in Fig. 3. These additional lines may be needed to define refined mesh areas especially near crack systems. The example in Figure 3 illustrates the addition of 6 lines at different angles around a cracked lines that will be later defined.

3) Input Stage #3: Crack Systems

The data for the crack systems (crack tip coordinates) is given in this step. Figure 4 illustrates an example where two crack systems are defined by their tip coordinates. This example describes horizontal system of unsymmetrical cracks, however, other crack systems can be easily defined, as shown in Fig. 1. Figure 5 shows the current status of the model after the first three stages of input. It is interesting to note that each crack tip has defined a circular line of nodes. An end circular perimeter of nodes is also defined with a radius being the minimum of half of the width or the length times a given ratio (0.85). This ratio is defined in the common parameters file. Finally, the IVVCT parameter near the number of cracks indicates whether to apply the VCCT approach for these crack tips (non zero value) or that the VCCT method will not be used (zero IVCCT value). The section on the VCCT program explains how the CHM program effects the model in the case where the VCCT option is requested.

4) Input Stage #4: Added Circular Lines

Circular lines are by default created in the mesh because of crack tips, at the (0.85) ratio of minimum radius, or as the notched circular hole itself, see Fig. 5. This optional input stage allows adding more circular lines to the model in order to control the refinement of the mesh at different areas. Figure 6 shows an example of adding two additional circular lines to the current model.

5) Input Stage #5: Mesh seeds in the Circumferential Direction

The line segments in the circumferential direction are created as a result of the intersection of the radial lines and the circular lines. These segments are seeded for meshing at this stage. The only input that is required is the number of elements that each line segment will have. Later, specific line segments can be modified to have different number of elements. Figure 7 shows an example of the seeds as a result of the given input.

6) Input Stage #6: Mesh seeds in the Radial Direction

A similar input is needed to mesh seed the radial line segments with equal number of elements. Figure 8 shows an example using three elements in the radial direction.

7) Input Stage #7: Modify Mesh seeds in the Radial Direction

The previously defined number of elements is for all segments in the radial direction. In the current stage of input, certain line segments ranges can be selected or defined with different number of elements than that in the previous stage. Figure 9 illustrates the use of the current input option to override the number of elements from 3 elements to 2 and from 3 elements to 4 in other segments. Note that the last range overrides the previously defined ranges as shown in the example.

8) Input Stage #8: Mesh seeds for the last radial segment

The last radial segment connects the plates' outer perimeter of nodes to the last circular line of nodes. The number of element in this zone is defined separately in this stage. Figure 10 shows the number of radial elements that divides this last radial segment.

9) Input Stage #9: Refined Meshes Around Crack-Tips

The next input in this stage is to define whether there is a need for refined crack tip meshes. If so, a non zero number is needed to indicate the number of crack tips that need mesh refinement around their tips. Figure 11 shows an example where it is indicated that no mesh refinement is needed around any crack tip. This is useful when working to prepare the mesh for the VCCT calculations of the ERR. In this case, a later stage will be added to invoke the VCCT option. It should be noted that it is highly recommended that the length of elements in the radial directions are set to be equal behind and in front of the crack tip. This can be achieved in the input stage #4 where added circular lines can bound the crack tip to insure equal length of elements.

Figure 12 illustrates the second option where a refined mesh around one of the crack tips is needed. the first input is the number of these tips. The second line of input identifies the following three parameters:

1) theta – Crack angle which needs detailed tip meshing

2) fract_rmax1 – [0+, 1.0] – Fraction which determines the radius for the first ring or circle as Rmin=fract_rmax1*Min(Lr,Lt)

3) fract_rmax2 – [0+, 1.0] – Fraction which determines the radius for the last circular tip – ring Rmax=fract_rmax2*Min(Lr,Lt)

Note: (fract_rmax2 > fract_rmax1)

The above fractions and distances are applied to the area that is free of elements around the crack tip. In order to create this area, elements need to be deleted in the radial direction behind the crack (r1 – direction), in the radial direction in front of the crack tip (r2 – direction), in the angular direction below the crack face (t1 – direction), and in the circumferential direction on top of the crack face (t2–direction). Figure 13 illustrates the case where the number of these elements are defined in one line of input. In addition this line of input requires two additional numbers that define the number of circular rings to define around the crack tip and the number of transition rings of elements that needed to connect with the rest of the mesh. Finally, Fig. 14 shows the outcome of the refined mesh around the crack tip.

10) Input Stage #10: Nodes and Elements Renumbering

Two numbers are needed in the next line of input. The first number is added to the node numbers that are generated thus far. This is useful in case the current mesh needs to be part of another global model and to avoid node number duplications. The second number is used in a similar fashion with the element numbers.

11) Input Stage #11: Translation of the Nodes

Thus far, the model was created in a local coordinate system where its center (the hole or the plate center) has a (0,0,0) coordinate. In this step, a line of input that contain the x,y, and z position of the center is needed.

12) Input Stage #12: Transition Mesh Zone and Interfacing with a Global Model

This stage includes input for interfacing with a global model by creating transition mesh or applying nodal constraints between the coarse global mesh and the local CHM. The first input line requires an integer input for a flag (iconnect). If the input is zero, no interfacing or mesh transition is occurred. In the case it is not zero, the global nodes that the local model will interface with need to be given. The list of nodes must be consecutive and starts from the lower left corner and counter clockwise. Two types of files for these nodes can be read by the CHM program. The first is a free–formatted ASCII file that each line has node number and the (x,y,z) coordinate. The second file is a patran neutral file that has *only* the nodes that the local model will be connected to. The parameter (Inp_type) is used for this purpose in the next line of input. If the parameter is set to 1, that the input file is an ASCII type, else, it is a Patran neutral file. The next line of input is the file name. This input must begin from the first column, i.e. no space before the first character, otherwise, the Unix system will not be able to open the correct file.

The following input line must consist of 4 node numbers that exist within the external nodal file. These are the four vertices of the global nodal perimeter starting from the lower left corner and progressing in a counter clockwise direction. The next input line is a number that indicates if the local–global interface is through layers of triangular elements (IOPT=1) or through nodal constrains (other value). No additional input is required in the later case. Figure 15 schematically shows the nodal constrains approach. If the triangular elements are needed for transition, an additional line with two parameters is required. These are the number of layers (rows) of elements to be used (at least 2) and the nodal reduction fraction between the layers of nodes as we transition between refine to coarse lines of nodes. Figure 16 shows examples of the triangular transition zone.

3. Template Input File

A template file was created as an example to help in creating input files to the CHM program. Figures 17 and 18 show this file. The needed input lines are not commented with a "c" or "C" characters at the beginning. The needed numerical input in these lines is marked using the symbols $I and $FM for integer and free–format floating numbers, respectively. The range of some of the input variables is fixed and given to aid in creating input files.

4. The VCCT Program

A second program that is able to read and post–process the ABAQUS binary results file was written during this project. The aim is to use the crack information data and read the displacements and force values in the springs to calculate the ERR. The input to this program includes a file with the information about the number of cracks, the tip nodal numbers, spring elements, and other needed information. The CHM program automatically generates this information once the IVCCT parameter is used. This feature is previously explained. Therefore, the user must prepare the mesh around the crack to have the same element size before and after the crack tip. Once this feature is requested, the nodes near the crack tip are connected with soft and hard spring elements, as shown in Fig. 19. The forces from the hard springs and the displacements from the soft springs are used in the post–processing program to calculate the EER.

Next, the program was used to verify the VCCT modeling of the CHM and the post–processing programs. To this end, the known solution of a notched plate with two symmetrically edge cracks, Fig. 20, is used for this purpose. The FE mesh created by the CHM program is also shown in Fig 20. The CHM program also created the input information for the post–processing program (aba_vcct.f). Figure 21 illustrates the accuracy of the proposed models along with the post–processing verification of the developed code for seven cases of geometry.

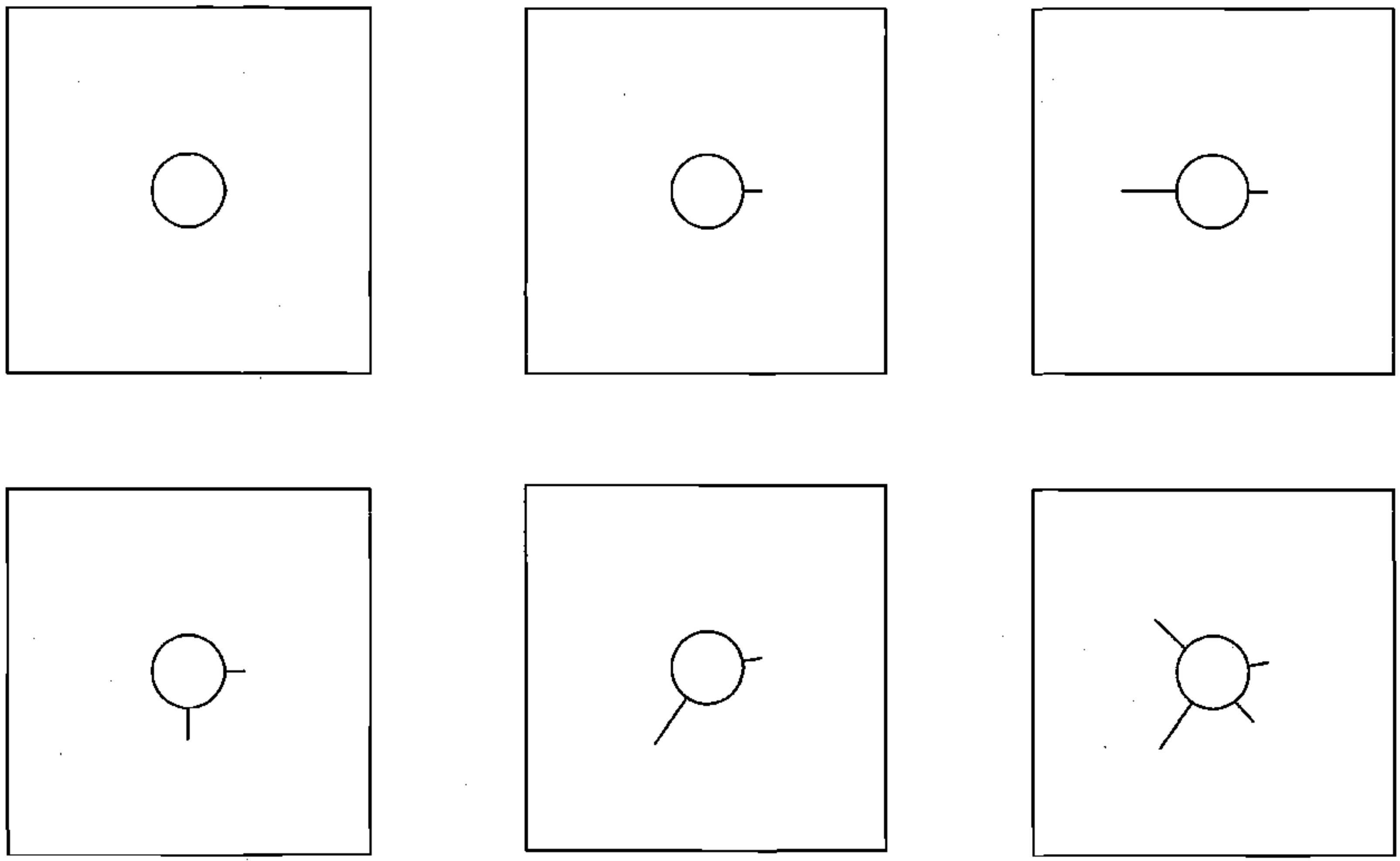


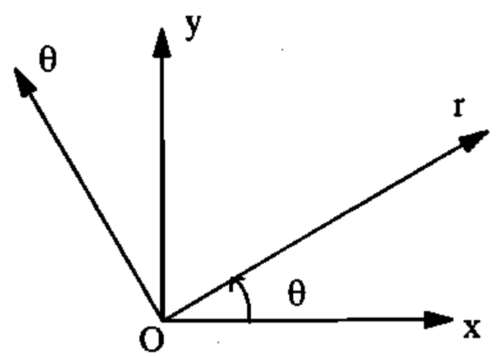
Figure 1. Different possible crack configurations that can be meshed with the current mesh generator

Input Stage #1

```
c
c 1)  Geometry of the plate and hole
c
c  a = width  b = hight   D = Diameter of the hole
c
c
  10.0     12.0      2.                  <------
c
```

NOTE:

At this stage, there are 8 initial blocks



Local Coordinates at O

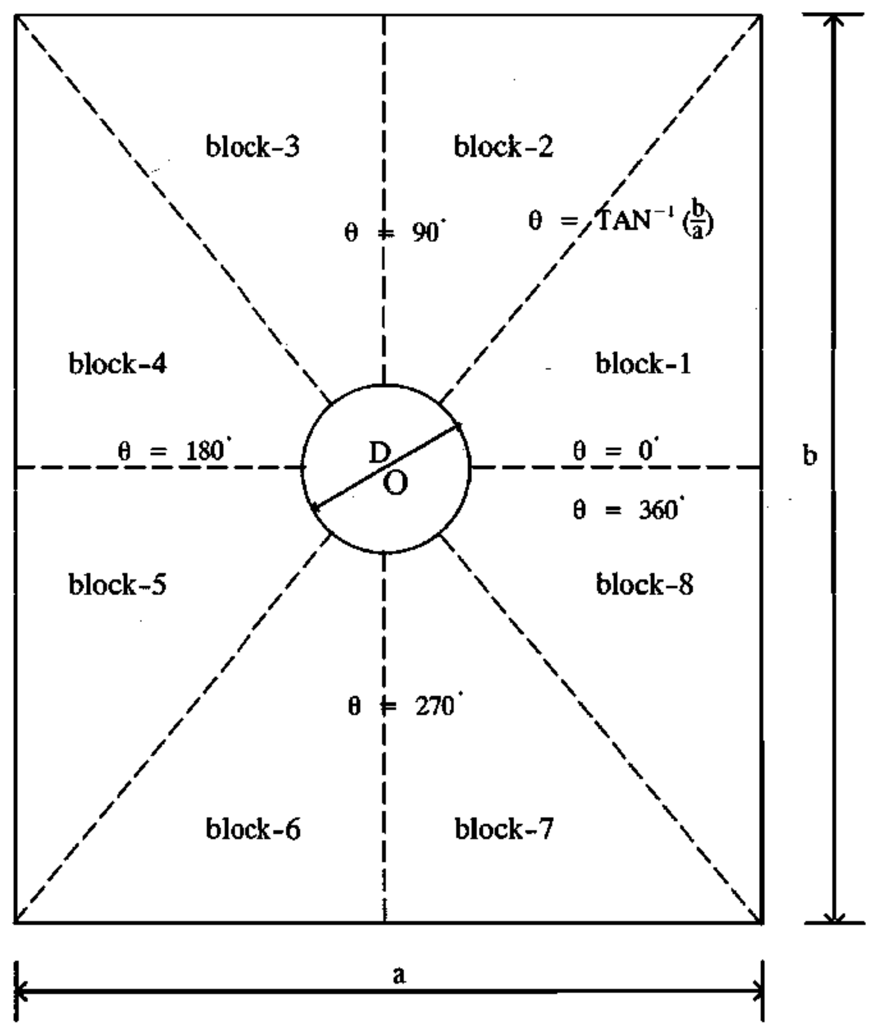


Figure 2.   Input stage #1: initial geometry with default partition of 8-blocks.

Input Stage #2

```
c
c 2) Input Additional Radial Lines (degrees)
c       that will be added to the default radial lines
c
c        Note:  These lines may or may NOT have
c                  radial cracks emanating from the
c                    edge of the hole.
c
c Number of added radial lines
  6                                          <------
c
c Angle (degrees) associated with each line
c
  -18. -5. 5. 18.   172.5  187.5             <------
c
```

- - - - - Default radial line

—— · —— Non-default (added) radial line

<u>NOTE:</u>

At this stage, there are 14 total blocks

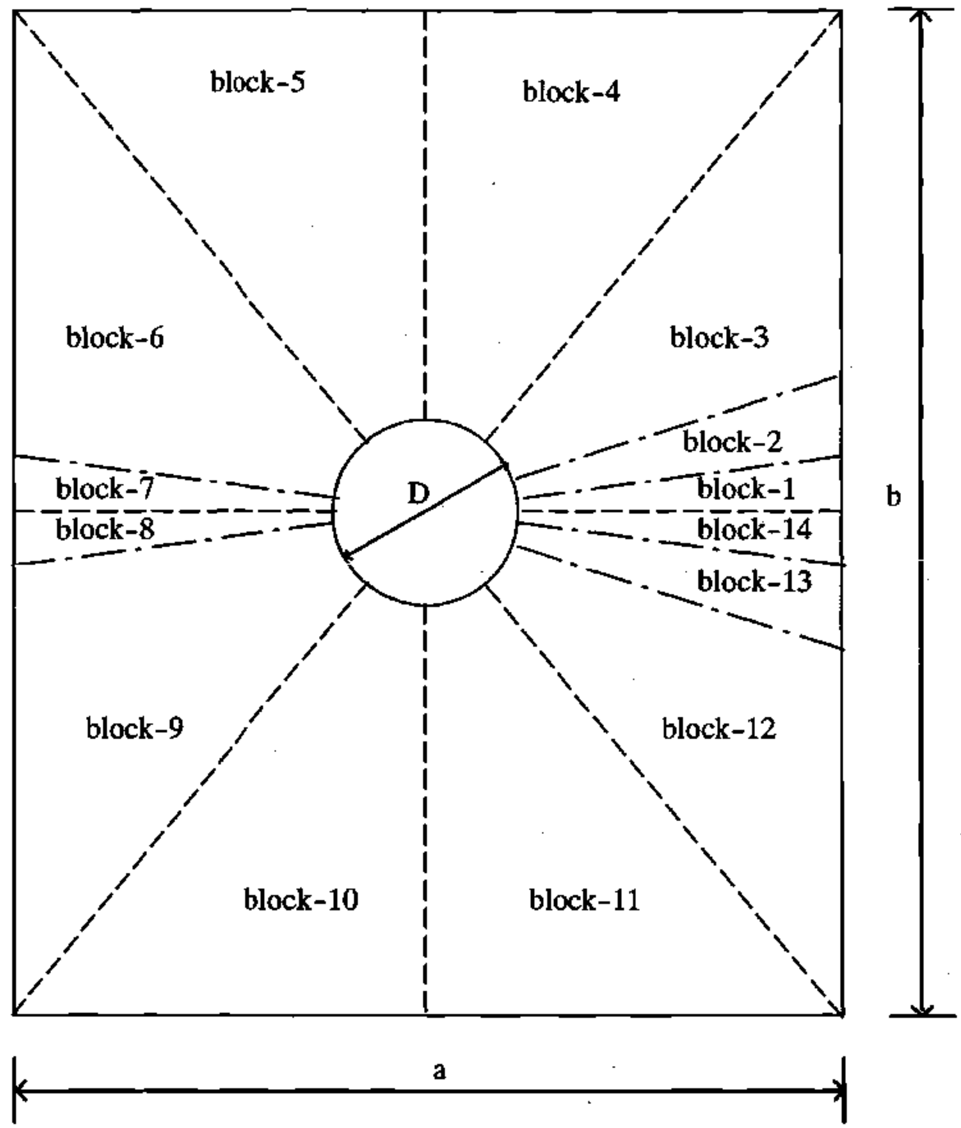


Figure 3. Input stage#2: adding more partition radial lines.

Input Stage #3: Crack Tips

```
c
c Crack Systems:
c
c 3) How many crack systems in this mesh/model?
c
c Num_cracks     IVCCT
  2                0          <----
c
c  For each crack system, enter the angle and
c  the radius of the crack tip (poolar coordinates)
c
c  Max Value of R_tip for each crack:
c Max(R) < 0.85*Min(a/2,b/2)
c              (0.85 predefined factor)
c
c  Theta       R
    0.0       1.8             <----
   180.0      3.2             <----
c
c
```

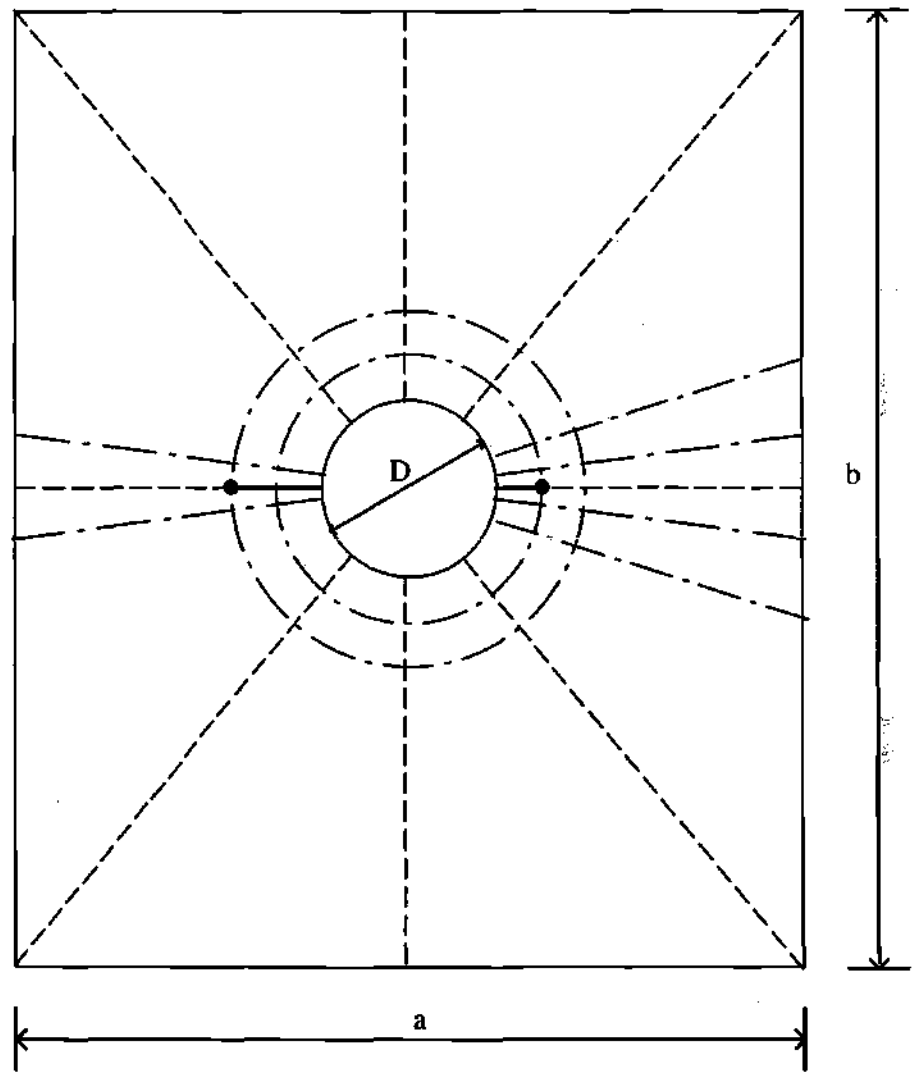


Figure 4. Input stage #3: Crack systems.

Current status:

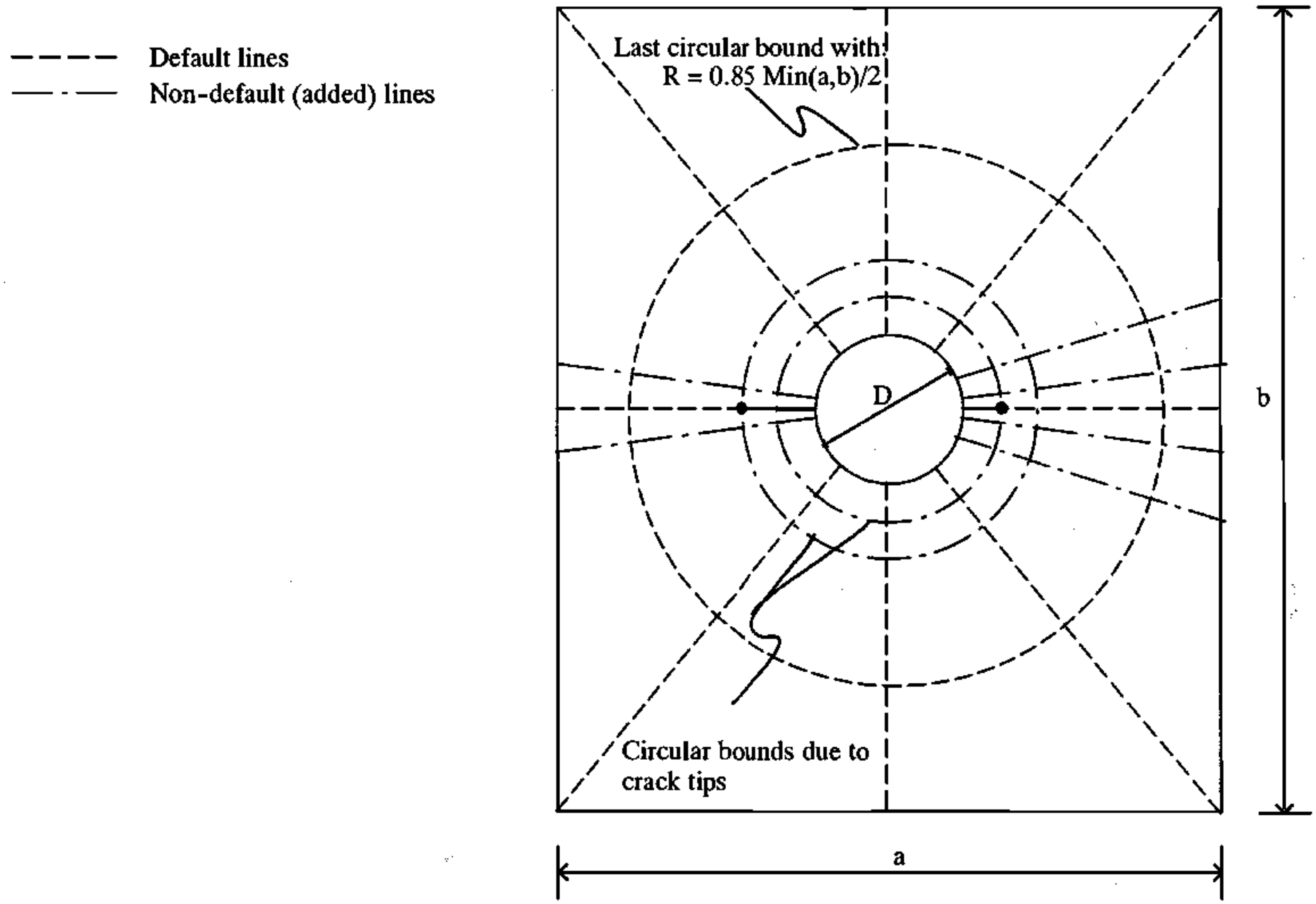


Figure 5. Current status of the model after the first three stages of input.

Input Stage #4: Added Circular Lines

```
c
c 4) Number of additional circular lines
c       (not due to crack tips)
c
c these will add additional "radial intervals"
c
c Total Radial Intervals= Number of DIFFERENT
c crack tips + num_add_r_int + 1
c
c Num_add_r_intrv
c
  2                                   <----
c
c  R1 R2 R3 .... R(Num_add_r_intrv)
c
 1.5  3.5                             <----
c
```

------ Default lines

—·— Non-default (added) lines

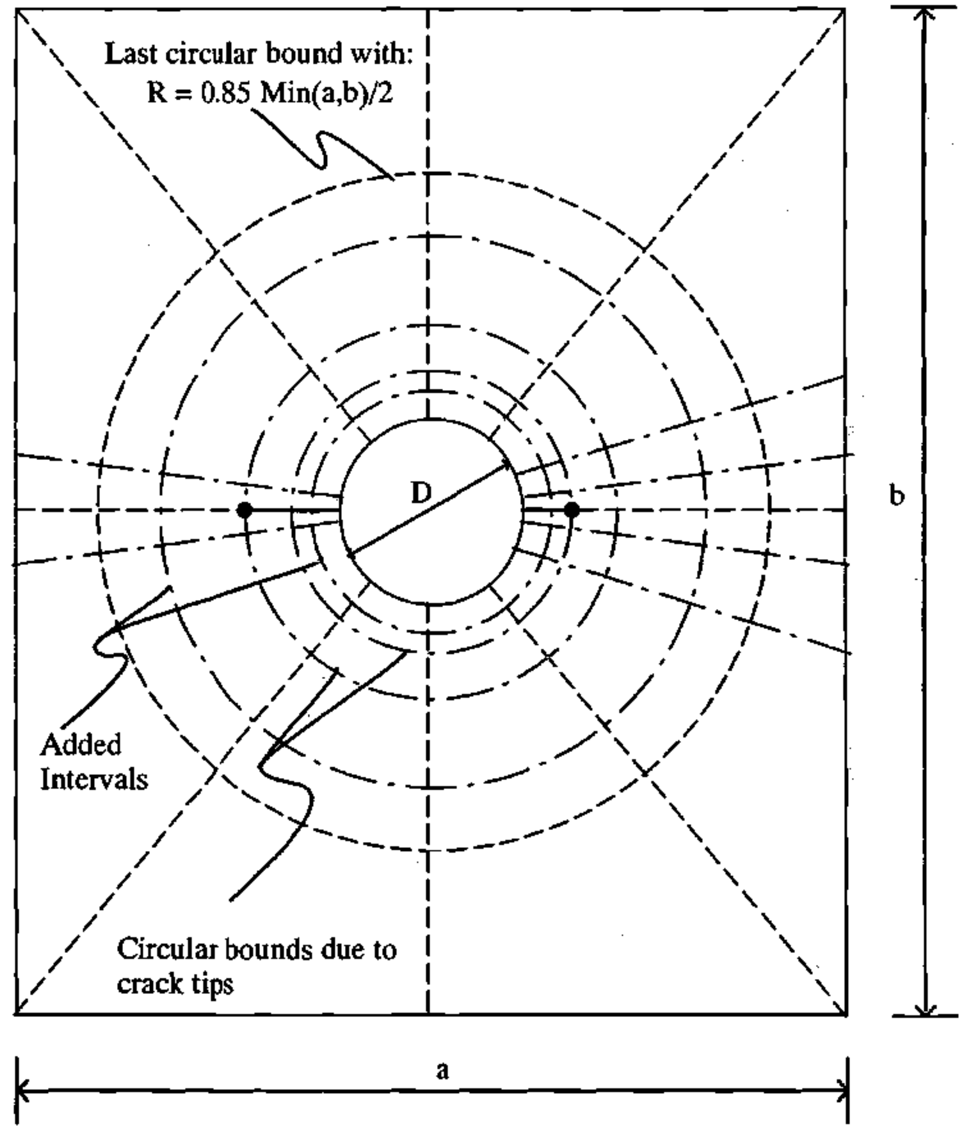


Figure 6. Input stage #4: added circular lines.

Input Stage #5:
Mesh Seeds in the Angular Direction

```
c
c 5)  Mesh seeds in the angular direction.
c     All line segments in the circumferential
c     direction will have the same following
c      number of elements (> 2).
c
c num_el_theta     ( > 2 )        <-----
   2
```

2-elements will be meshed between ALL line segments in the angular direction

■ - Mesh Seed

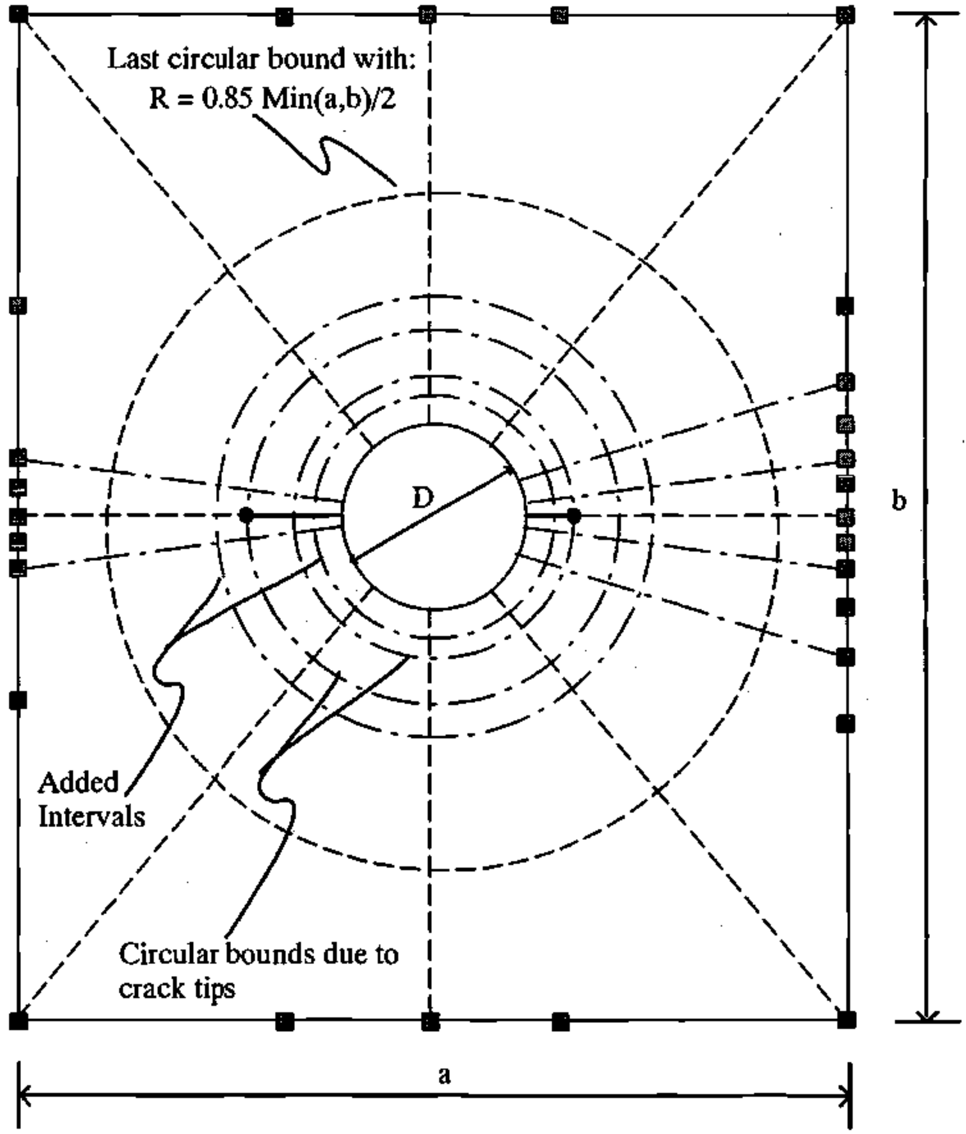


Figure 7. Input stage #5: Mesh seeds in the circumferential direction.

Input Stage #6:
Mesh Seeds in the Radial Direction

```
c
c 6) Mesh seeds in the radial direction.
c
c num_el_r
  3                    <-------
c
```

■ - Mesh Seed in Theta direction

● - Mesh Seed in the r-direction

<u>NOTE:</u>

2-elements will be meshed between ALL line segments in the circumferential direction

3-elements will be meshed between ALL line segments in the radial direction

ALL lines in the r-direction will share the same mesh seed segments

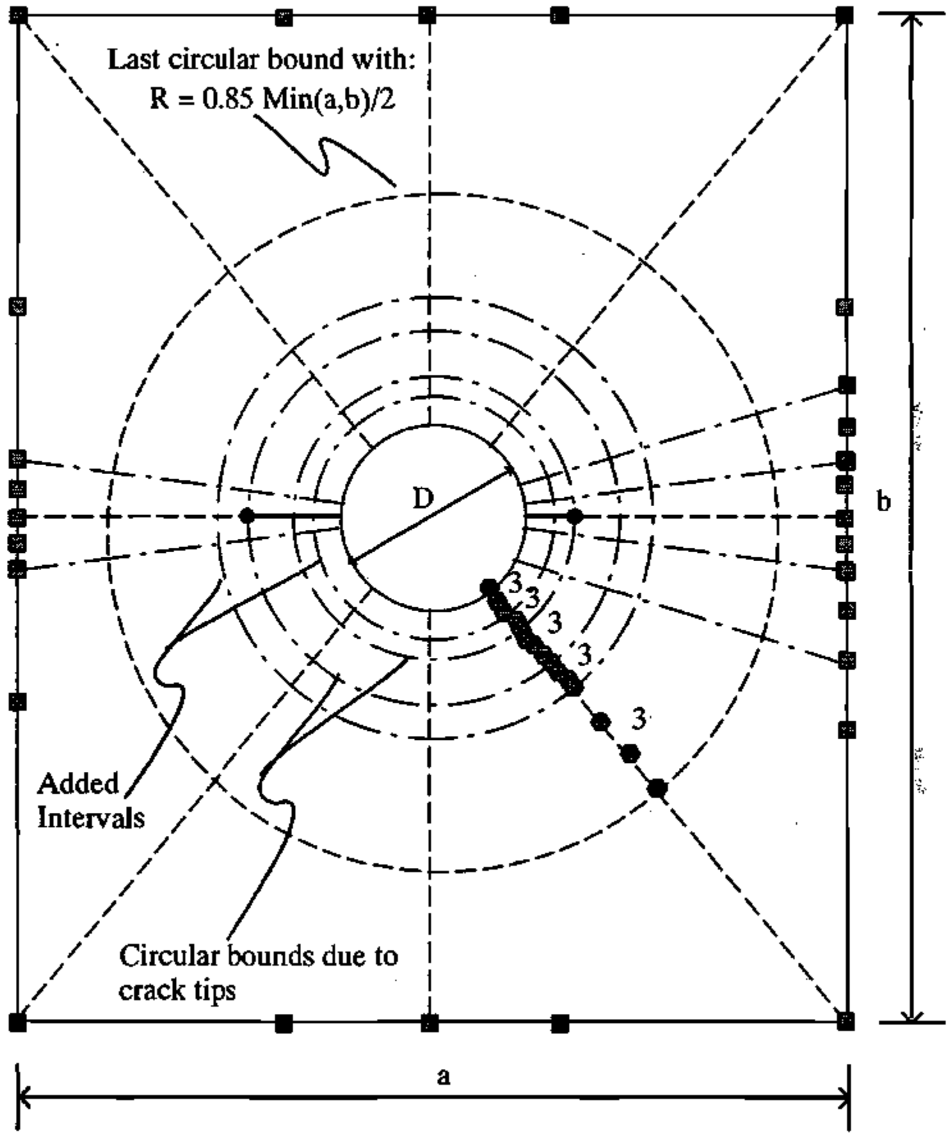


Figure 8. Input stage #6: mesh seeds in the radial direction.

Input Stage #7:
Mesh Seeds for Specific Radial Segments

```
c
c  7) Overwrite specifc number of seeds in the
c       radial segments within given ranges:
c                 R(i) < r > R(i+1)
c
c  num_el_rp
c
   2                                          <----
c
c  r1   r2    num_el_r_mod
 0.0  1.80     2                              <----
1.79  4.26     4                              <----
c
c
```

■ - Mesh Seed in Theta direction

● - Mesh Seed in the r-direction

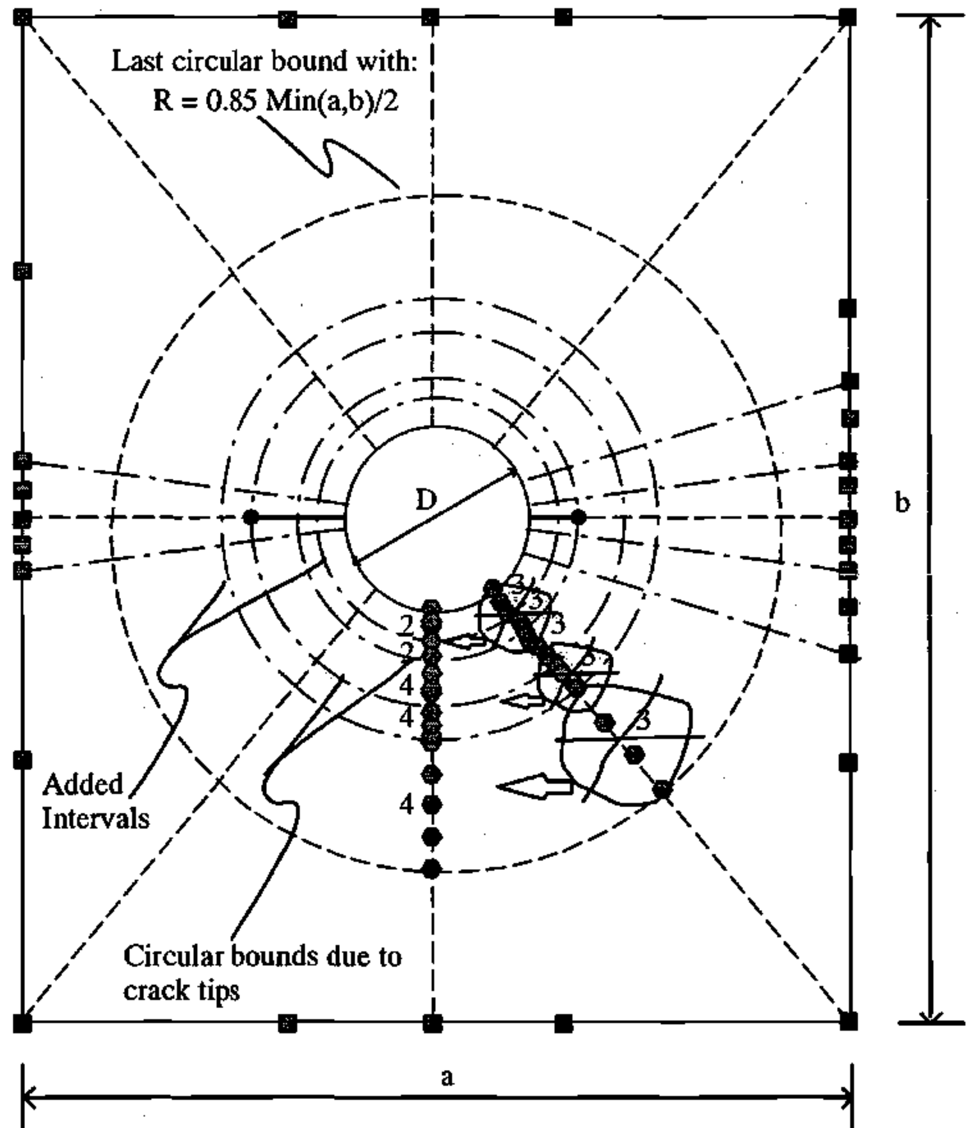


Figure 9. Input stage #7: mesh seeds for specific radial segments

Input Stage #8:
Mesh Seeds for the Last Radial Segment

```
c
c 8)  Number of element to add in the last radial
c       segment between the last circular line
c         and the plate perimeter
c
c num_el_add_r
   4
c
```

■ - Mesh seeds in the angular direction

● - Mesh seeds in the r-direction

▲ - Mesh seeds for the last radial segment

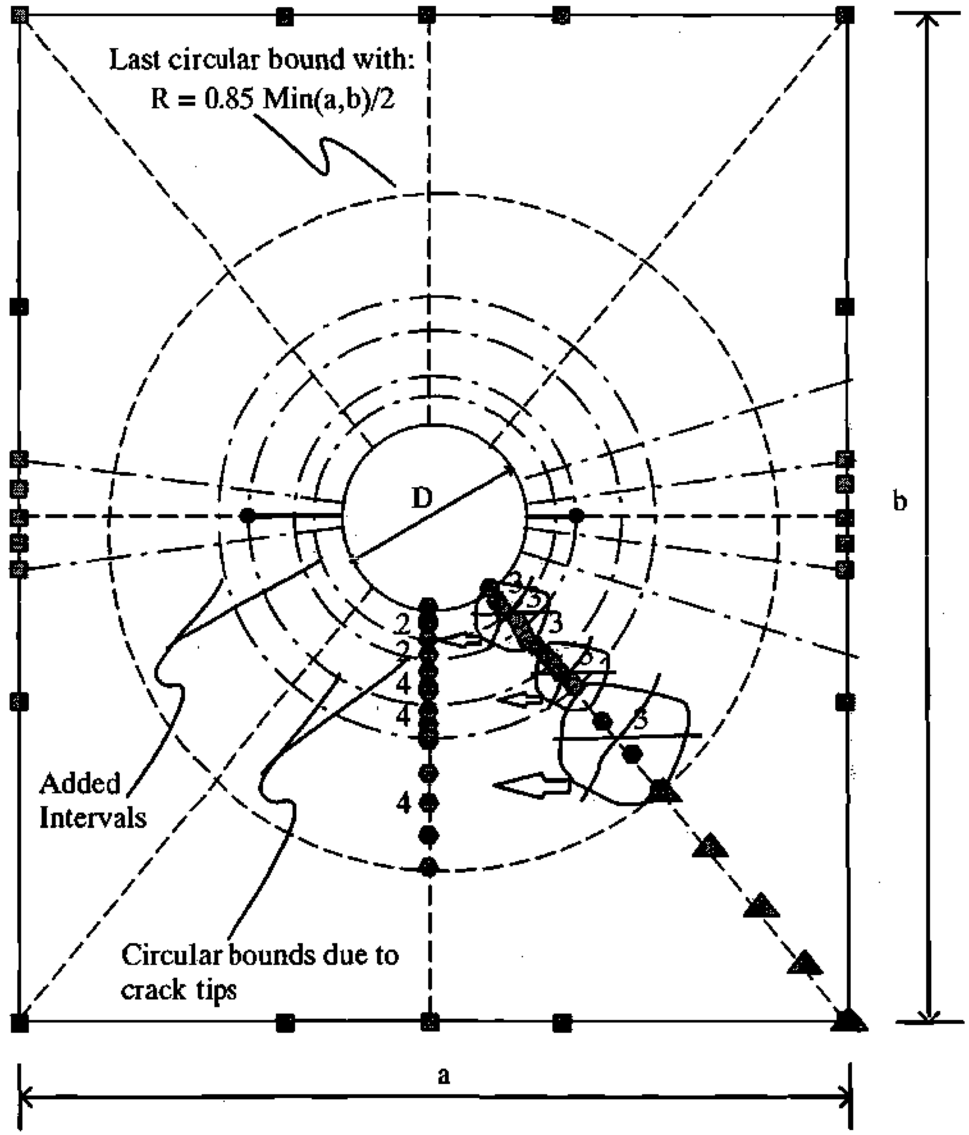
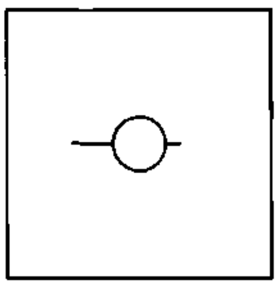


Figure 10. Input stage #8: mesh seeds in the last radial segement.

Input Stage #9:
(Option−1: Prepare for VCCT Mesh)
No Refined Meshes Around Crack−Tips

```
c
c 9)  Detailed Crack Tip Meshes (Y/N)
c        (Y/N) = (*/0)
c
c num_det_tips
c
  0
c
```



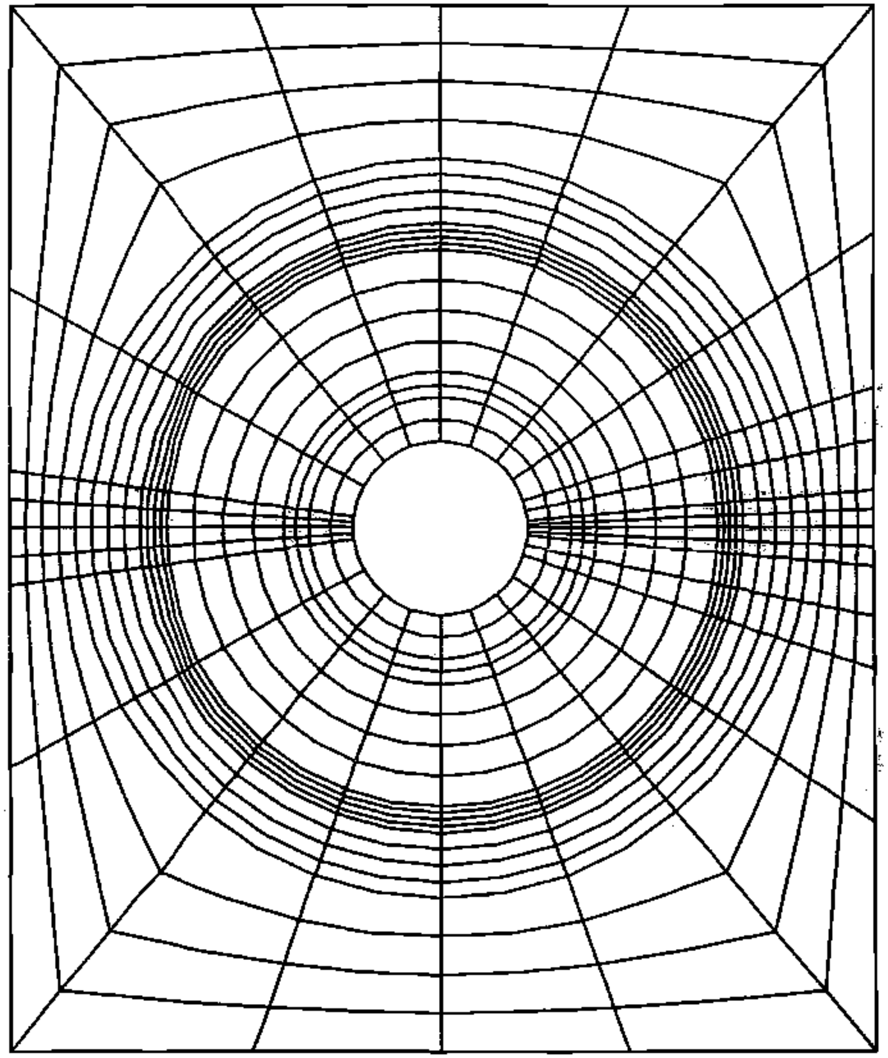


Figure 11. Input stage #9: no refined meshes around crack tips

Input Stage #9:
Refined Meshes Around Crack–Tips
(Option–2: Plasticity or J Calculation)

Define a refined mesh only
around the right crack.

```
c
c 9)  Refined Meshes Around Tips
c
c  num_det_tips
c
  1                          ←
c Identify which crack:
c  theta   fract_rmax1  fract_rmax2
c
   0.0   0.05   0.85         ←
c
c  num_el_tip_r1  num_el_ti_r2
c num_el_tip_t1  num_el_tip_t2
c num_rings num_add_rings
c
  3  2  4  4  5  2           ←
c
```

Note: next figure explains the above input variables

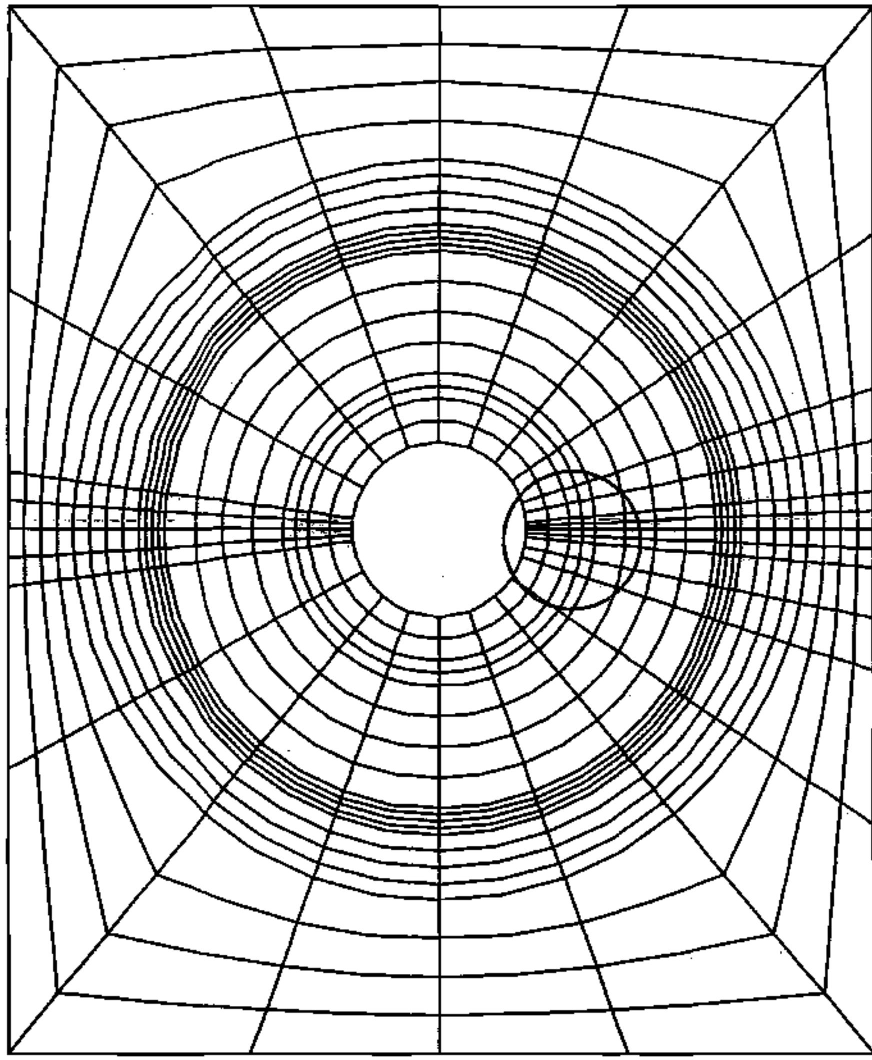


Figure 12. Input stage #9: refined meshes around crack tips

Input Stage #9:
Refined Meshes Around Crack–Tips
(Option–2: Plasticity or J Calculation)
(cont.)

```
c theta   fract_rmax1  fract_rmax2
c
  0.0  0.05  0.85          <------
c
c num_el_tip_r1  num_el_ti_r2
c num_el_tip_t1  num_el_tip_t2
c num_rings num_add_rings
c
  3 2 4 4 5 2              <------
```

A circle is created around the tip deleted area of elements such that its radius is 0.85 of the minimum lenght in the 4-directions (t1,t2,r1,r2) to ensure that the circle is bounded.

0.05 is a fraction of the distance for the first ring of elements. These are collapsed for 1/r or 1/sqrt(r) singularity.

5 is the number of rings between the 0.05 and 0.85 of the circle's raduis

2 are additional two rings of elements for transition between the circle and the original mesh

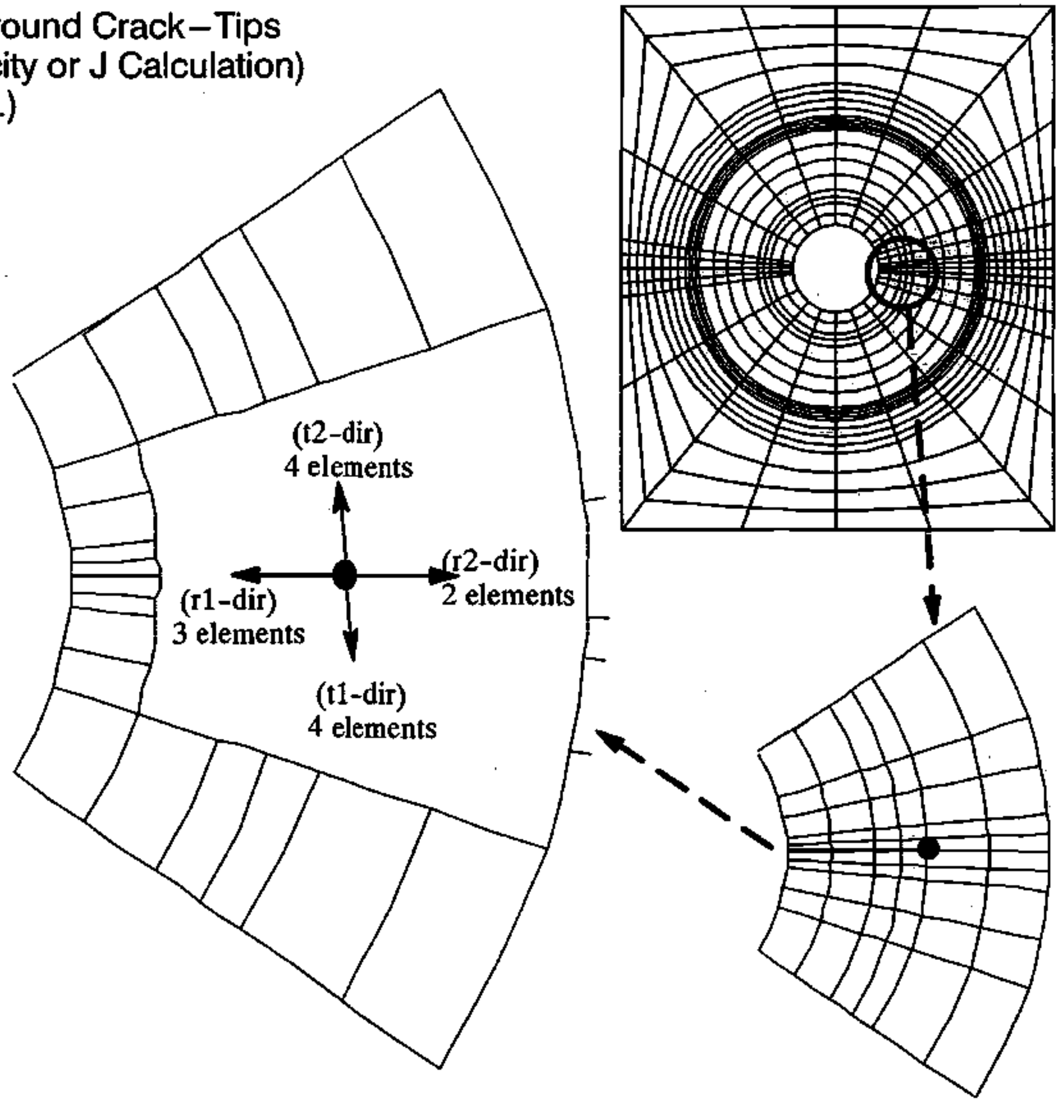


Figure 13. Input stage #9: refined meshes around crack tips (cont).

Input Stage #9:
Refined Meshes Around Crack–Tips
(Option–2: Plasticity or J Calculation)
(cont.)

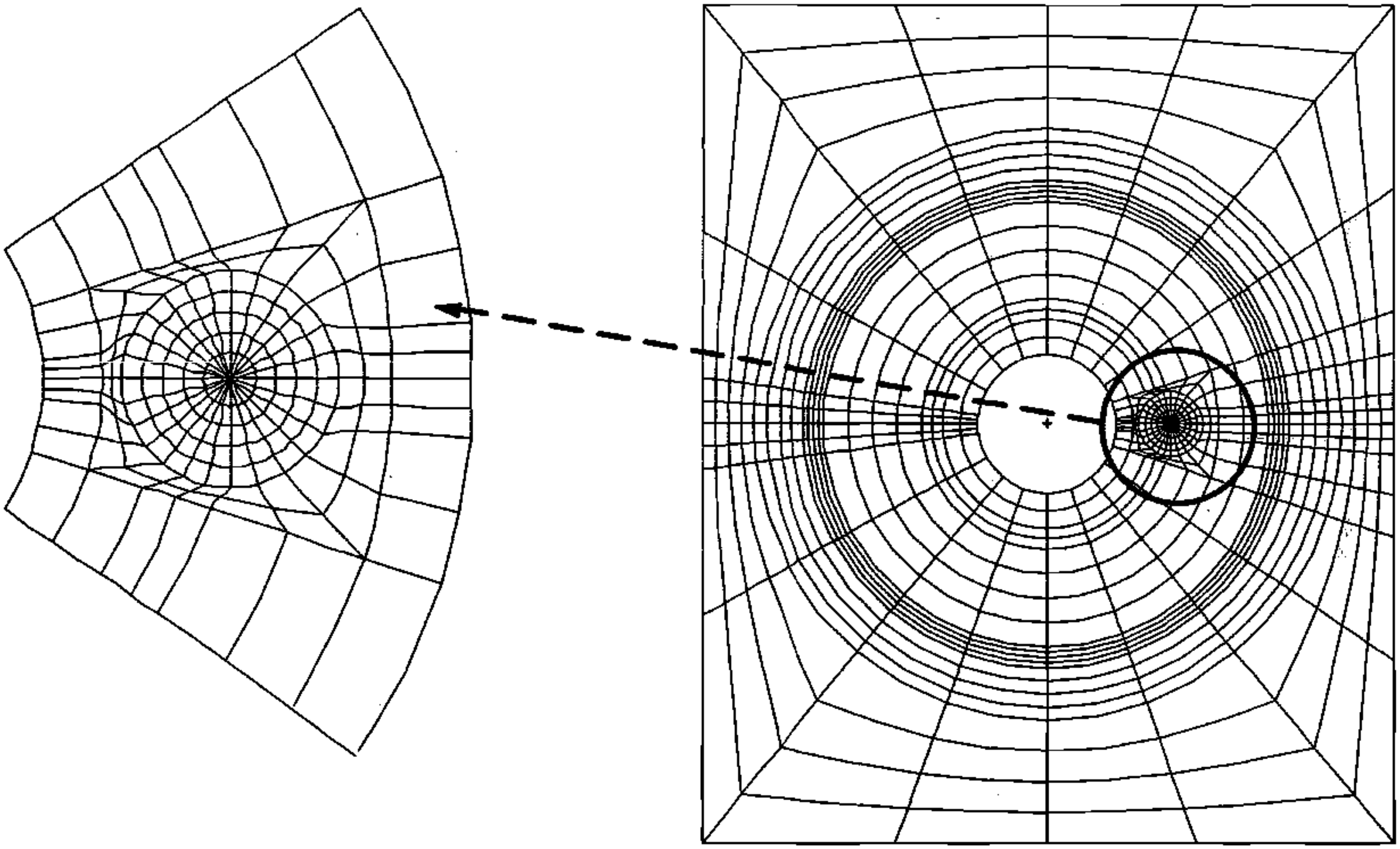
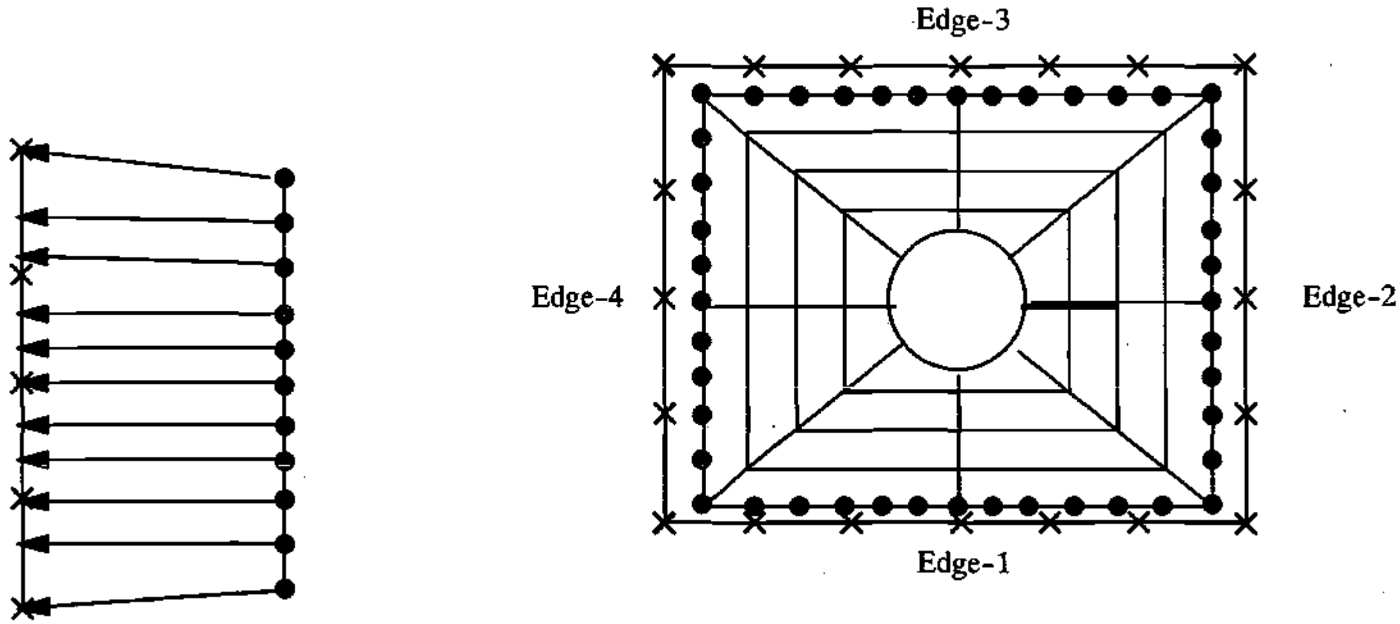


Figure 14. Input stage #9: refined meshes around crack tips (cont).

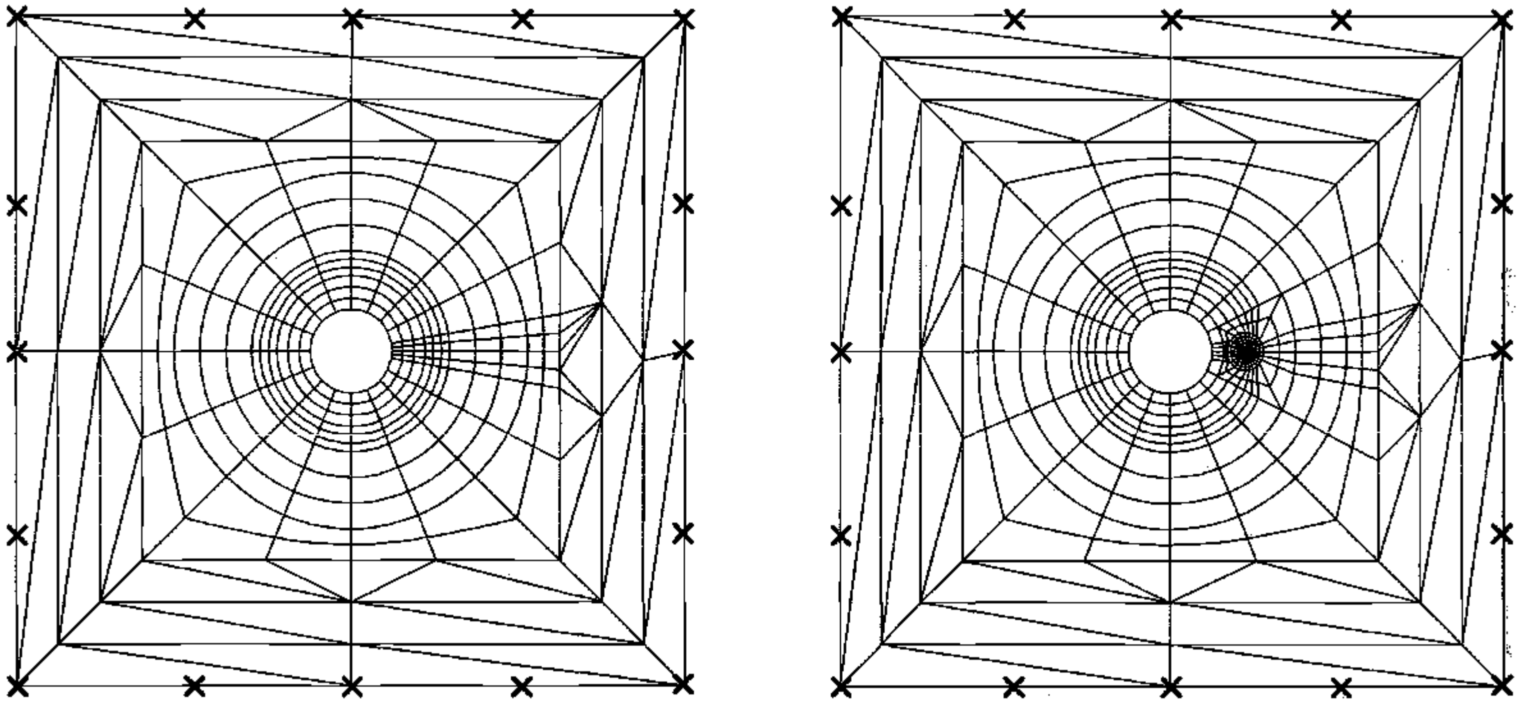
× - Edge node from the global coarse mesh

● - Edge node in CHM model



nodal constraints between any given edges

Figure 15. Local-global mesh connection through nodal constraints



X - Edge node from the global coarse mesh

Figure 16. Local-global transition examples using layers of triangular elements.

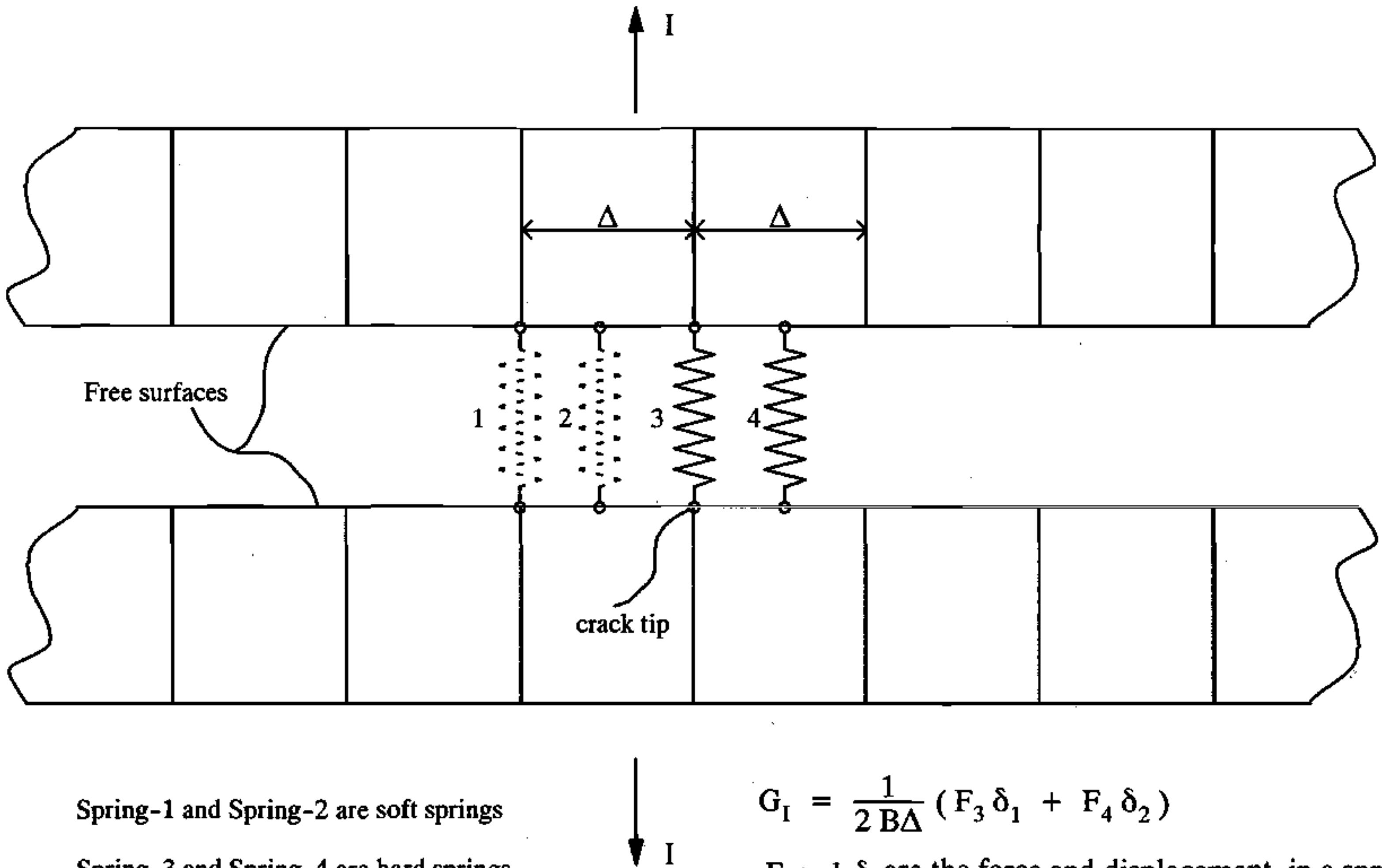
```
c----------------------------------------------------------------
c
c      Input to Cracked_hole Mesh Generator
c      ---------------------------------
c
c  Note: 1) Any line that starts with the letters
c           "c" or "C" is considered a comment line
c        2) "$" Character in this template means input
c        3) $FM -  Free format real input variable
c        4) $I  -  Free format integer input variable
c
c
c  Coordinate System:  X-Y (in-plane) coordinates.
c
c  The coordinate system is INITIALLY located at
c  the center of the hole.
c
c  1)  Geometry of the plate and hole
c
c  a = width   b = hight    D = Diameter of the hole
c
c
   $FM    $FM    $FM
c
c----------------------------------------------------------------
c
c  2) Input Additional Radial Lines (degrees)
c        that will be added to the default radial lines
c
c          Note:  These lines may or may NOT have
c                 radial cracks emanating from the
c                   edge of the hole.
c
c  Number of added radial lines
c
   $I1
c
c  Angle (degrees) associated with each radial line
c
c   IF $I1=0 comment the next input line
c
    $I1*($I)
c
c----------------------------------------------------------------
c
c  Crack Systems:
c
c  3)  How many crack systems in this mesh/model?
c
c  Num_cracks        IVCCT
c
    $I1              (0,$I)
c
c   For each crack system, enter the angle and
c   the radius of the crack tip (poolar coordinates)
c
c   Max Value of R_tip for each crack:
c   Max(R) < 0.85*Min(a/2,b/2)
c            (0.85 predifined factor)
c
c
c   IF $I1=0 comment the next input lines
c
c   Theta          R
c
    $FM1          $FM1
    $FM2          $FM2
    $FM3          $FM3
    $...          $...
c
c----------------------------------------------------------------
c
c  4)  Number of additional circular lines
c         (not due to crack tips)
c
c  these will add additional "radial intervals"
c
c Total Radial Intervals= Number of DIFFERENT
c  crack tips + num_add_r_int + 1
c
c  Num_add_r_intrv
c
   $I1
c
c   IF $I1=0 comment the next input line
c
c   R1 R2 R3 .... R(Num_add_r_intrv)
c
   $I1*($FM)
c
c----------------------------------------------------------------
c
c  5)  Mesh seeds in the angular direction.
c      All line segments in the circumferential
c      direction will have the same following
c       number of elements (> 2).
c
c  num_el_theta      ( > 2 )
c
    $I
c
c----------------------------------------------------------------
c
c  6)  Mesh seeds in the radial direction.
c
c  num_el_r
c
    $I
c
c----------------------------------------------------------------
c
c  7) Overwrite specifc number of seeds in the
c       radial segments within given ranges:
c                 R(i) < r > R(i+1)
c
c
c  num_el_rp
c
   $I1
c
c IF $I1=0 comment the next input lines in this step
c
c  r1   r2    num_el_r_mod
c
c  $I1 input lines
c
   $FM   $FM    $I
   $FM   $FM    $I
   $FM   $FM    $I
c
c----------------------------------------------------------------
```

Figure 17. Input template for the CHM program (1/2).

```
c
c  8)  Number of element to add in the last radial
c        segment between the last circular line
c          and the plate perimeter
c
c  num_el_add_r
c
    $I
c
c-------------------------------------------------------------
c
c  9)  Detailed Crack Tip Meshes (Y/N)
c        (Y/N) = (*/0)
c
c  num_det_tips
c
   $I1
c
c
c    theta    - Crack Angle which needs detailed tip meshing
c               the positive theta-direction.
c frac_rmax1 - [0+,1.0] Fraction which determines the Radius for the first
c               ring or circle as Rmax=fract_rmax*Min(Lr,Lt)
c frac_rmax2 - [0+,1.0] Fraction which determines the Radius for the last
c               circular ring ( fract_rmax2 > fract_rmax1 )
c
c num_el_tip_r1 -  Number of elements to delete from the crack tip in
c                  the negative r-direction.
c num_el_tip_r2 -  Number of elements to delete from the crack tip in
c                  the positive r-direction.
c num_el_tip_t1 -  Number of elements to delete from the crack tip in
c                  the negative theta-direction.
c num_el_tip_t2 -  Number of elements to delete from the crack tip in
c                  the positive theta-direction.
c   num_rings    -  Number of circular rings around the crack-tip
c
c IF $I1=0 comment the next input line
c
c  FOR EACH crack_tip input:
c
c  theta   fract_rmax1  fract_rmax2
c
    $FM  0.001<$FM<1.0    0.001< $FM <1.0
c
c num_el_tip_r1  num_el_ti_r2  num_el_tip_t1  num_el_tip_t2
c num_rings num_add_rings
c
     $I  $I  $I  $I    $I  $I
c
c  Repeat the above two lines of input $I1 times
c
c    Suggestion (may not always work)
c
c    0.   0.05   0.85
c    2   4  3  3  5  2
c
c ----------------------------------------------------------------------------
c
c 10) Input a number to add to the new nodes and elements
c
c
c    nd_shift    iel_shift
c
       $I          $I
c
```

```
c
c   Input the global x-y-z coordinate of the hole
c   (all nodes coordinates will be transformed from local
c    to global coordinates using this center point)
c
c
c     xg_center,yg_center,zg_center
c
        $FM        $FM          $FM
c
c ----------------------------------------------------------------------------
c
c 11) Transition Zone and connection to the global model
c
c        Iconnect = .eq. 0 - No connection to the global mesh is needed
c                   .else. - Perform connection with a given nodes set
c                            generated previously in a different FE model
c
c     Iconnect
c
       (0,$I1)
c
c  If above input is zero comment all lines in this input step
c
c
c   ASCII/patran filename for the nodes in the outside perimeter
c    (note: filename cannot start with the letter "c" or "C")
c
c
c        Inp_type = .eq.1  - ascii nodal file (node#,x,y,z)
c        Inp_type = .eq.2  - patran nodal set/group
c                            (patran file contains ONLY the edge nodes)
c        Note:  Both above files MUST have sequential node numbers
c
c    Inp_type
c
    $(1,2)
c
$file_name
c
c   Input the nodes for the 4-corners (vertex)
c
    $I1  $I1  $I1  $I1
c
c
c       IOPT= .eq.  1  -  Triangular transition mesh
c             .else.   -  Rigid link constraints
c
      $(1,I1)
c
c
c------------  IF( IOPT.ne.1) Then comment the next input line
c
c    Triangular transition zone
c
c       num_erow = Number of element rings between
c                  the two edges of node sets a and b. (>1)
c
c       red_fact = Reduction factor for the elements/nodes as going
c                  from inside to outside rings
c                           0. < red_fact <= 1.
c
c   num_erow   red_fact
c
      $I        0.001< $FM < 1.0
```

Figure 18. Input template for the CHM program (2/2).



Spring-1 and Spring-2 are soft springs

Spring-3 and Spring-4 are hard springs

$$G_I = \frac{1}{2\,B\Delta}\,(F_3\,\delta_1 + F_4\,\delta_2)$$

F and δ are the force and displacement in a spring

B- Thickness

Figure 19. Schematics showing the application of the VCCT to determine the ERR

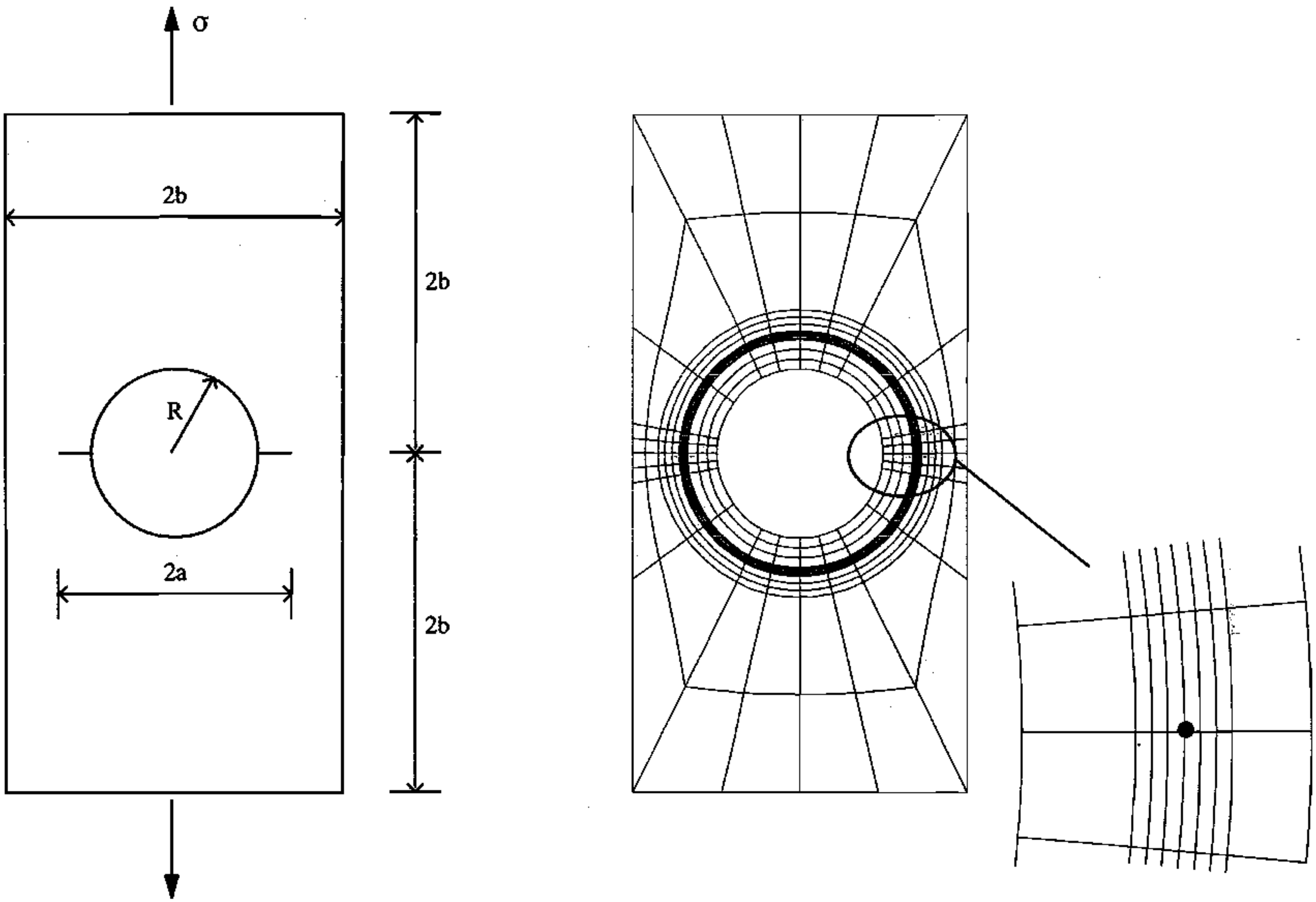


Figure 20. Verification problem for the VCCT and CHM modeling

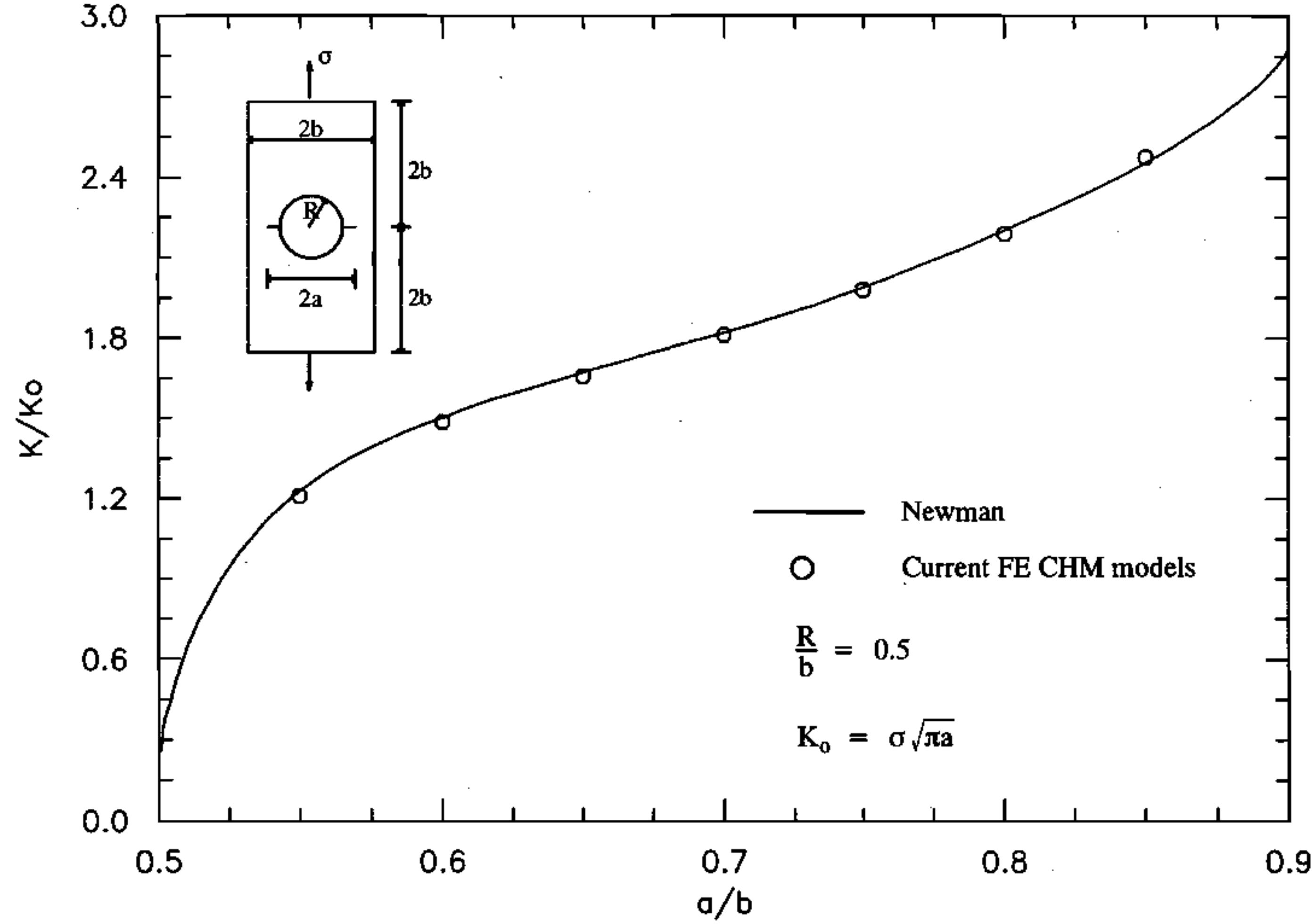


Figure 21. Variation of stress intensity factor for symmetric crack emanating from a notched hole.