University of Cincinnati

Date: 10/31/2014

I, Sili Wang, hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:
An ABAQUS Implementation of the Cell-based Smoothed Finite Element Method Using Quadrilateral Elements

Student's name:  Sili Wang

This work and its defense approved by:

Committee chair: Guirong Liu, Ph.D.
Committee member: Yijun Liu, Ph.D.
Committee member: David Thompson, Ph.D.
An ABAQUS Implementation of the Cell-based Smoothed Finite Element Method Using Quadrilateral Elements

A dissertation submitted to the
Graduate School of the University of Cincinnati
in partial fulfillment of the requirements
for the degree of

Master of Science
in Mechanical Engineering

in the Department of Mechanical & Materials Engineering
of the College of Engineering & Applied Science

By

Sili Wang
BS Safety Engineering, Beijing University of Chemical Technology

November 2014

Committee Chair: Dr. G.R. Liu
ABSTRACT

This thesis report a work on an implementation of a Cell-based Smoothed Finite Element method (CS-FEM) using quadrilateral element (Q4) in the framework of the commercial software package ABAQUS®. The CS-FEM is one of the Smoothed Finite Element Method (S-FEM) models introduced by Dr. G.R. Liu and his colleagues in recent years. Because smoothing domains used by CS-FEM are within the element, it is known as the model closest to the standard FEM. User-defined Element subroutine (UEL) feature was proposed to be used to implant the CS-FEM model into ABAQUS. In this paper, a couple of UELs are constructed respectively for 2-dimentional (2D) problems using Q4 elements. To implant CS-FEM into ABAQUS, a custom input file and corresponding user subroutine are developed. Then the implementation is verified using a number of linear elastic problems that has analytical solutions. In this article, details on data input file and the user element subroutine construction are provided, which contribute the key ingredients of this implementation. Several numerical examples utilizing the subroutines are presented to demonstrate the features and accuracy of the developed ABAQUS CS-FEM Q4 user element, by comparison with the analytical and the ABAQUS solutions.

Key words: Cell-based Smoothed Finite Element method (CS-FEM), ABAQUS/Standard, User-defined Element (UEL), FEM, Numerical methods.
ACKNOWLEDGMENTS

I would like to thank all my families and friends, especially my parents, for all the support they’ve given me during my study in University of Cincinnati. I could never have done it without knowing that they will always love me and will always be there for me. I love you all.

I would also like to express my gratitude towards my research advisor Dr. G.R. Liu. Without his guidance and patience, I would be so lost. I have learnt so much from him. His rigorous academic attitude is inspiring and honorable, and I shall benefit from it for the rest of my life. I would also like to thank Dr. David Thompson for his advice throughout my study in University of Cincinnati. I would also like to thank Dr. Yijun Liu for serving on my thesis committee.

I would also like to thank everybody in GR Lab for countless help and selfless support you’ve given me. I am extremely lucky to have all of you by my side to help me overcome the difficulties I encountered. Thank you all.

Last but not least, I would like to thank Ohio Supercomputer Center for providing access to ABAQUS user subroutine. The OSC technical support team has been extremely helpful.
## Contents

List of Table and Figures.................................................................................................................. vii

1 INTRODUCTION .............................................................................................................................. 1
  1.1 Introduction to S-FEM .................................................................................................................. 2
  1.2 Advantages of S-FEM .................................................................................................................. 4
  1.3 Motivation for the Thesis .............................................................................................................. 5
  1.4 Outline of the Thesis .................................................................................................................... 6

2 CELL-BASED SMOOTHEDFINITE ELEMENT METHOD .......................................................... 7
  2.1 Smoothed Finite Element Method ............................................................................................... 7
  2.2 Cell-based Smoothed Finite Element Method (CS-FEM) ....................................................... 8
    2.2.1 Problem domain discretization ............................................................................................... 9
    2.2.2 Construction of smoothing domain ....................................................................................... 9
    2.2.3 Shape function evaluation ................................................................................................... 10
    2.2.4 Formulation of CS-FEM [1] .................................................................................................. 11
    2.2.5 Recovery of strain and stress field ....................................................................................... 13

3 IMPLEMENTATION OF CS-FEM IN ABAQUS ................................................................. 14
  3.1 Constructing User-defined Element Subroutine for CS-FEM .................................................... 14
    3.1.1 Interface and passed in variables ....................................................................................... 15
    3.1.2 User-defined variables ....................................................................................................... 16
    3.1.3 Result variable ................................................................................................................... 16
  3.2 Constructing Input File for User-defined Elements ....................................................................... 17
  3.3 Post-processing for CS-FEM UEL Result ..................................................................................... 18

4 NUMERICAL EXAMPLES .................................................................................................................. 20
  4.1 Cantilever Beam (Plane Stress Example) .................................................................................... 20
    4.1.1 Problem description ........................................................................................................... 20
    4.1.2 Result analysis for regular mesh ......................................................................................... 22
    4.1.3 Result analysis for slightly distorted mesh ......................................................................... 28
    4.1.4 Result analysis for heavily distorted mesh ......................................................................... 32
  4.2 Infinite Plate with Hole (Plane Strain Example) .......................................................................... 34
    4.2.1 Problem description ........................................................................................................... 34
4.2.2 Result analysis for regular mesh .......................................................... 38
4.2.3 Result analysis for slightly distorted mesh ........................................ 43
4.2.4 Result analysis for heavily distorted mesh ........................................ 48
4.3 Semi-infinite Plane (Plane Strain Example) ............................................ 50
  4.3.1 Problem description ........................................................................... 50
  4.3.2 Result analysis for regular mesh ...................................................... 54
  4.3.3 Result analysis for slightly distorted mesh ....................................... 60
  4.3.4 Result analysis for heavily distorted mesh ....................................... 66
5 SUMMARY ................................................................................................. 69
REFERENCES .............................................................................................. 70
Appendix A: CS-FEM UEL .......................................................................... 73
Appendix B Sample of Input File ................................................................. 83
Appendix C: MATLAB Post-processing Program ........................................ 86
List of Table and Figures

Table 1 [1] Shape function values of all location in 4SD CS-FEM element (Fig. 2.2) .............12

Figure 2.2.1 [1] Division of a Q4 element into different numbers of smoothing domains in CS-
FEM models: (a) SD=1 (b) SD= 2 (c) SD=3 (d) SD=4 (e) SD=8 (f) SD=16 ........10
Figure 2.2.2 [1] CS-FEM element (SD=4) notations .............................................................11
Figure 4.1.1 Cantilever beam ................................................................................................21
Figure 4.1.2 Domain discretization of the cantilever using 4-noded quadrilateral elements:
(a) Regular mesh; (b) Slightly distorted mesh; (c) Heavily distorted mesh ........23
Figure 4.1.3 (a) Displacement results of U2 (m) at y=0 (m);
(b) Error (%) of U2 (m) at y=0 (m). ..................................................................................25
Figure 4.1.4 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at x=0.6 (m);
(b) Error (%) at x=0.6 (m). ................................................................................................26
Figure 4.1.5 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) results at x=0.6 (m);
(b) Error (%) at x=0.6 (m). ................................................................................................27
Figure 4.1.6 (a) Displacement results of U2 (m) at y=0 (m);
(b) Error (%) of U2 (m) at y=0 (m). ..................................................................................29
Figure 4.1.7 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at x=0.6 (m);
(b) Error (%) at x=0.6 (m). ................................................................................................30
Figure 4.1.8 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) results at x=0.6 (m);
(b) Error (%) at x=0.6 (m). ................................................................................................31
Figure 4.1.9 Displacement results U2 (m) at y=0 (m). ............................................................33
Figure 4.1.10 Normal stress $\sigma_{xx}$ (N/m$^2$) results at x=0.6 (m). .....................................33
Figure 4.1.11 Shear stress $\sigma_{xy}$ (N/m$^2$) results at x=0.6 (m) .................................................................34

Figure 4.2.1 Infinite plate with a circular hole subjected to x direction tension .........................35

Figure 4.2.2 Quarter model with size of 5m×5m with symmetric boundary conditions at left and bottom edges.........................................................................................36

Figure 4.2.3 Domain discretization of the quarter model:
(a) Regular mesh; (b) Slightly distorted mesh; (c) Heavily distorted mesh............37

Figure 4.2.4 (a) Displacement U1(m) results at bottom boundary, y = 0(m);
(b) Error (%) at bottom boundary, y = 0(m) .................................................................39

Figure 4.2.5 (a) Displacement U2(m) results at left boundary, x = 0(m);
(b) Error at left boundary, x = 0(m) .................................................................40

Figure 4.2.6 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) results at bottom boundary, y=0 (m);
(b) Error at bottom boundary, y=0 (m) .................................................................41

Figure 4.2.7 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at left boundary, x=0 (m);
(b) Error at left boundary, x=0 (m) .................................................................42

Figure 4.2.8 (a) Displacement U1(m) results at bottom boundary, y = 0(m);
(b) Error (%) at bottom boundary, y = 0(m) .................................................................44

Figure 4.2.9 Displacement U2(m) results at left boundary, x = 0(m);
(b) Error at left boundary, x = 0(m) .................................................................45

Figure 4.2.10 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) results at bottom boundary, y=0 (m);
(b) Error at bottom boundary, y=0 (m) .................................................................46

Figure 4.2.11 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at left boundary, x=0 (m);
(b) Error at left boundary, x=0 (m) .................................................................47

Figure 4.2.12 Displacement U1(m) results at bottom boundary, y = 0(m) .................................48
Figure 4.2.13 Displacement U2(m) results at left boundary, x = 0(m).

Figure 4.2.14 Normal stress $\sigma_{yy}$ (N/m$^2$) results at bottom boundary, y=0 (m).

Figure 4.2.15 Normal stress $\sigma_{xx}$ (N/m$^2$) results at left boundary, x=0 (m).

Figure 4.3.1 Semi-infinite plane subjected to a limited range uniform pressure.

Figure 4.3.2 Domain discretize of the semi-finite plane problem domain:
   (a) Regular mesh; (b) Slightly distorted mesh; (c) Heavily distorted mesh.

Figure 4.3.3 (a) Displacement U1(m) results on the top free boundary, y=0(m);
   (b) Error (%) on the top free boundary, y=0(m).

Figure 4.3.4 (a) Displacement U2(m) results on the top free boundary, y=0(m);
   (b) Error (%) on the top free boundary, y=0(m).

Figure 4.3.5 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) along diagonal line (y=-x);
   (b) Error (%) along diagonal line (y=-x).

Figure 4.3.6 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) along diagonal line (y=-x);
   (b) Error (%) along diagonal line (y=-x).

Figure 4.3.7 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) along diagonal line (y=-x);
   (b) Error (%) along diagonal line (y=-x).

Figure 4.3.8 (a) Displacement U1(m) results on the top free boundary, y=0(m);
   (b) Error (%) on the top free boundary, y=0(m).

Figure 4.3.9 (a) Displacement U2(m) results on the top free boundary, y=0(m);
   (b) Error (%) on the top free boundary, y=0(m).

Figure 4.3.10 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) along diagonal line (y=-x);
   (b) Error (%) along diagonal line (y=-x).
Figure 4.3.11 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) along diagonal line (y=-x);

(b) Error (%) along diagonal line (y=-x).......................................................64

Figure 4.3.12 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) along diagonal line (y=-x);

(b) Error (%) along diagonal line (y=-x)..................................................................65

Figure 4.3.13 Displacement $U_1$(m) results on the top free boundary, y=0(m). ..................66

Figure 4.3.14 Displacement $U_2$(m) results on the top free boundary, y=0(m). ....................67

Figure 4.3.15 Normal stress $\sigma_{xx}$ (N/m$^2$) along diagonal line (y=-x). ..........................67

Figure 4.3.16 Normal stress $\sigma_{yy}$ (N/m$^2$) along diagonal line (y=-x). ..........................68

Figure 4.3.17 Normal stress $\sigma_{xy}$ (N/m$^2$) along diagonal line (y=-x). ..........................68
1 INTRODUCTION

Ever since the birth of the first computer, the development of technology has been increasingly rely on the assistant of various computer programs. After generations and generations of revolution, computers nowadays are helping all professions and trades in ways that were unimaginable for us decades ago. For engineers, computers are certainly essential tools in countless procedures in every day’s routine. In product and engineering system design, computers are frequently used in designing, modeling, simulation, testing, construction, fabrication, etc. [1]. For modeling and simulation procedures, several numerical and computational methods have been developed over time, and Finite Element Method (FEM) [2, 3 and 21] has been developed into the most important technology of various engineering systems [2]. Correspondingly, commercial FEM packages are now widely used in basically all branches of engineering to do structures, solids, and fluids analysis [3]. However, as every other existence on earth, FEM is not perfect yet. Some restrictions in FEM are difficult to meet in real life simulation procedures and might cause inaccurate results. Encouraged by such reason, engineers and scholars have always been searching for more effective methods. Different, yet similar from the well-known FEM, Smoothed Finite Element Methods (S-FEM) were proposed in the past few years by Dr. G.R. Liu and his colleagues, and their work on S-FEM are well explained in details in Ref. [1].

In this article, a couple of user subroutines have been constructed in order to implant one of the S-FEM models into the well-developed and widely used commercial code. The S-FEM has a
number of models including the Cell-based Smoothed Finite Element Methods (CS-FEM) [10],
Node-based Smoothed Finite Element Methods (NS-FEM) [11, 12], Edge-based Smoothed
Finite Element Methods (ES-FEM) [13, 14] and Face-based Smoothed Finite Element Methods
(FS-FEM) [15]. In this paper, CS-FEM model was chosen to be implanted in to ABAQUS. This
is because it uses smoothing domains within the element, and hence it can be implemented
without using information of the neighboring elements (which is a must for NS-, ES- and FS-
FEM). It is known as the model closest to the standard FEM, and the User-defined Element
subroutine (UEL) feature in the ABAQUS® can be used for this work. In this implementation,
ABAQUS/CAE 6.11-2 is the specific version that is used for this issue. Note that some
commands and features of ABAQUS might vary from version to version, thus the techniques
proposed in this paper regarding UEL usage is not guaranteed to be applicable in all other
ABAQUS versions.

1.1 Introduction to S-FEM

In traditional FEM modeling, the so-called compatible strain field is being used directly, which is
acquired by the strain-displacement relation and the displacement field assumption. However, in
an S-FEM model, the compatible strain field will be modified after the evaluation; or a strain
field can be simply obtained using the assumed displacement field without the evaluation of the
compatible strain field [1]. This step contributes to a softer model with more accuracy, and
makes it free from several FEM modeling restrictions. The key idea of S-FEM is to modify the
compatible strain field, or construct a strain field using only the displacements. Modifications or
constructions as such can be performed within or beyond elements depends on the choice of
element type. In most of the cases, this is accomplished beyond elements, which means information of neighboring elements is required when constructing the element matrices. In this methodology, strain smoothing technique is used to modify or construct the strain field. Accordingly, smoothed Galerkin weak form [16] is constructed in order to guarantee stability and convergence [1]. Interested readers are referred to Ref. [1, 16] for more details. The S-FEM models are all work very well with triangular elements that can be generated automatically and hence is ideal for future automation in computation and adaptive analyses. In addition, the S-FEM models are theoretically proven “softer” than the standard FEM counterpart, which overcomes the so-called “overly stiff” feature in the FEM [1].

During strain smoothing procedure, the entire problem domain is divided into a number of no-gap and no-overlapping smoothing domains in order to modify or construct the compatible strain field. There are several ways to generate the smoothing domains, which result in different smoothing domain types. On account of different types of smoothing domain used in the models, there are four different kinds of S-FEM models existing. Smoothing domains based on cells (elements), nodes, edges, and faces are using to establish four corresponding S-FEM models: Cell-based S-FEM (CS-FEM), Node-based S-FEM (NS-FEM), Edges-based S-FEM (ES-FEM) and Face-based S-FEM (FS-FEM) [1, 10-15]. Note that these S-FEM models will have different features and properties which lead to different advantages and disadvantages. The NS-FEM is known as a soft model and hence can produce upper bound solution (when certain conditions are satisfied), and the ES-FEM is known as ultra-accurate and work well for static, dynamic, linear and nonlinear problem [1]. The CS-FEM using Q4 elements is proven always softer than the FEM counterpart, and much more robust against the mesh distortion, which is one of the most
critical problem with the standard FEM. In this particular article, only the CS-FEM model is
implanted into the commercial code ABAQUS for general static analysis with plane stress and
plane strain loading conditions due to previously stated reason.

1.2 Advantages of S-FEM

Although FEM is well-known in the engineering world and a variety of FEM commercial
packages are available in the market nowadays, there are several reasons explaining why S-FEM
models are worthy [1]:

a. Overly Stiff Issues. It is well-known that so-called locking behavior caused by “overly
   stiff” phenomenon can be found for all standard finite elements, especially for linear
   triangular elements. However, S-FEM models can solve this problem and can always
   provide a softer model.

b. Stress Accuracy Issues. This is another issue typically found in FEM models, especially
   when linear triangular and tetrahedral meshes are used. Strain field becomes
   discontinuous in the process of strain evaluation, which results in poor solutions in the
   stress field. The S-FEM models can offer ways to treat the discontinuities and improve
   both displacement and stress solution.

c. Mesh Distortion Issues. FEM strongly requires mesh quality in order to get a better result.
   S-FEM models naturally fix this problem, because no mapping is required of all the
   models for all types of elements, therefore, no Jacobian matrix is needed during
   formulation which eliminate the issues related to bad meshing.
d. Meshing Issues. When FEM models use triangular and tetrahedral elements, it tends to give out poor solutions. But for complicated geometries, it is always preferred to use triangular and tetrahedral elements. Fortunately, some S-FEM models can give out satisfying results with triangular and tetrahedral meshing.

e. Solution Certificate. While standard FEM can only give the lower bound solution, some of the S-FEM models can produce upper bounds, which provides an important closure for numerical solution bounds.

f. Computational Efficiency. Among all other type of methods might be able to solve the above problems, S-FEM models clearly have the advantages in computational efficiency and simplicity.

g. Lower-Order Elements. Lower-Order Elements are a preferred choice for several reasons such as calculation time, simplicity, range of application, etc. And S-FEM models are certainly good for lower-order elements.

1.3 Motivation for the Thesis

While a variety of FEM commercial packages existing in the market, only a hand full of S-FEM computer programs are available. And among the limited numbers of S-FEM codes, the majority of the packages are individually written original codes (not dependent on well-developed commercial codes). Most of the packages contain numerous subroutines which limited the mobility and workability of the S-FEM program. And imbedding S-FEM into a well-developed commercial code, on the one hand, reduces the workload of developing S-FEM program; on the other hand, increases the mobility and workability of S-FEM programs. Therefore, author was
motivated by such reason to complete an implementation of one of the S-FEM models into ABAQUS. As mentioned earlier in the article, there are four different types of S-FEM models. Once again, CS-FEM was picked to do the implementation because it is known as the closest to the standard FEM. And also, it is the only S-FEM model that can be realized by the User-defined Element subroutine (UEL) feature in the ABAQUS®, since NS-, ES- and FS-FEM models require neighboring element information, which is not provided by UEL, in order to generate smoothing domains.

1.4 Outline of the Thesis

Section 1 gives out a general introduction about the paper in addition with a brief introduction to Smoothed Finite Element Methods (S-FEM). Section 2 provides the detailed formulation of Cell-based Smoothed Finite Element Method (CS-FEM), which is chosen in this paper to do the implementation in ABAQUS. Section 3 detailed explains the usage of User-defined Element (UEL) subroutine in ABAQUS, including the construction details of input file and user subroutine. Section 4 presents several numerical examples with analytical solutions. Then the results from CS-FEM UEL for each numerical example are compared with both ABAQUS Q4 element results and analytical in order to validate the UEL subroutine and to show the feature of CS-FEM model. Section 5 summarizes the conclusions obtained from section 4 and discusses about possible future works and potential problems on this subject.
2 CELL-BASED SMOOTHED FINITE ELEMENT METHOD

Cell-based Smoothed Finite Element Method (CS-FEM) model was the first S-FEM model. In fact, it was originally called S-FEM itself. Now, S-FEM generally refers to all types of S-FEM models. This section will give out a brief introduction to S-FEM general procedure followed by a more detailed explanation of the CS-FEM model. Section 2 is mostly with reference to Ref. [1].

2.1 Smoothed Finite Element Method

As mentioned earlier in this article, S-FEM models have many advantages compare to standard FEM. It gives out a more accurate solution for triangular elements (T3) and tetrahedron elements (T4), and allows users to discretize the problem domain into n-sided polygonal elements. Following are the brief introduction to the general procedure for S-FEM models. Interested readers are referred to Ref. [1] for more detailed explanation.

Step 1: Discretize the problem domain into elements. In this step, if T3, T4 or Q4 (quadrilateral elements) elements are used, meshing can be performed as same as the way in FEM. If n-sided elements are used, one technique can be performed to discretize the problem domain using the Voronoi diagram, which is more detailed explained in Ref. [5].

Step 2: Create a displacement field via the construction of shape function. In this step, if constant strain elements such as T3 or T4 are used, S-FEM can use the same shape functions from FEM directly. However, the construction of shape functions for S-FEM models in general is different from FEM. In S-FEM models, no derivatives of the shape functions are needed, only the shape function values at locations on the smoothing domain boundaries are required to create
the S-FEM models. To get the shape functions for all types of S-FEM models, an approach called “linear PIM”, which is the simplest case of the general “PIM” [6], is used.

Step 3: For all types of the S-FEM models, this step can be skipped over. However, when T3 or T4 elements are used, evaluation of the compatible strain field for the problem domain can be done in the same way as it is in FEM during this step.

Step 4: Construct the smoothed strain field for all element types, or modify the compatible strain field from last step for T3 or T4 element types. This step exists in S-FEM only. To construct the smoothed strain field, simple surface integration using shape function values on local smoothing domain boundaries is performed. No coordinate mapping is needed to complete this step.

Step 5: Establish the discrete linear algebraic system of equations using “smoothed Galerkin” weak form and the assumed displacement and smoothed strain fields. S-FEM only requires a simple summation over the entire smoothing domains.

Step 6: Impose boundary conditions and solve the algebraic system of equations in order to obtain the displacement results. This step is basically the same as it is in FEM.

Step 7: Recover the strain field if required by the analysis. Similar yet different procedure needs to be carried out to complete this task.

Step 8: As same as in FEM, assess the result of the analysis.

2.2 Cell-based Smoothed Finite Element Method (CS-FEM)

Among all four types of S-FEM elements, CS-FEM was chosen to do the implementation into ABAQUS. Until now, two CS-FEM models have been developed: CS-FEM and nCS-FEM. In
CS-FEM model, bilinear quadrilateral (Q4) elements with quadrilateral smoothing domains are used. It is a model works perfectly for static and dynamic solid mechanics problems [10, 17 and 18]. And the Q4 based CS-FEM model is the default model for CS-FEM. Therefore, details regarding nCS-FEM for n-sided polygonal element will not be discussed in this article; interested readers are referred to Ref. [19] for more details. Followed are the detailed explanation of the theory and usage of CS-FEM model.

2.2.1 Problem domain discretization
As mentioned above, CS-FEM elements are based on Q4 elements. Thus, the discretization of the problem domain in FEM for Q4 elements can be used directly in CS-FEM models.

2.2.2 Construction of smoothing domain
As in the standard FEM-Q4 discretization procedure, CS-FEM divided the problem domain $\Omega$ into $N_{el}$ quadrilateral elements. Then each element $\Omega_{el}^{el}$ will be divided into $n_{el}^{sd} \in [1, \infty)$ quadrilateral smoothing domains in a no-cap and no-overlapping method which gives out $N_s = N_{el} \times n_{el}^{sd}$ number of smoothing domain in the entire problem domain. There are quite a few ways to achieve such subdivision, and the easiest way is connecting the mid-boundary-points of opposite boundaries to form smoothing domains as shown in Fig. 2.1 [1], this method also helps to preserve the bilinear feature on the boundaries of smoothing domains. In theory, subdivision of smoothing domains for a Q4 element can be as many as shown in Fig. 2.1. However, excessively divide elements into large number of smoothing domains is unnecessary and expensive. A more preferable way to divide a Q4 element for solid mechanics is as shown in Fig. 2.1 d (i.e. $n_{el}^{sd}=4$), proved by the numerical example showed in Ref. [1]. Hence, the default way
to form smoothing domains for a CS-FEM bilinear Q4 element is to divide the element domain $\Omega^e_l$ into four smoothing domains (i.e. 4SDs) as shown in Fig. 2.1 d.

![Diagram of Q4 element division into different numbers of smoothing domains in CS-FEM models](image)

**Figure 2.2.1 [1] Division of a Q4 element into different numbers of smoothing domains in CS-FEM models: (a) SD=1 (b) SD= 2 (c) SD=3 (d) SD=4 (e) SD=8 (f) SD=16.**

2.2.3 *Shape function evaluation*

As mentioned earlier, when using continuous linear displacement field along the smoothing domain edges, the matrix of smoothed strain-displacement can be evaluated by providing shape function values at mid-segment-points (Gauss points) on each of the smoothing domain boundary segments will be explained in the following sections. And the values of shape functions at each Gauss point is simply obtained by averaging two end points of the segment supporting the corresponding Gauss point. For example, for an arbitrary CS-FEM element with four smoothing domains as shown in Fig. 2.2, shape function values of Gauss point g7 are obtained by averaging shape function values of points #9 and #6, Gauss point g4 are obtained by averaging shape function values of points #1 and #8. Therefore, the shape function values of points #1 to #9 are necessary in the purpose of evaluating shape function values of Gauss point g1 to g12 on all 12 boundary segments of the total four smoothing domains. The notation of
nodes, Gauss points and smoothing domain for an arbitrary CS-FEM 4SD element is specified in Fig. 2.2.

![Figure 2.2](image)

**Figure 2.2 [1] CS-FEM element (SD=4) notations.**

In Ref. [1], the value of shape functions at all locations, including the field nodes, added nodes and Gauss points, are provided in Table 1 [1].

2.2.4 **Formulation of CS-FEM [1]**

All the formulations from this section can be found in Ref. [1]. By using the general formulation procedures of S-FEM models, the global smoothed stiffness matrix $K$ can be constructed. Its entries are given by

$$K_{ij} = \sum_{k=1}^{N_s} B^T_i c B_j A_{kr}^s$$

(Eq. 2.1)

in which $A_{kr}^s$ is the area of the quadrilateral smoothing domain, $N_s$ is the total number of smoothing domain in the entire problem domain. And each term of $K_{ij}$ can be easily embedded into the global smoothed stiffness matrix $K$, without any matrix assembling procedures.
Table 1 [1] Shape function values of all location in 4SD CS-FEM element (Fig. 2.2).

<table>
<thead>
<tr>
<th>Point</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Field node</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>Field node</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
<td>Field node</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>Field node</td>
</tr>
<tr>
<td>5</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>Side midpoint</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>Side midpoint</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>Side midpoint</td>
</tr>
<tr>
<td>8</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>Side midpoint</td>
</tr>
<tr>
<td>9</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>Intersection of two bimedians</td>
</tr>
<tr>
<td>g1</td>
<td>3/4</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g2</td>
<td>3/8</td>
<td>3/8</td>
<td>1/8</td>
<td>1/8</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g3</td>
<td>3/8</td>
<td>1/8</td>
<td>1/8</td>
<td>3/8</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g4</td>
<td>3/4</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g5</td>
<td>1/4</td>
<td>3/4</td>
<td>0</td>
<td>0</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g6</td>
<td>0</td>
<td>3/4</td>
<td>1/4</td>
<td>0</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g7</td>
<td>1/8</td>
<td>3/8</td>
<td>3/8</td>
<td>1/8</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g8</td>
<td>0</td>
<td>1/4</td>
<td>3/4</td>
<td>0</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g9</td>
<td>0</td>
<td>0</td>
<td>3/4</td>
<td>1/4</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g10</td>
<td>1/8</td>
<td>1/8</td>
<td>3/8</td>
<td>3/8</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g11</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>3/4</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
<tr>
<td>g12</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
<td>3/4</td>
<td>Gauss point (mid-segment-point of $\Gamma_{k,p}^s$)</td>
</tr>
</tbody>
</table>
And the strain-displacement matrix $B_l$ is computed by

$$B_l = \begin{bmatrix} b_{lx} & 0 \\ 0 & b_{ly} \\ b_{ly} & b_{lx} \end{bmatrix} \quad \text{(Eq. 2.2)}$$

$$b_{lh} = \frac{1}{A_k} \sum_{p=1}^{n^S_k} n_{h,p} N_l(x_p^G) l_p, \quad h = x, y. \quad \text{(Eq. 2.3)}$$

where $A_k^S$ is the area of the smoothing domain; $n^S_l$ is the total number of boundary segments, in the case of SD=4, $n^S_l = 4$; $n_{h,p}$ is the outward unit normal of the boundary segment; $x_p^G$ is the Gauss point (midpoint) of the segment; $l_p$ is the length of the boundary segment. Therefore, the global smoothed stiffness matrix can be constructed via simple evaluation without computing shape function derivative.

### 2.2.5 Recovery of strain and stress field

In the S-FEM models, strains and stresses at nodes are evaluated in a way different from standard FEM. In S-FEM models, strains and stresses at node ‘j’ is the area-weighted average value of all the smoothing domains around node ‘j’. The nodal strain can be computed by [1]

$$\epsilon(x_j) = \frac{1}{A_j^{ns}} \sum_{k=1}^{n^S_j} \epsilon_k A_k^{S}, \quad \text{(Eq. 2.4)}$$

in which $n^S_j$ is the number of smoothing domains around node ‘j’; $A_j^{ns}$ is the total area of smoothing domains around node ‘j’; $\epsilon_k$ and $A_k^{S}$ are the strain and area of individual smoothing domain around node ‘j’.
3 IMPLEMENTATION OF CS-FEM IN ABAQUS

In this section, main feature of ABAQUS user subroutine UEL [4] relates to implementation CS-FEM into ABAQUS is introduced. The main function of ABAQUS UEL is that the user is allowed to use any numerical approach to evaluate the element matrices. However, it does have certain restrictions when come to actual cases. The one restriction have the most impact on S-FEM element implementation is that only current element information is passed into the user subroutine, no neighboring elements information is available when constructing element matrices. As mentioned earlier, NS-, ES- and FS-FEM element types require neighboring elements’ information to form smoothing domain in order to create the element or overall matrices. For those element types, UEL might not be an appropriate choice to accomplish the implementation. Fortunately, CS-FEM models do not require such information to form smoothing domains. Thus, it is feasible to use UEL to imbed CS-FEM into ABAQUS. UEL can be used in either ABAQUS command line or ABAQUS/CAE. In this paper, only the usage of UEL in ABAQUS/CAE will be discussed in details. To use UEL in ABAQUS/CAE, two fundamental files are needed: input file (.inp file), user subroutine (.f file). Details on constructing these two files will be provided.

3.1 Constructing User-defined Element Subroutine for CS-FEM

To use the User-defined Element (UEL) feature in ABAQUS, a corresponding UEL subroutine needs to be constructed. UEL is flexible yet still limited in many aspects. For example, as mentioned earlier, for 4SD CS-FEM element type, UEL allows users to use the numerical
algorithm of CS-FEM to generate the user-defined smoothed element stiffness matrix. According to the pass-in variables of UEL, all the information needed in constructing CS-FEM element type is accessible such as nodal coordinates, material properties, element number in the overall problem domain, etc. However, for other S-FEM element types such as ES-FEM, the same sort of information of neighboring elements is needed to construct the element stiffness matrix, and none of such information of neighboring elements is accessible in UEL. Fortunately, it is possible to obtain all the variables needed to construct 4SD CS-FEM elements since the CS-FEM smoothing domains falls inside the element range. Appendix A is the entire UEL subroutine for 4SD CS-FEM element type. The current UEL subroutine is constructed in the language of FORTRAN and saved as an .f file as required by ABAQUS CAE solver. The ABAQUS CAE solver used in this specific paper is ABAQUS/CAE 6.11-2.

3.1.1 Interface and passed in variables
When constructing the UEL subroutine, the specific interface in Appendix A needs to be used [4]. Passed-in variables used in this specific UEL subroutine for 4SD CS-FEM element are COORDS, PROPS, U and JELEM. COORDS is a two-dimensional array containing current element’s un-deformed global nodal coordinates, which is essential for smoothed stiffness matrix calculation. The sequence of node coordinates in this array follows the element-node connectivity information provided in the input file (.inp file). PROPS is a one-dimensional array contains the material properties for the user defined element. The length and the value of each term of this array are defined in the input file (.inp file). U is the result displacement array of the current element nodes from last step of calculation, which is used to calculate the strains and
stresses of each smoothing domain. JELEM is the element number in the global problem domain, which is used for output reason only.

3.1.2 User-defined variables

UEL requires the user to define at least three variables in the user subroutine, RHS, AMATRX and SVARS. In this specific case, RHS is the residual force vector of the current element. AMATRX is the definition of stiffness matrix of the current element. In this implementation, when constructing AMATRX for the user element, Eq. 2.1 - Eq. 2.3 is used to evaluate the element smoothed stiffness matrix. Though CS-FEM take a different approach to construct element stiffness matrix, the constitutive matrix for plane stress and plane strain problems are no different than those in FEM, which can be easily found in Ref. [2, 3 and 20]. SVARS is supposed to be the solution dependent variables that can be calculated from last step solutions and get updated at the end of current increment. It can also be output by ABAQUS [4]. Since a different output method is used in this user subroutine, the SVARS array is assigned to be the smoothing domain areas to avoid errors in jobs. Apart from the passed in variables used to define the element stiffness matrix, all the other variables required to calculate the smoothed stiffness matrix, such as shape function values of Gauss points, coordinates of middle-edge points, etc., are either defined or calculated at the very beginning of the user subroutine (See Appendix A).

3.1.3 Result variable

For post-processing reasons, a two-dimensional array called ENSNA is assembled in the user subroutine for output purpose. In matrix term, ENSNA has seven rows and four columns. Each
column contains strains, stresses and area results of corresponding smoothing domain. The data arrangement of ENSNA for each element will be

\[
[\varepsilon_{11}^i; \varepsilon_{12}^i; \varepsilon_{12}^i; \sigma_{11}^i; \sigma_{12}^i; \sigma_{22}^i; A_i]_{7 \times 4} (i = 1, 4)
\]

in which \( i \) is the index number of \( i \)-th smoothing domains in the element.

3.2 Constructing Input File for User-defined Elements

Input file contains modeling information. It tells ABAQUS everything about the model that needs to be analyzed. There are basically two ways to generate an input file: (1) using input file keywords and corresponding parameters to write the input file, keywords instructions can be found in Ref. [9]; (2) using ABAQUS CAE to construct a CAE model and job of the problem and use the ‘write input file’ option to obtain the input file for the corresponding model.

However, writing input file from scratch with keywords for a large model can be time consuming and ineffective. It is more acceptable to construct the problem model in ABAQUS/CAE, or any other pre-processing software compatible with ABAQUS, and modified the input file in a way that is desired to use the User-defined Element subroutine.

Before going any further about input file, it is necessary to point out that, in the writer’s own experience, ABAQUS tend not to recognize geometries when UEL subroutine is used. This is also the reason why ABAQUS cannot plot any UEL results. Geometry definition keywords such as ‘*Surface’ in the input file seems to be ineffective. Therefore, all loading conditions and boundary conditions are recommended to define on nodes or node sets. For the same reason, it is recommended by the writer that, if the same input file construction method is used as in this
article, it is safer (might not be absolutely necessary) to create the mesh independent from the model part.

Appendix B gives out an input file template for reference. In #1 section of input file, '*User element' keyword line is used to define a user element type called U104 with four nodes, three material properties and nodal coordinates of two. Followed data line suggests the 1 and 2 degree of freedom (i.e. x and y) is activated for this element. The ‘*Element’ keyword line suggests that the following elements generated will be of the U104 User-defined Element type, and also an element set named ‘UEL’ is generated for all the elements included. In #2 section of input file, keyword line ‘*Uel property’ is used to specify the three user element material properties and the element set these properties applies to. The following data line simply specify the three material properties, when using in UEL, the first, second and third parameters can be obtained from PROPS(1), PROPS(2) and PROPS(3) respectively.

3.3 Post-processing for CS-FEM UEL Result

When both .inp and .f files (i.e. input file and user subroutine file) are generated, a job using the current input file and can be created in the ABAQUS GUI interface for analysis. It also can be achieved using command line [4]:

```
ABAQUS job=<input file name> user=UEL
```

When the analysis is completed, nodal displacements can be output to an .rpt file following the path: ABAQUS GUI>Report>Field Output>Unique Nodal>U: Spatial Displacement. Since ABAQUS does not post-process the information generated by user elements [7] and also not able
to recognize geometry, ABAQUS cannot plot the deformed shape after using CS-FEM UEL. To plot the deformed shape after running UEL subroutine, there are several suggestions: (1) If necessary, another input file of the exact model and mesh using FEM-Q4 element can be generated; apply nodal displacement results from UEL to the FEM model as initial boundary conditions; run this input file as a new job so ABAQUS is able to recognize geometry; plot the displacement result from the new job which should be the same as the result of UEL. However, because the strain and stress calculation procedure are different between FEM and CS-FEM, the result only shows the correct displacement of CS-FEM, none of the strain and stress results are CS-FEM results. (2) Overlay elements [7]. (3) Any other post-processing software that might be able to process the results.

After obtaining the displacement result from .rpt file, one can find whatever that is requested to output from UEL using ‘write’ command can be found in .log file. Also, it is possible to appoint the output to a certain file under certain path. In this case, the final result of array ENSNA is obtained. However, to obtain the exact smoothed strain/stress results on each node, post-processing procedure is needed. In this article, a MATLAB program is constructed according to Eq. 2.4 to recover the nodal smoothed stress. Appendix C is a sample of the MATLAB program.
4 NUMERICAL EXAMPLES

In this section, three cases were studied varies from plane stress to plane strain examples to show the accuracy of the proposed CS-FEM (4SD) UEL subroutine. For each numerical example, three types of mesh are used: regular mesh, slightly distorted mesh and heavily distorted mesh. For the regular mesh cases, the UEL results from every example were compared with analytical solutions, ABAQUS Q4 (4GP) results and reference FEM solutions if applicable. For the slightly distorted mesh cases, the UEL results were compared with analytical solutions and ABAQUS Q4 (4GP) results only since reference FEM solutions are only applicable for a couple of numerical examples using regular mesh. Also, to show CS-FEM’s resistance against heavily distorted mesh, each example was added in an irregular mesh which was unable to be calculated by ABAQUS due to sever distortion. Therefore, the results from irregular mesh will be only compared with analytical solution.

4.1 Cantilever Beam (Plane Stress Example)

4.1.1 Problem description
A rectangular cantilever beam as shown in Fig. 4.1.1 is studied. The beam has the length of \( L=1.2m \), width of \( D=0.3m \). Young’s Modules of the beam is \( E=3.0 \times 10^7 \text{N/m}^2 \), Poisson’s Ratio is \( \gamma=0.3 \). The beam is subjected to a parabolic traction at the free end, \( P=1000\text{N} \). Unit thickness is assumed for this beam, so plane stress condition is valid.
The analytical solution can be found in Ref. [8]:

\[ u_x = \frac{Py}{6EI} \left[ (6L - 3x)x + (2 + y) \left( y^2 - \frac{D^2}{4} \right) \right] \]  
(Eq. 4.1.1)

\[ u_y = -\frac{P}{6EI} \left[ 3y^2(L - x) + (4 + 5y) \frac{D^2x}{4} + (3L - x)x^2 \right] \]  
(Eq. 4.1.2)

and \( I \), which is the moment of inertia for the beam, is given by \( I = \frac{D^3}{12} \). And the stresses components corresponding to Eq. 4.1.1-4.1.2 are [1]:

\[ \sigma_{xx}(x, y) = \frac{P(\ell-x)y}{I} \]  
(Eq. 4.1.3)

\[ \sigma_{yy}(x, y) = 0, \]  
(Eq. 4.1.4)

\[ \tau_{xy}(x, y) = -\frac{P}{2I} \left( \frac{D^2}{4} - y^2 \right) \]  
(Eq. 4.1.5)

As discussed earlier in this section, three types of mesh have been created for this instance. For regular mesh, as shown in Fig. 4.1.2 (a), the entire problem domain was discretized. In the x-direction, the beam has been evenly divided into 48 elements; in the y-direction, the beam has been evenly divided into 12 elements. For slightly distorted mesh, all the nodes inside the problem domain have been randomly moved in a rather small range while all the nodes on the
edges were untouched as shown in Fig. 4.1.2 (b). For heavily distorted mesh, all the nodes inside the problem domain have been moved randomly in a larger range intentionally while all the nodes on the edges were untouched as shown in Fig. 4.1.2 (c).

For every mesh type, both displacement and stress results of the selected sections are compared. For displacement result comparison, y-direction displacement \(U_2\) of nodes on the beam axis \((y=0 \text{ (m)})\) were analyzed. For stress results, normal stress \(\sigma_{xx} (N/m^2)\) and shear stress \(\sigma_{xy} (N/m^2)\) at line \(x=0.6 \text{ (m)}\) were analyzed. Note that for slightly and heavily distorted meshes, the same node sets are used to compare the displacement and stress results even the coordinates of the nodes have been changed randomly. Correspondingly, the analytical solutions used to compare the results with were also recalculated according to the new location of every node in order to guarantee the correct analytical solution at each node. Since the result plot from different numerical methods overlap each other heavily, percentile error was calculated to show the trend and difference between different numerical methods.

4.1.2 Result analysis for regular mesh

For the regular mesh case, results from ABAQUS Q4 element with 4 Gauss Points (GP=4), FEM Q4 element with 4 Gauss Points (GP=4) and CS-FEM element with 4 smoothing domain (SD=4) were used to compare with analytical result. As shown in Fig. 4.1.3 (a), displacement results from all the numerical methods, including CS-FEM, agree perfectly with analytical solution.
Figure 4.1.2 Domain discretization of the cantilever using 4-noded quadrilateral elements:

(a) Regular mesh; (b) Slightly distorted mesh; (c) Heavily distorted mesh.
However, it can be obviously observed from Fig. 4.1.3 (b) that CS-FEM produces more accurate displacement result than both ABAQUS and FEM. As shown in Fig. 4.1.4 (a) and Fig. 4.1.5 (a), stress results from all the numerical methods, including CS-FEM, agree perfectly with analytical solution. However, it can be obviously observed from Fig. 4.1.3 (b) that CS-FEM produces more accurate stress result than both ABAQUS and FEM.

Note that the theory of S-FEM proves only that the S-FEM is softer than FEM [1]. Since FEM is known as always stiffer than the exact model [1] the CS-FEM model based on Q4 element is expected more accurate in terms of strain energy solution. This is because the mathematic prove must be performed in a norm (which is equivalent to the strain energy for solid mechanics problems). However, a method is proven more accurate in a norm does not necessary guarantee better accuracy in other measures, such as the displacement, strains (or stresses). For this particular case, CS-FEM out performs FEM, but for other cases, this may not be true. Therefore, the best comparison should be performed based on strain energy solution, but it is quite difficult to perform, for the limit time constrain for the master candidature. In the other examples, we will perform the same comparison, for the purpose of validating the code but not necessary arguing about the accuracy.
Figure 4.1.3 (a) Displacement results of $U_2$ (m) at $y=0$ (m);

(b) Error (%) of $U_2$ (m) at $y=0$ (m).
Figure 4.1.4 (a) Normal stress $\sigma_{xx} \ (N/m^2)$ results at $x=0.6 \ (m)$; 

(b) Error (%) at $x=0.6 \ (m)$. 
Figure 4.1.5 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) results at $x=0.6$ (m);
(b) Error (%) at $x=0.6$ (m).
4.1.3 *Result analysis for slightly distorted mesh*

Both displacement and stress results for the same node sets, \((y=0)\) and \((x=0.6)\), have been chosen to compare with analytical solutions. Due to the change of node coordinates, the analytical solution plot was recalculated according to the new coordinates of nodes. In the plots, the new sets of nodes are still referred as beam axis \((y=0)\) and line \((x=0.6)\) for convenience. For the slightly distorted mesh, there are no reference FEM results available. Thus, no FEM result is being compared within the following plots.

It is shown in Fig. 4.1.6 (a) that displacement results from both CS-FEM UEL and ABAQUS Q4 agree well to analytical solution. However, it can be clearly observed from Fig. 4.1.6 (b) that CS-FEM produces a more accurate displacement than ABAQUS in this case. For stress results, as shown in Fig. 4.1.7 (a) and Fig. 4.1.8 (a), both stress results from CS-FEM and ABAQUS are satisfactory comparing to analytical solution. However, when comparing the percentile error rate, it is shown that CS-FEM does not necessarily provide a more accurate solution than ABAQUS. It is because, for CS-FEM elements, post-processing procedures are still being researched and perfected, while ABAQUS is a very well-known and well developed commercial code that has a more accurate post-processing technique. In the future, with the perfection of CS-FEM post-processing methods, we are confident that we can produce more accurate stress results for CS-FEM element type. Nevertheless, the error rates of most CS-FEM stress results are within reasonable range. The two jump points on Fig. 4.1.7 (b) exist because the analytical solutions on those points are extremely small.
Figure 4.1.6 (a) Displacement results of U2 (m) at y=0 (m);
(b) Error (%) of U2 (m) at y=0 (m).
Figure 4.1.7 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at x=0.6 (m);
(b) Error (%) at x=0.6 (m).
Figure 4.1.8 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) results at $x=0.6$ (m);
(b) Error (%) at $x=0.6$ (m).
4.1.4 Result analysis for heavily distorted mesh

Both displacement and stress results for the same node sets, (y=0) and (x=0.6), have been chosen to compare with analytical solutions. Due to the change of node coordinates, the analytical solution plot was recalculated according to the new coordinates of nodes. In the plots, the new sets of nodes are still referred as beam axis (y=0) and line (x=0.6) for convenience. For reader’s information, there are no ABAQUS results being compared for irregular mesh in this section is because the mesh is too heavily distorted for ABAQUS to perform the computation. Therefore, there is no result produced by ABAQUS for this case.

It is clearly shown in Fig. 4.1.9 that CS-FEM is able to give out accurate displacement results when the mesh is heavily distorted. Because all the nodes inside the problem domain have been intentionally moved randomly, the data points are not as evenly distributed along analytical solution plot as they are in Fig. 4.1.3. The stress results are as shown in Fig. 4.1.10 and Fig. 4.1.11. It is also clear that the normal stress results also agree well with analytical solution under irregular mesh situation. These results clearly demonstrated CS-FEM element’s ability against mesh distortion. It is noticeable that shear stress results at \( x = 0.6 \) (m) is not as perfect as it is in regular mesh case. However, more details regarding post-processing approaches are still being researched for all S-FEM models, we are confident that it is possible to give out more accurate stress results for irregular mesh cases in the near future.
Figure 4.1.9 Displacement results $U_2$ (m) at $y=0$ (m).

Figure 4.1.10 Normal stress $\sigma_{xx}$ (N/m$^2$) results at $x=0.6$ (m).
4.2 Infinite Plate with Hole (Plane Strain Example)

4.2.1 Problem description
An infinite plate with a circular hole is studied in this case. As shown in Fig. 4.2.1, the infinite plate with hole is subjected to unidirectional tension \( \sigma = 1 \text{N/m} \) at infinity at the x-axis direction. The radius of the circular hole is \( a = 1 \text{m} \). Plane strain condition is considered in this case. The material property of the plate is described as \( E=1.0 \times 10^3 \text{N/m}^2 \) and \( \nu=0.3 \). Due to symmetry condition, only the upper right quadrant (shown as Fig. 4.2.2) of the plate is. Symmetric boundary conditions are applied on the left and bottom boundaries. Exact displacement solutions obtained from Eq.4.2.1 - Eq. 4.2.3 are applied at the top and right boundaries of the model.
The exact stresses solutions for this example can be found in [10]

\[ \sigma_{11} = 1 - \frac{a^2}{r^2} \left( \frac{3}{2} \cos 2\theta + \cos 4\theta \right) + \frac{3a^4}{2r^4} \cos 4\theta \]  
(Eq. 4.2.1)

\[ \sigma_{22} = -\frac{a^2}{r^2} \left( \frac{1}{2} \cos 2\theta - \cos 4\theta \right) - \frac{3a^4}{2r^4} \cos 4\theta \]  
(Eq. 4.2.2)

\[ \tau_{12} = -\frac{a^2}{r^2} \left( \frac{1}{2} \sin 2\theta + \sin 4\theta \right) + \frac{3a^4}{2r^4} \sin 4\theta \]  
(Eq. 4.2.3)

And the displacement components corresponding to the stresses are

\[ u_1 = \frac{a}{8\mu} \left[ \frac{r}{a} (k + 1) \cos \theta + \frac{2a}{r} \left( (1 + k) \cos \theta + \cos 3\theta \right) - \frac{2a^3}{r^3} \cos 3\theta \right] \]  
(Eq. 4.2.4)

\[ u_2 = \frac{a}{8\mu} \left[ \frac{r}{a} (k - 3) \sin \theta + \frac{2a}{r} \left( (1 - k) \sin \theta + \sin 3\theta \right) - \frac{2a^3}{r^3} \sin 3\theta \right] \]  
(Eq. 4.2.5)

where \( \mu = E/(2(1 + \nu)) \) and \( k = 3 - 4\nu \).

![Figure 4.2.1 Infinite plate with a circular hole subjected to x direction tension.](image-url)
Figure 4.2.2 Quarter model with size of 5m×5m with symmetric boundary conditions at left and bottom edges.
Figure 4.2.3 Domain discretization of the quarter model:

(a) Regular mesh; (b) Slightly distorted mesh; (c) Heavily distorted mesh.
As usual, three types of mesh for this problem have been created. For the regular mesh, the problem domain is further discretized into quadrilateral elements as shown in Fig. 4.2.3 (a). For slightly distorted mesh and heavily distorted mesh, all the nodes inside the problem domain have been randomly moved in on purpose while all the nodes on the edges were untouched as shown in Fig. 4.2.3 (b) and Fig. 4.2.4 (c).

4.2.2 Result analysis for regular mesh
As can be obviously observed in Fig. 4.2.4 (a) and Fig. 4.2.5 (a), the displacement results on the bottom and left boundary from different numerical methods agree perfectly well with analytical solution. However, Fig. 4.2.4 (b) and Fig. 4.2.5 (b) show that CS-FEM produce a more accurate displacement result than FEM but not necessarily have a more accurate result than ABAQUS. And Fig. 4.2.6 (a) and Fig. 4.2.7 (a) shows respectively that the normal stress results perpendicular to the bottom and left boundaries from different numerical methods agree perfectly well with analytical solution as well. However, shown by Fig. 4.2.6 (b) and Fig. 4.2.7 (b), CS-FEM does not necessarily produce a more accurate result than ABAQUS while providing a more accurate solution than FEM. The post-processing ABAQUS uses can generally give out better stress solutions, however, judging from the jump point in Fig. 4.2.7 (b), too much post-processing procedures might lead to extreme bad result for unique nodes.

It is clearly shown in Fig. 4.2.4 – Fig. 4.2.7 that, generally speaking, ABAQUS Q4 element gives out a better result than CS-FEM element while using regular mesh. However, most of the error rate remains less than 1%, which still proves that CS-FEM elements have a very decent accuracy regardless.
Figure 4.2.4 (a) Displacement $U_1(m)$ results at bottom boundary, $y = 0(m)$; 
(b) Error (%) at bottom boundary, $y = 0(m)$. 
Figure 4.2.5 (a) Displacement $U_2(m)$ results at left boundary, $x = 0(m)$;

(b) Error at left boundary, $x = 0(m)$. 
Figure 4.2.6 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) results at bottom boundary, y=0 (m);
(b) Error at bottom boundary, y=0 (m).
Figure 4.2.7 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at left boundary, $x=0$ (m);
(b) Error at left boundary, $x=0$ (m).
4.2.3 Result analysis for slightly distorted mesh

Same as the regular mesh type both displacement and stress results for the left and bottom boundaries have been chosen to compare with analytical solutions. Since the boundary points are unchanged during the process to make distorted mesh, the same analytical solutions were used in this case. For the slightly distorted mesh, there are no reference FEM results available. Thus, only results from CS-FEM UEL and ABAQUS Q4 element are being compared with analytical solution in the following plots.

For displacements, as can be obviously observed in Fig. 4.2.8 (a) and Fig. 4.2.9 (a), the results on the bottom and left boundary from both CS-FEM and ABAQUS agree perfectly well with analytical solution. However, Fig. 4.2.8 (b) and Fig. 4.2.9 (b) show that CS-FEM does not necessarily produce a more accurate result than ABAQUS. In this case, the result error points are more scattered than regular mesh, which implies the influence from distorted mesh. For stress results, Fig. 4.2.10 (a) and Fig. 4.2.11 (a) shows respectively that the normal stress results perpendicular to the bottom and left boundaries from both CS-FEM and ABAQUS agree perfectly well with analytical solution as well. However, shown by Fig. 4.2.10 (b) and Fig. 4.2.11 (b), CS-FEM does not necessarily produce a more accurate result than ABAQUS. And it is difficult to tell which of the two numerical methods is better than the other since most of the points are scattered and mixed together. Nonetheless, most of the error rate remains less than 1%, which proves that CS-FEM can provide accurate result under distorted mesh.
Figure 4.2.8 (a) Displacement U1(m) results at bottom boundary, y = 0(m);
(b) Error (%) at bottom boundary, y = 0(m).
Figure 4.2.9 Displacement U2(m) results at left boundary, x = 0(m);

(b) Error at left boundary, x = 0(m).
Figure 4.2.10 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) results at bottom boundary, $y=0$ (m);
(b) Error at bottom boundary, $y=0$ (m).
Figure 4.2.11 (a) Normal stress $\sigma_{xx}$ (N/m$^2$) results at left boundary, x=0 (m);

(b) Error at left boundary, x=0 (m).
4.2.4 Result analysis for heavily distorted mesh

The displacement and stress results calculated by CS-FEM UEL with irregular mesh of the same problem are plotted in Fig. 4.2.12 – Fig. 4.2.15. It can be clearly observed that CS-FEM model can give satisfactory results on both displacement and stress when a highly distorted mesh as shown in Fig. 4.2.3 (c) is used. Yet again, there are no ABAQUS results being compared for irregular mesh is because the mesh is too heavily distorted, the ABAQUS failed to perform the computation and could not produce any results. These results clearly demonstrated CS-FEM element’s ability against mesh distortion.

![Figure 4.2.12 Displacement U1(m) results at bottom boundary, y = 0(m).](image)

Figure 4.2.12 Displacement U1(m) results at bottom boundary, y = 0(m).
Figure 4.2.13 Displacement U2(m) results at left boundary, x = 0(m).

Figure 4.2.14 Normal stress $\sigma_{yy}$ (N/m$^2$) results at bottom boundary, y=0 (m).
Figure 4.2.15 Normal stress $\sigma_{xx}$ (N/m$^2$) results at left boundary, x=0 (m).

4.3 Semi-infinite Plane (Plane Strain Example)

4.3.1 Problem description

In this case, a semi-infinite plane subjected to a finite pressure on the top boundary is being studied. As shown in Fig. 4.3.1, x-axis is the infinite top boundary of the semi-infinite plane, the plane exist from x-axis and extend to the infinite of negative direction of y-axis. The semi-infinite plane is subjected to a uniform pressure within the range of $(-a \leq x \leq a)$. The uniform pressure on the top boundary is given by $p = 1MPa$. Rang of the limited area which is subjected to the pressure is specified by $a = 0.2(m)$. Plane strain condition is considered for this problem.
Figure 4.3.1 Semi-infinite plane subjected to a limited range uniform pressure.
Figure 4.3.2 Domain discretize of the semi-finite plane problem domain:

(a) Regular mesh; (b) Slightly distorted mesh; (c) Heavily distorted mesh.
The exact solutions for stresses are given by [10]:

\[
\sigma_{11} = \frac{p}{2\pi} [2(\theta_1 - \theta_2) - \sin 2\theta_1 + \sin 2\theta_2] \quad (Eq. 4.3.1)
\]

\[
\sigma_{22} = \frac{p}{2\pi} [2(\theta_1 - \theta_2) + \sin 2\theta_1 - \sin 2\theta_2] \quad (Eq. 4.3.2)
\]

\[
\tau_{12} = \frac{p}{2\pi} (\cos 2\theta_1 - \cos 2\theta_2) \quad (Eq. 4.3.3)
\]

The corresponding displacements can be expressed as [10]:

\[
u_1 = \frac{p(1 - v^2)}{\pi E} \left[ \frac{1 - 2v}{1 - v} \left( (x + a)\theta_1 - (x - a)\theta_2 \right) + 2y \ln \frac{r_1}{r_2} \right] \quad (Eq. 4.3.4)
\]

\[
u_2 = \frac{p(1 - v^2)}{\pi E} \left[ \frac{1 - 2v}{1 - v} \left( y(\theta_1 - \theta_2) + 2H \tan^{-1} \frac{1}{c} \right) + 2(x - a) \ln r_2 \right] - 2(x + a) \ln r_1 + 4a \ln a + 2a \ln(1 + c^2) \quad (Eq. 4.3.5)
\]

The directions of \(\theta_1\) and \(\theta_2\) are suggested in Fig. 4.3.1. The length of \(r_1\) and \(r_2\) are also as demonstrated in Fig. 4.3.1. The material’s Young’s Modulus is \(E = 1 \times 10^5 N/m^2\), Poisson’s Ratio \(v = 0.3\). \(c\) is a coefficient with the value of 100. \(H = ca\) is defined as the distance from the origin \(O\) to point \(O'\) at which vertical displacement is assumed to be zero. Due to the symmetric condition, only the right half plane with the length of \(5a \times 5a\) is modeled to solve the problem. Exact displacement result of the left and bottom boundaries are obtained from Eq. 4.3.4 - Eq. 4.3.5 and are used as displacement boundary conditions for the two edges. Right boundary is subjected to tractions obtained from Eq. 4.3.1 – Eq. 4.3.3.

As usual, three types of mesh were created for this problem, shown in Fig. 4.3.2. For regular mesh, the problem domain was evenly discretized into 1600 elements with 40 elements on each edge as shown in Fig. 4.3.2 (a). For slightly distorted and heavily distorted mesh, the problem
domain was first divided into 400 elements with 20 elements on each edge, and then the coordinates of nodes within the problem domain were randomly modified to create the distorted mesh while the edge nodes were kept at the same locations. This is to see if CS-FEM can also give out satisfactory result with less element number when the mesh is distorted.

4.3.2 Result analysis for regular mesh
In this example, no FEM results are available to compare. Thus only CSFEM and ABAQUS results are presented. The x and y displacement results at the free edge \((y = 0)\) are compared with analytical solution in Fig. 4.3.3 and Fig. 4.3.4, along with the stress components on the diagonal line \((y = -x)\) of the model, Fig. 4.3.5 – Fig. 4.3.7. It is obvious that the results from both CS-FEM and ABAQUS have extremely good agreement with analytical solution.

To determine which of CS-FEM and ABAQUS has a better accuracy than the other, the percentile errors are once again calculated for both numerical methods. The percentile errors of both numerical methods are pretty close to each other. It is difficult to say if one is more accurate than the other while using regular mesh. And for both CS-FEM and ABAQUS numerical results, the most inaccurate result for displacement occurs at the edge of top surface limited pressure \(x = 0.2(m)\) which is a stress singular point. However, it is worth mentioning that most of the error rates of CS-FEM results are below 1%, which still ascertains the accuracy of CS-FEM elements under regular mesh. Therefore, the results of CS-FEM with regular mesh are just as good as ABAQUS and have a satisfactory agreement with analytical solution.
Figure 4.3.3 (a) Displacement $U_1(m)$ results on the top free boundary, $y=0(m)$;

(b) Error (%) on the top free boundary, $y=0(m)$. 
Figure 4.3.4 (a) Displacement $U_2(m)$ results on the top free boundary, $y=0(m)$;

(b) Error (%) on the top free boundary, $y=0(m)$. 
Figure 4.3.5 (a) 4.3.5 Normal stress $\sigma_{xx}$ (N/m$^2$) along diagonal line (y=-x);

(b) Error (%) along diagonal line (y=-x).
Figure 4.3.6 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) along diagonal line (y=x);
(b) Error (%) along diagonal line (y=x).
Figure 4.3.7 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) along diagonal line ($y=-x$); (b) Error (%) along diagonal line ($y=-x$).
4.3.3 Result analysis for slightly distorted mesh

In this mesh case, no FEM results are available to compare. Thus only CSFEM and ABAQUS results are presented. The x and y displacement results at the free edge \((y = 0)\) are compared with analytical solution in Fig. 4.3.8 and Fig. 4.3.9, along with the stress components on the diagonal line \((y = -x)\) of the model, Fig. 4.3.10 – Fig. 4.3.12.

For displacement results showed in Fig. 4.3.8 (a) and Fig. 4.3.9 (a), both CS-FEM and ABAQUS results agree well with analytical solution, hence, percentile error rates were presented to do further analysis. From Fig. 4.3.8 (b) and Fig. 4.3.9 (b), it is clear that those results from CS-FEM and ABAQUS are very close to each other thus it’s difficult to make a judgment about which numerical method has a better result. However, for node \((x=0.2 \text{ (m)})\) and nodes close to it on the top boundary, both CS-FEM and ABAQUS gives out poor accuracy result. The reason for this inaccuracy is that point \((0.2, 0)\) is the stress singular point.

For stress results compared in Fig. 4.3.10 - Fig. 4.3.12, it is clear that stress results from both CS-FEM and ABAQUS agree well with analytical solution in general. And when percentile error rates were calculated to do further study, CS-FEM and ABAQSU results are scattered around and mixed with each other. Once again, it is impossible to tell which of the two numerical methods provides better stress results.
Figure 4.3.8 (a) Displacement U1(m) results on the top free boundary, y=0(m);

(b) Error (%) on the top free boundary, y=0(m).
Figure 4.3.9 (a) Displacement U2(m) results on the top free boundary, y=0(m);

(b) Error (%) on the top free boundary, y=0(m).
Figure 4.3.10 (a) 4.3.10 Normal stress $\sigma_{xx}$ (N/m$^2$) along diagonal line (y=-x);

(b) Error (%) along diagonal line (y=-x).
Figure 4.3.11 (a) Normal stress $\sigma_{yy}$ (N/m$^2$) along diagonal line (y=-x);

(b) Error (%) along diagonal line (y=-x).
Figure 4.3.12 (a) Shear stress $\sigma_{xy}$ (N/m$^2$) along diagonal line ($y=\pm x$);

(b) Error (%) along diagonal line ($y=\pm x$).
4.3.4 Result analysis for heavily distorted mesh

The same sets of nodes from were chosen to compare the results between CS-FEM and analytical solution. Once again, due to the change of node coordinates, analytical solutions were recalculated according to the new location of nodes. Therefore, the analytical solution plot in Fig. 4.3.15 – Fig. 4.3.17 are not as smooth as those with regular mesh, and the CS-FEM results are not as evenly distributed as they were before. However, as shown in Fig. 4.3.13 – Fig 4.3.17, CS-FEM UEL is still able to give out satisfactory displacement results that have a perfect agreement with analytical solution. Although the results of stress components at chosen points are not as perfect as displacement results, they are still acceptable and close to analytical solution. As discussed earlier in this article, the irregular mesh that was chosen cannot be calculated by ABAQUS. Therefore, no result from ABAQUS is being compared in this section.

![Displacement U1(m) results on the top free boundary, y=0(m).](image)

**Figure 4.3.13 Displacement U1(m) results on the top free boundary, y=0(m).**
Figure 4.3.14 Displacement U2(m) results on the top free boundary, y=0(m).

Figure 4.3.15 Normal stress $\sigma_{xx}$ (N/m$^2$) along diagonal line (y=-x).
Figure 4.3.16 Normal stress $\sigma_{yy}$ (N/m$^2$) along diagonal line (y=-x).

Figure 4.3.17 Normal stress $\sigma_{xy}$ (N/m$^2$) along diagonal line (y=-x).
5 SUMMARY

An implementation of 4SD CS-FEM in ABAQUS via User-defined Element subroutine was presented in this article. Three numerical examples (one plane stress condition, two plane strain condition) were used to verify the accuracy of the CS-FEM UEL subroutines. From the result provided at the end of each example, it is clear that CS-FEM element has a great advantage over FEM result. CS-FEM mostly provides more accurate results than ABAQUS Q4 element in plane stress situation, while giving out satisfactory results comparing with analytical solution in plane strain situation. However, CS-FEM UEL is able to calculate heavily distorted mesh which is impossible to calculate by ABAQUS Q4 element and still gives out reliable results that agree well with analytical solution. Also, more post-processing approaches for CS-FEM models are still under research. It is believed that with more optimized post-processing approaches, CS-FEM might be able to give out more accurate stress results under distorted meshes. To sum up, the implementation was successful. Both accuracy of CS-FEM element and its ability to resist distorted mesh were demonstrated perfectly.

For future works, post-processing procedure can be perfected in order to give better stress results. Also CS-FEM subroutines for nonlinear problems (both geometry and material) can be developed following the similar path that was presented in this article. However, if one were to try implementation any other type of S-FEM, more access to the ABAQUS source file is needed in order to complete smoothing domain construction among elements. If it were able to do implementation with other S-FEM models, more accurate results can be provided since generally all other S-FEM models are stronger than CS-FEM.
REFERENCES


Appendix A: CS-FEM UEL

SUBROUTINE UEL (RHS, AMATRX, SVARS, ENERGY, NDOFEL, NRHS, NSVARS, & PROPS, NPROPS, COORDS, MCRD, NNODE, U, DU, V, A, JTYPE, TIME, & DIIME, KSTE, KINC, JELEM, PARAMS, NDLOAD, JDLTYP, ADLMAG, & PREDEF, NPREDF, LFLAGS, MLVARX, DDLMAG, MDLOAD, PNEWDT, JPROPS, & NPRO, PERIOD)
  INCLUDE 'ABA_PARAM.INC'
  DIMENSION RHS(MLVARX,*), AMATRX(NDOFEL,NDOFEL), PROPS(3), & SVARS(*), ENERGY(8), COORDS(MCRD, NNODE), U(NDOFEL), & DU(MLVARX,*), V(NDOFEL), A(NDOFEL), TIME(2), PARAMS(*), & JDLTYP(MDLOAD,*), ADLMAG(MDLOAD,*), DDLMAG(MDLOAD,*), & PREDEF(2, NPREDF, NNODE), LFLAGS(*), JPROPS(*)

Important Variables

GAUSVAL: Value of shape functions at gauss points.
D: Constitutive matrix.
P1-P4: Element nodes.
P5-P9: Added nodes to form the smoothing domain.

P4-----P7-----P3
| SD4 | SD3 |
P8-----P9-----P6
| SD1 | SD2 |
P1-----P5-----P2

SSD: Array contains area of each smoothing domain.
AKMAT: Temporary storage matrix for Kij (i,j=1,8).
BI: Matrix contains strain-displacement matrix of every smoothing
domain at each node---Bij (i, the shape function used to
evaluate Bij; j, the smoothing domain # in which the strain-
displacement matrix is being evaluated); Bij is a 3x2 matrix.
For example, B23 is the strain-displacement matrix of
smoothing domain three (SD3) at node two (P2).
BIT: Matrix contains the transpose matrix of each Bij from BI matrix
---BijT. Note, it is not the transpose of BI matrix itself.
B1SD-B4SD: Strain-displacement matrix of smoothing domain 1-4.
E1SD-E4SD: Calculated strain of smoothing domain 1-4.
S1SD-S4SD: Calculated stress of smoothing domain 1-4.
C ENSNA: Matrix contains strain/stress and area results for each
C smoothing domain. This matrix is constructed for output purpose.
C THICK: Thickness of the planner.
C E: Young's modulus of the material.
C EMU: Poisson's ratio of the material.
C---------------------------------------------------------------------------
C
C---------- Define parameters, dimensions and element type
REAL(8) B(3,8), BT(8,3),
& PONVAL(9,4), GAUSVAL(12,4),D(3,3), SSTRAIN(3), SSTRESS(3),
& P1(2), P2(2), P3(2), P4(2), P5(2), P6(2), P7(2), P8(2), P9(2),
& SSD(4), AKMAT(2,2), BI(12,8), BIT(12,8), B1SD(3,8),
& B2SD(3,8), B3SD(3,8), B4SD(3,8), ENSNA(7,4),
& S1SD(3,1)
PARAMETER (NEG=-1.D0, ZERO=0.D0, HALF=0.5D0,
& ONE=1.D0, TWO=2.D0)
C---------- Initialize matrices
call get_matzero(AMATRIX,8,8)
call get_matzero(B,3,8)
call get_matzero(ENSNA,7,4)
call get_matzero(B1SD,3,8)
call get_matzero(B2SD,3,8)
call get_matzero(B3SD,3,8)
call get_matzero(B4SD,3,8)
C---------- Define coordinates for P1-P4
P1(1)=COORDS(1,1)
P1(2)=COORDS(2,1)
P2(1)=COORDS(1,2)
P2(2)=COORDS(2,2)
P3(1)=COORDS(1,3)
P3(2)=COORDS(2,3)
P4(1)=COORDS(1,4)
P4(2)=COORDS(2,4)
C---------- Calculate coordinates for location 5-9
P5(1)=(COORDS(1,1)+COORDS(1,2))/2
P6(1)=(COORDS(1,2)+COORDS(1,3))/2
P7(1)=(COORDS(1,3)+COORDS(1,4))/2
P8(1)=(COORDS(1,4)+COORDS(1,1))/2
P9(1)=(COORDS(1,1)+COORDS(1,2)+COORDS(1,3)+COORDS(1,4))/4
P5(2)=(COORDS(2,1)+COORDS(2,2))/2
P6(2)=(COORDS(2,2)+COORDS(2,3))/2
P7(2)=(COORDS(2,3)+COORDS(2,4))/2
P8(2)=(COORDS(2,4)+COORDS(2,1))/2
P9(2)=(COORDS(2,1)+COORDS(2,2)+COORDS(2,3)+COORDS(2,4))/4
C---------- Assign data to GAUSVAL matrix
data GAUSVAL /
& \quad 0.75D0, \; 0.375D0, \; 0.375D0, \; 0.75D0, \; 0.25D0, \; 0.0D0, \\
& \quad 0.125D0, \; 0.0D0, \; 0.0D0, \; 0.125D0, \; 0.0D0, \; 0.25D0, \\
& \quad 0.25D0, \; 0.375D0, \; 0.125D0, \; 0.0D0, \; 0.75D0, \; 0.75D0, \\
& \quad 0.375D0, \; 0.25D0, \; 0.0D0, \; 0.125D0, \; 0.0D0, \; 0.0D0, \\
& \quad 0.0D0, \; 0.125D0, \; 0.125D0, \; 0.0D0, \; 0.0D0, \; 0.25D0, \\
& \quad 0.375D0, \; 0.75D0, \; 0.75D0, \; 0.375D0, \; 0.25D0, \; 0.0D0, \\
& \quad 0.0D0, \; 0.125D0, \; 0.375D0, \; 0.25D0, \; 0.0D0, \; 0.0D0, \\
& \quad 0.125D0, \; 0.0D0, \; 0.25D0, \; 0.375D0, \; 0.75D0, \; 0.75D0/

C------------- Calculate Bij, BijT and SSD
C------------- B11,B11T---B12,B12T---B13,B13T---B14,B14T
          call get_bnb(p1,p5,p9,p8,GAUSVAL(1,1),GAUSVAL(2,1),GAUSVAL(3,1),
            & GAUSVAL(4,1),BI(1:3,1:2),BIT(1:2,1:3),SSD(1))
          call get_bnb(p5,p2,p6,p9,GAUSVAL(5,1),GAUSVAL(6,1),GAUSVAL(7,1),
            & GAUSVAL(2,1),BI(1:3,3:4),BIT(1:2,4:6),SSD(2))
          call get_bnb(p9,p6,p3,p7,GAUSVAL(7,1),GAUSVAL(8,1),GAUSVAL(9,1),
            & GAUSVAL(10,1),BI(1:3,5:6),BIT(1:2,7:9),SSD(3))
          call get_bnb(p8,p9,p7,p4,GAUSVAL(3,1),GAUSVAL(10,1),
            & GAUSVAL(11,1), GAUSVAL(12,1),BI(1:3,7:8),BIT(1:2,10:12),SSD(4))

C------------- B21,B21T---B22,B22T---B23,B23T---B24,B24T
          call get_bnb(p1,p5,p9,p8,GAUSVAL(1,2),GAUSVAL(2,2),GAUSVAL(3,2),
            & GAUSVAL(4,2),BI(4:6,1:2),BIT(3:4,1:3),SSD(1))
          call get_bnb(p5,p2,p6,p9,GAUSVAL(5,2),GAUSVAL(6,2),GAUSVAL(7,2),
            & GAUSVAL(2,2),BI(4:6,3:4),BIT(3:4,4:6),SSD(2))
          call get_bnb(p9,p6,p3,p7,GAUSVAL(7,2),GAUSVAL(8,2),GAUSVAL(9,2),
            & GAUSVAL(10,2),BI(4:6,5:6),BIT(3:4,7:9),SSD(3))
          call get_bnb(p8,p9,p7,p4,GAUSVAL(3,2),GAUSVAL(10,2),
            & GAUSVAL(11,2), GAUSVAL(12,2),BI(4:6,7:8),BIT(3:4,10:12),SSD(4))

C------------- B31,B31T---B32,B32T---B33,B33T---B34,B34T
          call get_bnb(p1,p5,p9,p8,GAUSVAL(1,3),GAUSVAL(2,3),GAUSVAL(3,3),
            & GAUSVAL(4,3),BI(7:9,1:2),BIT(5:6,1:3),SSD(1))
          call get_bnb(p5,p2,p6,p9,GAUSVAL(5,3),GAUSVAL(6,3),GAUSVAL(7,3),
            & GAUSVAL(2,3),BI(7:9,3:4),BIT(5:6,4:6),SSD(2))
          call get_bnb(p9,p6,p3,p7,GAUSVAL(7,3),GAUSVAL(8,3),GAUSVAL(9,3),
            & GAUSVAL(10,3),BI(7:9,5:6),BIT(5:6,7:9),SSD(3))
          call get_bnb(p8,p9,p7,p4,GAUSVAL(3,3),GAUSVAL(10,3),
            & GAUSVAL(11,3), GAUSVAL(12,3),BI(7:9,7:8),BIT(5:6,10:12),SSD(4))

C------------- B41,B41T---B42,B42T---B43,B43T---B44,B44T
          call get_bnb(p1,p5,p9,p8,GAUSVAL(1,4),GAUSVAL(2,4),GAUSVAL(3,4),
            & GAUSVAL(4,4),BI(10:12,1:2),BIT(7:8,1:3),SSD(1))
          call get_bnb(p5,p2,p6,p9,GAUSVAL(5,4),GAUSVAL(6,4),GAUSVAL(7,4),
            & GAUSVAL(2,4),BI(10:12,3:4),BIT(7:8,4:6),SSD(2))
          call get_bnb(p9,p6,p3,p7,GAUSVAL(7,4),GAUSVAL(8,4),GAUSVAL(9,4),
            & GAUSVAL(10,4),BI(10:12,5:6),BIT(7:8,7:9),SSD(3))
          call get_bnb(p8,p9,p7,p4,GAUSVAL(3,4),GAUSVAL(10,4),
            & GAUSVAL(11,4), GAUSVAL(12,4),BI(10:12,7:8),BIT(7:8,10:12),SSD(4))
C------------- Assemble Constitutive matrix---D matrix
C------------- Pass material properties into variables
  THICK=PROPS(1)
  E=PROPS(2)
  EMU=PROPS(3)

C------------- Plane Strain condition
  D(1,1)=E*(1.0-EMU)/((1.0+EMU)*(1.0-2*EMU))
  D(1,2)=E*EMU/((1.0+EMU)*(1.0-2*EMU))
  D(1,3)=0.0
  D(2,1)=E*EMU/((1.0+EMU)*(1.0-2*EMU))
  D(2,2)=E*(1.0-EMU)/((1.0+EMU)*(1.0-2*EMU))
  D(2,3)=0.0
  D(3,1)=0.0
  D(3,2)=0.0
  D(3,3)=E*(1.0-2*EMU)/(2*(1.0+EMU)*(1.0-2*EMU))

C------------- Calculate Kij and assemble K matrix
  do i = 1, 4
    do j = 1, 4
      do m = 1, 4
        call get_kmat(BIT((2*i-1):2*i,(3*m-2):3*m),D,
                      &                    BI((3*j-2):3*j,(2*m-1):2*m),2,3,3,2,SSD(m),AKMAT)
        AMATRX((2*i-1):2*i,(2*j-1):2*j)=AMATRX((2*i-1):2*i,
                      &                    (2*j-1):2*j)+AKMAT
      enddo
    enddo
  enddo

C------------- Calculate B1SD, E1SD, S1SD and save the result for output
  do i = 1,4
  enddo

  do i = 1,3
    do j = 1,8
      ENSNA(i,1)=ENSNA(i,1)+B1SD(i,j)*U(j)
    enddo
  enddo

  do i = 1,3
    do j = 1,3
      ENSNA(i+3,1)=ENSNA(i+3,1)+D(i,j)*ENSNA(j,1)
    enddo
  enddo

C------------- Calculate B2SD, E2SD, S2SD and save the result for output
  do i = 1,4
  enddo
C---------- Calculate B3SD, E3SD, S3SD and save the result for output
  do  i = 1,4
    B3SD(1:3,(2*i-1):2*i)=B3SD(1:3,(2*i-1):2*i)+BI((3*i-2):3*i,5:6)
  enddo
  do  i = 1,3
    do  j = 1,8
      ENSNA(i+3,2)=ENSNAl(i+3,2)+D(i,j)*ENSNAl(j,2)
    enddo
  enddo
  do  i = 1,4
    B3SD(1:3,(2*i-1):2*i)=B3SD(1:3,(2*i-1):2*i)+BI((3*i-2):3*i,5:6)
  enddo
C---------- Calculate B4SD, E4SD, S4SD and save the result for output
  do  i = 1,4
  enddo
  do  i = 1,3
    do  j = 1,8
      ENSNA(i,4)=ENSNAl(i,4)+B4SD(i,j)*U(j)
    enddo
  enddo
  do  i = 1,4
  enddo
C---------- Save smoothing domain area into ENSNA for output
  do  i = 1,4
ENSNA(7,i)=SSD(i)
SVARS(i)=SSD(i)
enddo

C---------- Calculate the residual force vector
  do k1 = 1, NDOFEL
    do KRHS = 1, NRHS
      RHS(K1,KRHS) = ZERO
    enddo
  enddo
  do I=1,8
    do J=1,8
      RHS(I,1)=RHS(I,1)-AMATRX(I,J)*U(J)*THICK
    enddo
  enddo

C---------- Output the saved results
  write (*,*) JELEM
  do I=1,4
    do J=1,7
      write (*,*) ENSNA(J,I)
    enddo
  enddo
  return
end

C***********************************
C SUBROUTINES
C***********************************

C# Subroutine to calculate Bij and BijT
subroutine get_bnb(p1,p2,p3,p4,ng1,ng2,ng3,ng4,b,bt,area)
  INCLUDE 'ABA_PARAM.INC'
  real(8) ng1,ng2,ng3,ng4,x(4),y(4),l(4)
  real(8) bi1,bi2,p1(2),p2(2),p3(2),p4(2)
  real(8) area
  real(8) outnorm(2,4),b(3,2),bt(2,3)

C-------------------------------------
C Variables
C  bi1: bix.
C  bi2: biy.
C  p1-p4: Coordinates of the four nodes supporting the smoothing domain.
C  x: X coordinates of the four nodes supporting the smoothing domain.
C  y: Y coordinates of the four nodes supporting the smoothing domain.
C  ng1-4: Values of shape functions at four gauss points at each boundary
C  of the smoothing domain.
C  l: Length of four boundaries.
C  area: Area of the current smoothing domain.
C     outnorm: Matrix contains four outward unit normal for each boundary.
C     b: Final result of Bij matrix.
C     bt: Transpose of Bij matrix (i.e., BijT).

x(1)=p1(1)
y(1)=p1(2)
x(2)=p2(1)
y(2)=p2(2)
x(3)=p3(1)
y(3)=p3(2)
x(4)=p4(1)
y(4)=p4(2)

call get_quadarea(x,y,area,l)
call get_outnorm(x,y,l,outnorm)

bi1=(outnorm(1,1)*ng1*l(1) + outnorm(1,2)*ng2*l(2) +
     outnorm(1,3)*ng3*l(3) + outnorm(1,4)*ng4*l(4))/area
&
bi2=(outnorm(2,1)*ng1*l(1) + outnorm(2,2)*ng2*l(2) +
     outnorm(2,3)*ng3*l(3) + outnorm(2,4)*ng4*l(4))/area

call get_matzero(b,3,2)

b(1,1)=b(1,1)+bi1
b(2,2)=b(2,2)+bi2
b(3,1)=b(3,1)+bi2
b(3,2)=b(3,2)+bi1

call get_mattran(b,bt,3,2)

return
end

C############### Subroutine to calculate quadrilateral area

subroutine get_quadarea(x,y,area,edge)
INClude 'ABA_PARAM.INC'
real(8) x(4),y(4),area,angle(4),edge(4),area1,area2,dia(2)

call get_quad_angle(x,y,angle,edge,dia)

if (angle(1)>=pi .or. angle(3)>=pi ) then
  area1=0.5*(x(1)*y(2)+x(2)*y(3)+x(3)*y(1)-
  x(2)*y(1)-x(3)*y(2)-x(1)*y(3))
&
  area2=0.5*(x(1)*y(3)+x(3)*y(4)+x(4)*y(1)-
  x(1)*y(4)-x(3)*y(1)-x(4)*y(3))

  area=abs(area1)+abs(area2)
else if(angle(2)>=pi .or. angle(4)>=pi) then
  area1=0.5*(x(1)*y(2)+x(2)*y(4)+x(4)*y(1)-
  x(1)*y(4)-x(2)*y(1)-x(4)*y(2))
  area2=0.5*(x(2)*y(3)+x(3)*y(4)+x(4)*y(2)-
  x(2)*y(3)-x(3)*y(1)-x(4)*y(2))
else
  area1=0.5*(x(1)*y(2)+x(2)*y(3)+x(3)*y(4)+x(4)*y(1)-
  x(1)*y(4)-x(2)*y(3)-x(3)*y(2)-x(4)*y(1))
  area2=0.5*(x(1)*y(3)+x(2)*y(4)-x(3)*y(1)-x(4)*y(2))

  area=abs(area1)+abs(area2)
endif
& \quad x(2)y(4)-x(3)y(2)-x(4)y(3))
\text{area}=\text{abs}(\text{area1})+\text{abs}(\text{area2})
\text{else if } (\text{angle}(1)<\pi \text{ and } \text{angle}(2)<\pi \text{ and } \text{angle}(3)<\pi
& \quad \text{and } \text{angle}(4)<\pi \text{ and } \text{dia}(1)<\text{dia}(2)) \text{then}
\text{area1}=0.5*(x(1)y(2)+x(2)y(3)+x(3)y(1)-
& \quad x(1)y(3)-x(2)y(1)-x(3)y(2))
\text{area2}=0.5*(x(1)y(3)+x(3)y(4)+x(4)y(1)-
& \quad x(1)y(4)-x(3)y(1)-x(4)y(3))
\text{area}=\text{abs}(\text{area1})+\text{abs}(\text{area2})
\text{else if } (\text{angle}(1)<\pi \text{ and } \text{angle}(2)<\pi \text{ and } \text{angle}(3)<\pi
& \quad \text{and } \text{angle}(4)<\pi \text{ and } \text{dia}(1)>\text{dia}(2)) \text{then}
\text{area1}=0.5*(x(1)y(2)+x(2)y(4)+x(4)y(1)-
& \quad x(1)y(4)-x(2)y(1)-x(4)y(2))
\text{area2}=0.5*(x(2)y(3)+x(3)y(4)+x(4)y(2)-
& \quad x(2)y(4)-x(3)y(2)-x(4)y(3))
\text{area}=\text{abs}(\text{area1})+\text{abs}(\text{area2})
\text{end if}
\text{return}
\text{end}

C************************** Subroutine to calculate quadrilateral inner angle
\textbf{subroutine get_quad_angle(x,y,angle,edge,dia)}
\textbf{INCLUDE 'ABA_PARAM.INC'}
\textbf{real(8)} \textbf{x(4)},\textbf{y(4)},\textbf{angle(4)},\textbf{d1,d2,edge(4),dia(2)}
\textbf{real(8)} \textbf{coA1,coA2,coB1,coB2,coC1,coC2,coD1,coD2}
\textbf{call get_length(x(1),y(1),x(3),y(3),d1)}
\textbf{call get_length(x(2),y(2),x(4),y(4),d2)}
\textbf{do i=1,3}
\quad \textbf{call get_length(x(i),y(i),x(i+1),y(i+1),edge(i))}
\textbf{enddo}
\textbf{call get_length(x(4),y(4),x(1),y(1),edge(4))}
\textbf{coA1=(d1**2+edge(4)**2-edge(3)**2)/(2*d1*edge(4))}
\textbf{coA2=(d1**2+edge(1)**2-edge(2)**2)/(2*d1*edge(1))}
\textbf{coB1=(d2**2+edge(1)**2-edge(4)**2)/(2*d2*edge(1))}
\textbf{coB2=(d2**2+edge(2)**2-edge(3)**2)/(2*d2*edge(2))}
\textbf{coC1=(d1**2+edge(2)**2-edge(1)**2)/(2*d1*edge(2))}
\textbf{coC2=(d1**2+edge(3)**2-edge(4)**2)/(2*d1*edge(3))}
\textbf{coD1=(d2**2+edge(3)**2-edge(2)**2)/(2*d2*edge(3))}
\textbf{coD2=(d2**2+edge(4)**2-edge(1)**2)/(2*d2*edge(4))}
\textbf{angle(1)=acos(coA1)+acos(coA2)}
\textbf{angle(2)=acos(coB1)+acos(coB2)}
\textbf{angle(3)=acos(coC1)+acos(coC2)}
\textbf{angle(4)=acos(coD1)+acos(coD2)}
dia(1)=d1
dia(2)=d2

return
end

C############### Subroutine to calculate distance between two points
subroutine get_length(x1,y1,x2,y2,l)
  INCLUDE 'ABA_PARAM.INC'
  real(8) x1,y1,x2,y2,l
  l=sqrt((x1-x2)**2+(y1-y2)**2)
  return
end

C############### Subroutine to calculate outward unit normal
subroutine get_outnorm(x,y,length,norm)
  INCLUDE 'ABA_PARAM.INC'
  real(8) x(4),y(4),length(4),norm(2,4)
  do i=1,3
    norm(1,i)=0.5*(y(i+1)-y(i))/(length(i)/2)
    norm(2,i)=-0.5*(x(i+1)-x(i))/(length(i)/2)
  end do
  norm(1,4)=0.5*(y(1)-y(4))/(length(4)/2)
  norm(2,4)=-0.5*(x(1)-x(4))/(length(4)/2)
  return
end

C############### Subroutine to make zero matrix
subroutine get_matzero(A,n,m)
  INCLUDE 'ABA_PARAM.INC'
  real(8) A(n,m)
  do i = 1, n
    do j = 1, m
      A(i,j)=0.d0
    enddo
  enddo
  return
end

C############### Subroutine to multiply matrix
subroutine get_matmul(A,B,C,l,n,m)
  INCLUDE 'ABA_PARAM.INC'
  real(8) A(l,n),B(n,m),C(l,m)
  call get_matzero(C,l,m)
  do i = 1, l
    do j = 1, m
      do k = 1, n
\[
C(i,j) = c(i,j) + A(i,k) * B(k,j)
\]

```fortran
C(i,j) = c(i,j) + A(i,k) * B(k,j)
enddo
enddo
enddo
return
end

C+++++++++++++++++++++++ Subroutine to Calculate KIJ
subroutine get_kmat(A,B,C,k,l,m,n,area,D)
  INCLUDE 'ABA_PARAM.INC'
  real(8) A(k,l),B(l,m),C(m,n)
  real(8) D(k,n),E(k,m)
  real(8) area
  call get_matzero(D,k,n)
  call get_matzero(E,k,m)
  call get_matmul(A,B,E,k,l,m)
  call get_matmul(E,C,D,k,m,n)
  do i=1,k
    do j=1,n
      D(i,j) = D(i,j) * area
    enddo
  enddo
return
end

C+++++++++++++++++++++++ Subroutine to do matrix transpose
subroutine get_mattran(A,B,m,n)
  INCLUDE 'ABA_PARAM.INC'
  real(8) A(m,n),B(n,m)
  call get_matzero(B,n,m)
  do i=1,n
    do j=1,m
      B(i,j) = A(j,i)
    enddo
  enddo
return
end
```
Appendix B  Sample of Input File

*Heading
** Job name: New_Beam Model name: New Beam
** Generated by: ABAQUS/CAE 6.11-2
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
*Part, name=Part-1
*Node
   1,      0., -0.150000006
   2,  0.0250000004, -0.150000006
   3,  0.0500000007, -0.150000006
   .
   .
   635,   1.14999998,  0.150000006
   636,   1.17499995,  0.150000006
   637,   1.20000005,  0.150000006
****************************************************************
** #1 User-defined Element Definition
****************************************************************
*User element, nodes=4, type=U104, properties=3, coordinates=2, variables=7
  1, 2
*Element, type=U104, elset=UEL
  1,   1,   2,  51,  50
  2,  2,   3,  52,  51
  3,  3,   4,  53,  52
  .
  .
  574, 585, 586, 635, 634
  575, 586, 587, 636, 635
  576, 587, 588, 637, 636
****************************************************************
** #2 User Element Material Property Definition
****************************************************************
*Uel property, elset=UEL
  1.0 , 3.0e7 , 0.3
*Nset, nset=_PickedSet2, internal, generate
  1,  637,    1
*Elset, elset=_PickedSet2, internal, generate
  1,  576,    1
** Section: Section-1
*Solid Section, elset=_PickedSet2, material=Material-1
  1.,
*End Part
**
** ASSEMBLY
**Assembly, name=Assembly**

**Instance, name=Part-1-1, part=Part-1**

**End Instance**

#3 Node Sets Generation for Loading

**Nset, nset=_PickedSet4, internal, instance=Part-1-1, generate 1, 589, 49**

**Elset, elset=_PickedSet4, internal, instance=Part-1-1, generate 1, 529, 48**

**Nset, nset=Set-1, instance=Part-1-1 49,**

. .

**Nset, nset=Set-11, instance=Part-1-1 539,**

**Nset, nset=Set-12, instance=Part-1-1 588,**

**Nset, nset=Set-13, instance=Part-1-1 637,**

**End Assembly**

**MATERIALS**

**Material, name=Material-1**

**Elastic**

**3e+07, 0.3**

**BOUNDARY CONDITIONS**

**Name: BC-1 Type: Symmetry/Antisymmetry/Encastre**

#4 Boundary Conditions

**Boundary _PickedSet4, ENCASTRE**

**STEP: Load**

#5 Load Step

**Step, name=Load**

**Static**

1, 1., 1e-05, 1.

**LOADS**

**
** Name: Load-1   Type: Concentrated force
*Cload
Set-1, 2, -9.83795
** Name: Load-2   Type: Concentrated force
*Cload
Set-2, 2, -37.037
** Name: Load-3   Type: Concentrated force
*Cload
Set-3, 2, -68.287
  .
  .
** Name: Load-11  Type: Concentrated force
*Cload
Set-11, 2, -68.287
** Name: Load-12  Type: Concentrated force
*Cload
Set-12, 2, -37.037
** Name: Load-13  Type: Concentrated force
*Cload
Set-13, 2, -9.83795
*Restart, write, frequency=0
**
** OUTPUT REQUESTS
****************************************************************
** #6 Output
************************************************************
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step

Appendix C: MATLAB Post-processing Program

% node_area: total SD area around the node
% elerst: element result matrix of original data
% eleid: element global ID
a=load('CS-FEM original output data file name');
elenod=csvread('Element & node connectivity fine name');
coord=csvread('Model nodal coordinate file name');

% Get total element number and node number
nele=length(elenod);
nnode=length(coord);

elerst=zeros(nele,29);
node_area=zeros(nnode,1);
node_strain=zeros(nnode,3);
node_stress=zeros(nnode,3);

% form element result matrix
for i=1:1:nele
    elerst(i,1:29)=a(1+29*(i-1):29*i);
end

% Initialize node number array and single element result array
nodsys=zeros(1,4);
sgelerst=zeros(1,29);

% loop for every element in system
for eleid=1:nele
    for i=1:4
        nodsys(i)=elenod(eleid,i+1);     % loop to get nodal global ID
    end
    for j=1:29
        sgelerst(j)=elerst(eleid,j);     % loop to get single element result
    end
    for i=1:4
        % loop to calculate node_area
        nodid=nodsys(i);
        node_area(nodid)=node_area(nodid)+sgelerst(1+i*7);
        node_strain(nodid,:)=node_strain(nodid,:)+sgelerst(i*7-5:i*7-3)*sgelerst(1+i*7);
        node_stress(nodid,:)=node_stress(nodid,:)+sgelerst(i*7-2:i*7)*sgelerst(1+i*7);
    end
end

for i=1:nnode
    node_strain(i,:)=(node_strain(i,:))/node_area(i);
    node_stress(i,:)=(node_stress(i,:))/node_area(i);
end