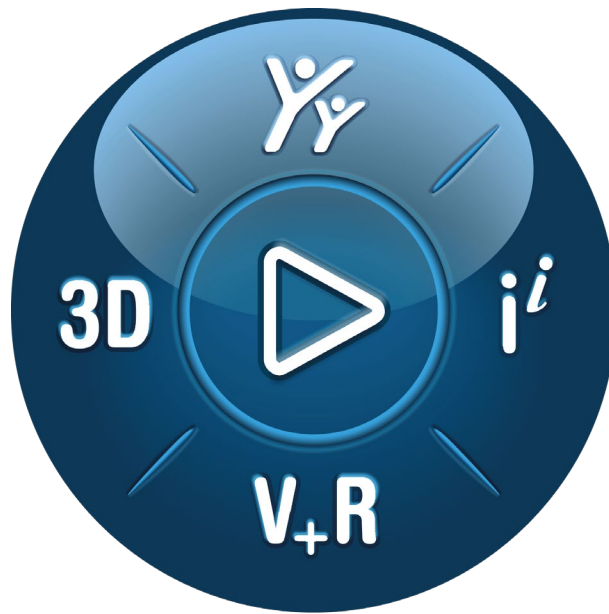


# Powder Compaction Simulation

With SIMULIA Abaqus



**3DEXPERIENCE®**

Version 1.0 - 8/21/2020

Written by: Shashwat SINHA  
Validated by: Changjian WANG  
Edited by: Arati DESAI

**SIMULIA Abaqus 2020 Golden**

## Executive Summary

***This Best Practices document describes the key concepts for finite element simulation of the process in which a column of powder is compacted to produce a tablet. A predictive understanding of the process is of crucial importance in the pharmaceutical industry, specifically in the prevention of tablet defects like chipping, sticking, picking, and so on. Since tablets are mass produced, and represent an easy means of drug delivery, defective tablets are not only undesirable but can potentially cause significant revenue loss. By conducting tableting simulations, process parameters like punch speed and lubrication can be studied numerically and optimized to minimize tablet defects.***

***The simulation procedure described herein is with SIMULIA Abaqus/Standard, which is a general-purpose finite element (FE) code. Abaqus has appropriate material models and modeling techniques that can accurately capture evolution of material properties in powder during compaction and can simulate the whole tableting process. We assume familiarity with the details of the powder compaction process. This document discusses in detail the material model used to represent the powder and its subsequent compaction along with all the details of finite element model.***

***We assume a strong facility with Abaqus concepts and a background in material modeling in finite element simulations.***

***Target audience: Abaqus users, analysts, process engineers***

### ***Disclaimer:***

- ***The finite element model described here is for demonstration purposes only. The actual problem and simulation conditions might differ in your scenario.***
- ***You must make appropriate engineering judgment before attempting such simulations.***
- ***Symbols and common terms used in material modeling are not explained.***

## Table of contents

1. Introduction .....	4
2. Material Model .....	4
2.1. Material Properties .....	7
2.2. Relative Density .....	10
2.3. Subroutine .....	10
3. Model Details .....	11
3.1. Analysis Steps .....	12
3.2. Analysis Procedure .....	13
3.2.1. Convergence .....	13
3.3. Boundary Conditions .....	14
3.3.1. Initial Boundary Conditions .....	14
3.3.2. Step Boundary Conditions .....	15
3.4. Contact .....	15
3.4.1. Friction .....	16
3.5. Setting up non-flat tablet surface model .....	17
4. Results .....	19
5. References .....	21
6. Document History .....	22

# 1. Introduction

Tablets or pills are widely used as a convenient means of dosage form for drug delivery to patients. Other than ease of use for patients, tablets are cheap to produce, easy to store, have good tolerance to environmental conditions, and are easy to transport. Given these benefits, pharmaceutical companies mass produce tablets in batches of millions. The key process in production of tablets is the uniaxial die compaction of pharmaceutical powder. In the process of powder compaction by the die, also called tableting, the tablets can sometimes develop defects such as chipping, sticking, picking, capping and lamination. The defective tablets may not be good for use and result in potential loss of revenue if the tableting process is not designed to minimize such defects.

To minimize the defects, it is of crucial importance to simulate the tableting procedure and understand the cause of defects through detailed stress analysis. This can help in optimizing process parameters like punch speed, lubrication of dies, compaction sequence, level of applied pressure, and so on. This can also affect the upstream process of powder formulation for better compacting properties. Thus, careful simulation can provide a key to understanding several related processes.

To perform a simulation that is representative of the tableting process, choosing the appropriate material model to represent the powder becomes crucial. Due to compaction being the central piece of the simulation, soil models are best suited to model the powder, as they can handle large amount of pressure based compaction. The Abaqus Modified Drucker-Prager with Cap hardening soil model has all the necessary features to capture the process of compaction. The yield surface in this model evolves based on powder compaction.

This document discusses this material model and all other aspects of finite element modeling for powder compaction simulation. All salient aspects of modeling are discussed which can be applied in a real life model. It should be noted that the model described is only for demonstration purposes, and must be treated only as a reference for general guideline in setting up such simulations.

## 2. Material Model

Among the soil models available in Abaqus, the Modified Drucker-Prager model with Cap Hardening (DPC) is the most suitable model to capture powder compaction in numerical simulations. It can correctly capture the inelastic compaction that occurs during the tableting process. The yield surface accounts for both shear failure and hydrostatic compression yielding, which adequately describes compaction.

In addition to the right material model, the evolution of the inelastic material parameters based on Relative Density (RD) is critical in correctly simulating powder compaction. The material properties at any stage of compaction must be consistent with the relative density of the material at that point. This is achieved by making the DPC material parameters dependent on relative density through a field variable. By including dependency on a field variable representing RD, the material properties are constantly updated during the analysis.

## Best Practices

A uniform relative density within the tablet after compaction ensures that tablets are less susceptible to defects.

As is obvious, detailed testing of the material [2] is a prerequisite to perform such simulations successfully. Due to complex interdependence and evolution of material properties during analysis, any error in material properties can lead to incorrect or unconverged analyses.

The schematic of the yield surface of the DPC model is as shown in Fig 2.1 and the hardening curve is shown in Fig 2.3.

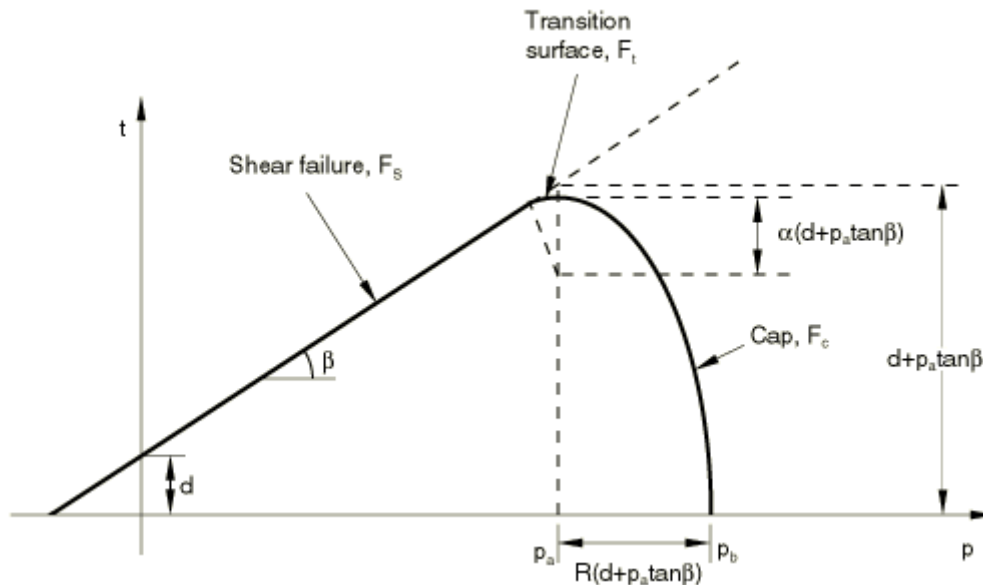


Fig 2.1: Yield Surface of DPC model

The failure surface is given by the following formula:

$$F_s = t - p \tan \beta - d = 0$$

where,  $\beta = \text{angle of friction}$  and  $d = \text{material cohesion}$ ;

$$t = \frac{1}{2} q \left[ 1 + \frac{1}{K} - \left( 1 - \frac{1}{K} \right) \left( \frac{r}{q} \right)^3 \right]$$

and,

$$p = -\frac{1}{3} \text{trace}(\sigma) \quad \text{is the equivalent pressure stress,}$$

$$q = \sqrt{\frac{3}{2} \mathbf{S} : \mathbf{S}} \quad \text{is the Mises equivalent stress,}$$

## Best Practices

$r = \left(\frac{9}{2}\mathbf{S}:\mathbf{S}:\mathbf{S}\right)^{\frac{1}{3}}$  is the third stress invariant, and

$\mathbf{S} = \boldsymbol{\sigma} + p\mathbf{I}$  is the deviatoric stress.

$K$  is a material parameter that controls the dependence of the yield surface on the value of the intermediate principal stress, as shown in Fig. 2.2.

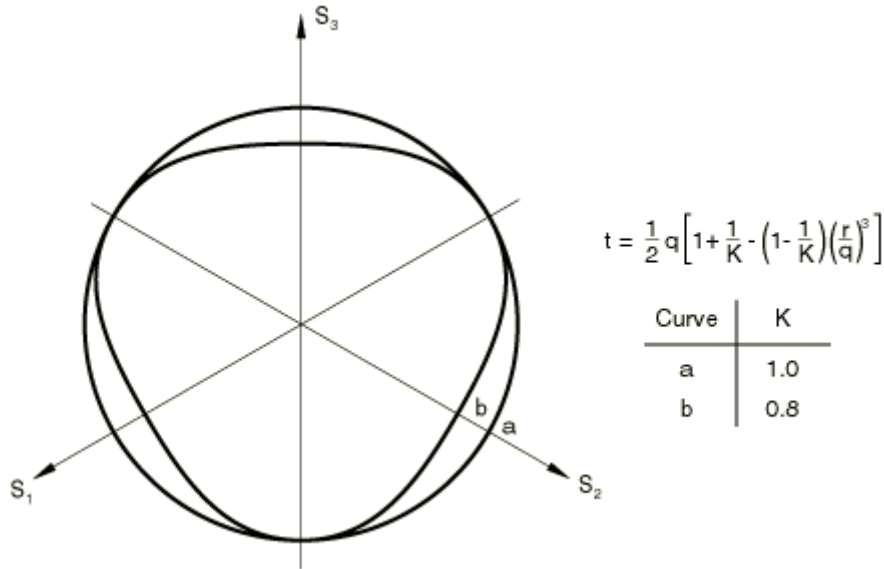


Fig 2.2: Yield surface in pi-plane

The cap added to the yield surface not only bounds the yield surface in hydrostatic compression but also provides an inelastic hardening mechanism to represent plastic compaction [1]. This feature is useful in the tableting process where, during compaction, both shearing along the die walls and compression of the powder is happening simultaneously. In addition, friction between grains in the powder contributes to the shearing effect and yielding in shear.

The cap yield surface has an elliptical shape with constant eccentricity, and hardens or softens based on volumetric inelastic strain. It is given by:

$$F_c = \sqrt{[p - p_a]^2 + \left[\frac{Rt}{(1 + \alpha - \alpha/\cos\beta)}\right]^2} - R(d + p_a \tan\beta) = 0$$

where,

$R$  is cap eccentricity parameter that controls the shape of the cap

$\alpha$  is transition surface radius parameter, a small number compared to 1

$p_a$  is an evolution parameter that represents inelastic strain driven hardening/softening

$$p_a = \frac{p_b - Rd}{(1 + R \tan\beta)}$$

where,

$p_b$  is hydrostatic compression yield stress

The cap hardening definition relates  $p_b$  to volumetric inelastic strain and typically takes a curve as shown in Fig 2.3. The volumetric inelastic strain is given by:

$$\varepsilon_{vol}^{in} = \varepsilon_{vol}^{in}|_0 + \varepsilon_{vol}^{pl} + \varepsilon_{vol}^{cr}$$

The volumetric inelastic strain in the hardening curve has an arbitrary origin, that is,

$$\varepsilon_{vol}^{in}|_0 = \varepsilon_{vol}^{pl}|_0 + \varepsilon_{vol}^{cr}|_0$$

This corresponds to the initial state of the material, thus defining the position of cap ( $p_b$ ), at the start of the analysis.

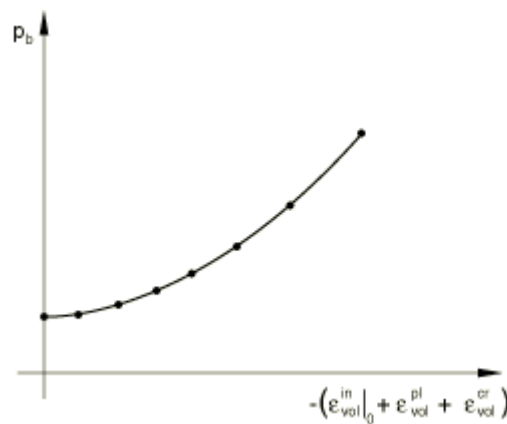


Fig 2.3: Cap hardening curve

Note that  $p_a$  is based on the value of  $p_b$  as its evolution depends on the hardening curve defined. For more details on transition surface and plastic flow, refer to the SIMULIA User's Guide [1].

The powder also exhibits non-linear elastic behavior due to the dilation phenomenon in unloading, which is important for accurate modeling of tablet compaction.

### 2.1. Material Properties

The material properties used in the simulation are detailed here and are obtained from Zavaliango [2]. The data for the main inputs of cap plasticity and cap hardening are as follows:

**\*Cap Plasticity, dependencies=1**

$d, \beta, R, \varepsilon_{vol}^{in}|_0, \alpha, K, Temp, FV1$

$Temp$  is not used and  $FV1$  is the field variable representing RD.

**\*Cap Hardening, dependencies=1**

$p_b, abs(\varepsilon_{vol}^{in}), Temp, FV1$

## Best Practices

*Temp* is not used and *FV1* is the field variable representing RD. For demonstration purposes, cap hardening values are defined for each value of relative density.

The properties needed for complete material definition are described below in the keyword format of Abaqus input file. Units are N-mm-sec.

**\*Material, name=Tablet\_Orig**

**\*Cap Plasticity, dependencies=1**

0.1,	40.,	0.05,	0.02,	0.01,	1.,	, 0.3
0.2,	45.,	0.075,	0.02,	0.01,	1.,	, 0.35
0.3,	48.,	0.1,	0.02,	0.01,	1.,	, 0.4
0.5,	52.,	0.13,	0.02,	0.01,	1.,	, 0.45
0.9,	55.,	0.16,	0.02,	0.01,	1.,	, 0.5
1.25,	57.,	0.21,	0.02,	0.01,	1.,	, 0.55
1.7,	59.,	0.25,	0.02,	0.01,	1.,	, 0.6
2.,	60.,	0.31,	0.02,	0.01,	1.,	, 0.65
2.6,	64.,	0.4,	0.02,	0.01,	1.,	, 0.7
3.4,	65.,	0.45,	0.02,	0.01,	1.,	, 0.75
4.3,	67.,	0.52,	0.02,	0.01,	1.,	, 0.8
5.4,	69.,	0.6,	0.02,	0.01,	1.,	, 0.85
6.7,	71.,	0.66,	0.02,	0.01,	1.,	, 0.9
8.1,	72.5,	0.78,	0.02,	0.01,	1.,	, 0.95
8.5,	73.,	0.84,	0.02,	0.01,	1.,	, 1.

**\*Cap Hardening, dependencies=1**

1.,	0.,	, 0.30
1.1,	0.154151,	, 0.30
**		
1.1,	0.,	, 0.35
1.1001,	0.154151,	, 0.35
2.5,	0.287682,	, 0.35
**		
2.5,	0.,	, 0.4
2.5001,	0.287682,	, 0.4
5.0,	0.405465,	, 0.4
**		
5.,	0.,	, 0.45
5.001,	0.405465,	, 0.45
7.50,	0.510826,	, 0.45
**		
7.5,	0.,	, 0.5
7.5001,	0.510826,	, 0.5
10.0,	0.606136,	, 0.5
**		
10.,	0.,	, 0.55
10.001,	0.606136,	, 0.55



## Best Practices

15.0,	0.693147,	,	0.55
**			
15.,	0.0,	,	0.6
15.001,	0.693147,	,	0.6
19.0,	0.77319,	,	0.6
**			
19.,	0.,	,	0.65
19.001,	0.77319,	,	0.65
25.0,	0.847298,	,	0.65
**			
25.,	0.,	,	0.7
25.001,	0.847298,	,	0.7
30.0,	0.916291,	,	0.7
**			
30.,	0.,	,	0.75
30.001,	0.916291,	,	0.75
43.0,	0.980829,	,	0.75
**			
43.,	0.,	,	0.8
43.001,	0.980829,	,	0.8
60.0,	1.04145,	,	0.8
**			
60.,	0.,	,	0.85
60.001,	1.04145,	,	0.85
80.0,	1.09861,	,	0.85
**			
80.,	0.,	,	0.9
80.001,	1.09861,	,	0.9
120.0,	1.15268,	,	0.9
**			
120.,	0.,	,	0.95
120.001,	1.15268,	,	0.95
1500.0,	1.20397,	,	0.95
**			
1500.,	0.,	,	1.
1500.001,	1.20397,	,	1.

### \*Elastic, dependencies=1

100.,	0.024,	,	0.3
150.,	0.037,	,	0.35
250.,	0.0385,	,	0.4
400.,	0.05,	,	0.45
500.,	0.0625,	,	0.5
900.,	0.08,	,	0.55
1200.,	0.1,	,	0.6
1500.,	0.12,	,	0.65
2200.,	0.139,	,	0.7



Confidential information. Copyright Dassault Systèmes [2020].

This document is provided for information purpose only and may be used by authorized users only for their own internal use. Any other use without prior written authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

## Best Practices

3000.,	0.17, ,	0.75
3900.,	0.19, ,	0.8
5000.,	0.225, ,	0.85
6400.,	0.25, ,	0.9
8000.,	0.29, ,	0.95
9000.,	0.31, ,	1.

Based on the value of the field variable (given by “dependencies” in the last column) representing RD, the material properties are interpolated to get the correct value for a given increment. The field variable is invoked by keyword **\*Depvar** used with the keyword to invoke the subroutine:

### **\*User Defined Field**

#### **\*Depvar**

1

Both keywords are used under the **\*Material** keyword.

## 2.2. Relative Density

As the powder is compacted, the relative density of the powder continuously increases. Plasticity parameters in the material model continuously evolve with relative density. It is directly a function of initial relative density, height of the powder column and distance between top and bottom punch [2]. The increase of relative density can be well approximated with the evolution of volumetric plastic strain if the creep strain is zero. The relation between volumetric plastic strain and relative density [3] is given by:

$$\varepsilon_v^p = \ln(RD/RD_0)$$

where  $RD_0$  is the initial relative density of the powder.

The initial relative density can be specified in the input file by using a parameter:

#### **\*Parameter**

```
initRD = 0.3
```

To calculate RD in every increment, and subsequently affect the material properties, a user subroutine is employed to capture this specific form of evolution.

## 2.3. Subroutine

To accomplish the variation of material properties based on relative density, user subroutine USDFLD is used. In this subroutine, evolution of RD is numerically determined in every increment based on the volumetric plastic strain obtained from the analysis. The relative density is then stored in a field variable (FV1), which is referred to as “dependencies” in the material definition.

The subroutine USDFLD is invoked by the keyword:

### **\*User Defined Field**

#### **\*Depvar**

1

The initial value of relative density defined as a parameter and is read in the subroutine through a parameter table, which can be defined as shown below:

#### **\*Parameter Table, type=RD**

<initRD>

Since this subroutine provides access to material point quantities only at the start of the increment, the solution dependence introduced in this way is explicit: the material properties for a given increment are not influenced by the results obtained during the increment. Hence, the accuracy of the results depends on the size of the time increment [1].

You can either write the subroutine on their own, or obtain the compiled form along with the example model described here in form of an Abaqus input file for a nominal cost. Please contact your sales rep for more details.

## 3. Model Details

The model described here is based on Han et al [4] in which a tablet with a flat top and bottom surface is produced after powder compaction. The final shape of the tablet is like a round disc. To accomplish the full tableting simulation, several steps are created in addition to the main compaction step. The compaction of the powder is done by an upper punch, which compresses the powder in a die. The schematic of the model is shown in Fig 3.1. The model consists of 4 parts:

- Powder
- Die
- Punch-Upper
- Punch-Lower

Other than the powder, all parts are analytical rigid parts.

Considering the cylindrical column of powder, axial symmetry is utilized to create an axisymmetric model. The powder column is meshed with fully integrated axisymmetric elements (CAX4).

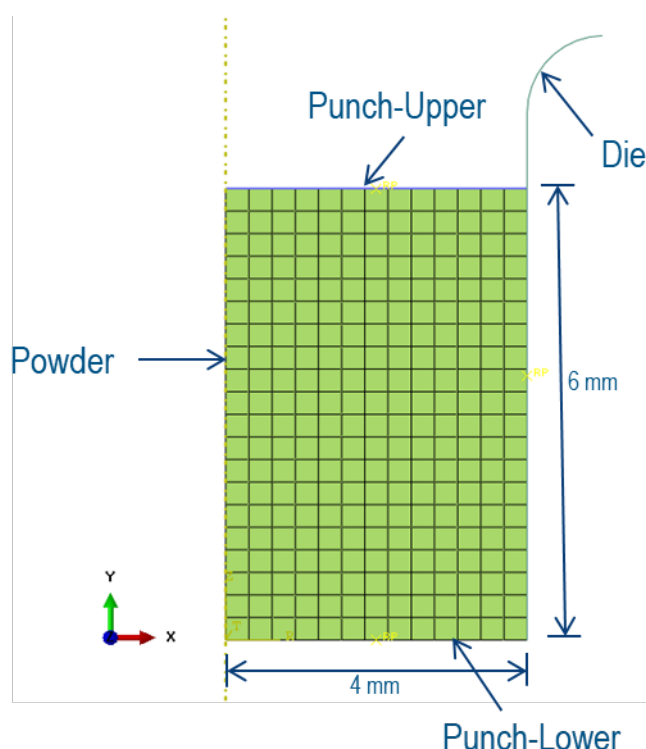


Fig 3.1: Schematic of finite element model

The mesh is adequately refined considering the elements used are fully integrated. If reduced integration elements are used, a mesh convergence study should be completed.

### 3.1. Analysis Steps

The main steps that define the tableting process during the analysis are as follows:

- **Compact**  
The Compact step compacts the powder column and turns it into a tablet. In this step, the punch moves down on the powder column until a specified distance. As the powder compacts, its relative density increases and a tablet is formed at the end of the step.
- **Release**  
The Release step removes the top punch and allows the tablet to relax. In this step, the elastic deformation of the tablet obtained after compaction step is recovered, and the tablet assumes its final shape.
- **Eject**  
The Eject step pushes the tablet gently out of its current location after Release step. This signifies ejection of the tablet from the die and completes the tableting process.

In addition to the above steps, two additional steps can be defined after the Compact and Eject steps if there are convergence issues in the Compact and Eject steps:

- Compact-nostab
- Eject-nostab

Both these “nostab” steps are defined to remove the effects of stabilization that occur in the Compact and Eject steps. Stabilization is applied to obtain a converged solution in the Compact and Eject steps, discussed in section 3.2.1.

In terms of keywords, the steps are summarized as follows:

```
*Step, name=Compact, nlgeom=YES, inc=100000
```

```
*Step, name=Release, nlgeom=YES, inc=100000
```

```
*Step, name=Eject, nlgeom=YES, inc=100000
```

## 3.2. Analysis Procedure

The dynamic effects in the compaction process are not significant, so the static procedure (\*Static) is used to define all Abaqus steps. This ensures full equilibrium and converged solution at all stages of the analysis. The static procedure is defined as follows:

```
*Static  
0.01, 1., 1e-10, 0.01
```

It is important to note that the maximum time increment must not be set to 1. Due to the exponential evolution of relative density (see section 2.2), which is highly non-linear, maximum size of an increment should not be large as the solution may diverge to produce incorrect solution. In this case, maximum increment size is set to 0.01, which is reasonably small to prevent any divergence in solution.

### 3.2.1. Convergence

In the model described here, convergence issues do not occur; however, with the complexity of the material model and initial low resistance of the powder to compaction it might be sometimes difficult to achieve convergence in the Compact step. If convergence issues occur, then to improve the efficiency of this highly discontinuous behavior, the following may be used as part of step definition:

```
*Controls, analysis=discontinuous
```

As the soft powder gets compacted by the upper punch, the process may also lead to some local instabilities, causing non-convergence. This can be addressed by applying stabilization to the static step. A similar situation of non-convergence may occur in the Eject step, when the tablet is pushed out. In this case, contact with the top punch is removed and the non-convergence may arise due to under-constrain on the tablet. Stabilization can be again used here to obtain a converged solution.

```
*Static, stabilize, factor=1.e-4
```

```
0.01, 1., 1e-10, 0.01
```

To get an estimate of the “factor” used in stabilization, the analysis can be run with “**\*Static, stabilize**” (without “**factor**”) for just two increments and then killed. At the end of the first increment, Abaqus prints its estimate of the stabilization factor in the message (.msg) file. This provides a ballpark value of the factor to start with. To minimize stabilization effects, lower values of the factor parameter (reducing by one order) can be tested. This is an iterative process to get to the minimum value needed for the analysis. In some cases, the value given by Abaqus may also need to be increased.

Using stabilization can have an effect of producing artificial viscous forces in the model. Though small in magnitude, it can constrain the deformation from what it should be without stabilization. To remove the artificial effects of viscous forces generated by stabilization in these steps, follow-up “do-nothing” steps can be added after the Compact and Eject steps. These steps (Compact-nostab and Eject-nostab) can be run for a long step time so the viscous forces are slowly ramped down.

```
*Step, name=Compact-nostab, nlgeom=YES, inc=100000
```

```
*Static, stabilize, continue=yes, allsdtol
```

```
0.01, 10000., 1e-10
```

In cases when convergence cannot be obtained by stabilization, the following convergence criteria can be used:

```
*Controls, parameters=field, field=displacement  
,1.0
```

For more details on this topic, refer to the SIMULIA User’s Guide [1].

### 3.3. Boundary Conditions

#### 3.3.1. Initial Boundary Conditions

To fully constrain the model, boundary conditions are applied initially to fix the die, the upper punch and the lower punch in their respective position. To achieve this, the active degrees of freedom (1, 2, 6) at the reference points of the rigid bodies of the punches and the die are fixed. The case for the die is shown below:

```
*Boundary
```

```
Die-1.die_RP, 1, 1
```

```
Die-1.die_RP, 2, 2
```

```
Die-1.die_RP, 6, 6
```

In the same way, the upper punch and the lower punch are constrained.

For axial symmetry, the left side nodes of the powder are constrained in degree of freedom 1.

### **\*Boundary**

```
Tablet-1.axial, 1, 1
```

The other degrees of freedom for the powder are automatically constrained by contact.

To move the punches up or down later in the analysis, these boundary conditions are modified in subsequent steps.

### 3.3.2. Step Boundary Conditions

In addition to initial boundary conditions, boundary conditions are also applied in the Compact, Release, and Eject steps. The step names are self-explanatory for the purpose of the step.

- For step Compact, the following boundary condition is applied:

#### **\*Boundary**

```
Punch-Upper-1.punch_RP, 2, 2, -4.
```

This shows that the upper punch is pushed down by 4 mm onto the powder to compact it from its original position.

- For step Release, the following boundary condition is applied:

#### **\*Boundary**

```
Punch-Upper-1.punch_RP, 2, 2
```

This shows that the upper punch is brought back to its original position.

- For Eject step, the following boundary condition is applied:

#### **\*Boundary**

```
Punch-Lower-1.punch_RP, 2, 2, 0.5.
```

This shows that the lower punch pushes the tablet up by 0.5 mm indicating ejection of the tablet.

It should be noted that in Abaqus the displacements applied for boundary conditions are always “total” measured from the initial position, and not incremental.

## 3.4. Contact

Contact interactions between different parts of the model are defined through contact pairs. In most cases, the default penalty constraint enforcement method of contact is sufficient. If

there are difficulties in contact convergence initially, using a nonlinear penalty constraint sometimes helps. In this case, a hybrid approach is adopted. For contact between the die wall and powder, a nonlinear penalty formulation is used to define the surface behavior and is adequate for contact convergence. It is defined as follows:

```
*Surface Behavior, penalty=NONLINEAR
```

For punch and powder contact, to avoid any penetration of punch into the top powder surface, the Lagrange constraint enforcement method is used, which does not allow penetration of stiffer surface into the other softer surface. As the powder is softer and has low resistance to compaction, the rigid punch can penetrate the top powder surface during the analysis. Use of the Lagrange constraint avoids this penetration providing more accurate results. However, this formulation is expensive and can take some time to converge, so it should be used with caution. It is defined as follows:

```
*Surface Behavior, direct
```

In the Eject step, contact definition between the upper punch and the top surface of the tablet is removed:

```
*Model Change, type=CONTACT PAIR, remove
```

```
Tablet-1.top, Punch-Upper-1.punch_Surf
```

### 3.4.1. Friction

The friction definition is crucial in terms of overall response of powder compaction; it represents the amount of lubrication applied to the die. Since there is no sliding between the punches and the powder surfaces at the top and bottom, a frictionless contact interaction is assumed. For the die wall surface and the corresponding powder surface, friction is introduced in the contact definition. Typically, as the powder is compacted, the friction between the die wall and the powder reduces based on contact pressure [2]. To capture this effect, friction is defined as a function of contact pressure. The friction definition within the overall contact definition is as follows:

```
*Surface Interaction, name=Fric_contPress
```

```
1.,
```

```
*Friction, slip tolerance=0.005
```

```
0.985, , 0.000
```

```
0.944, , 0.214
```

```
0.740, , 0.427
```

```
0.635, , 0.854
```

```
0.504, , 1.068
```



## Best Practices

0.444, , 1.708

0.391, , 2.135

0.345, , 3.416

0.311, , 5.125

0.283, , 7.900

0.255, , 9.822

0.225, , 13.238

0.206, , 16.655

0.187, , 19.644

0.157, , 31.601

0.142, , 36.513

0.129, , 44.626

0.120, , 53.808

0.116, , 60.641

0.109, , 78.577

0.105, , 94.804

0.103, , 102.491

### 3.5. Setting up non-flat tablet surface model

The main issue in setting up models that have non-flat top and bottom tablet surfaces is to make sure that the analysis is able to converge initially. In many cases, a thin ledge is present around the perimeter of the tablet that can cause localized deformation and not allow the analysis to converge. An example of such tablet, with a doubly curved surface, is shown in Fig 3.2:

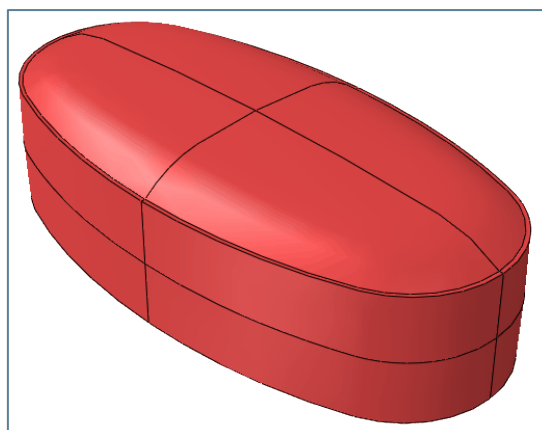


Fig 3.2: Tablet with doubly curved top and bottom surface

Two approaches can be adopted to get around this problem:

- In the first approach, the initial shape of the top powder column surface can be assumed to correspond to the shape of the non-flat punch. Even though it is not physically representative of initial condition of top flat powder surface, it is a sufficiently representative approximation. In this case, the compaction pressure is more evenly distributed on the powder surface. Ultimately, with full compaction, the final results are correct. In some cases, due to the presence of a thin flat ledge around the perimeter of the tablet (see Fig 3.2), it may still produce localized high forces which can cause elements to distort and cause convergence difficulties; in that case the second approach can be used.
- In the second approach, the initial shape of the top powder column surface is flat, as it will be physically, before the compaction begins. In this case, part of the compaction can be performed with a flat punch, and in the later stage, punch with actual tablet shape can be introduced. In the analysis, contact with the flat punch can be removed and the curved punch can be activated. The curved punch must be precisely placed at the location where flat punch will stop. After initial compaction, the powder develops resistance to further compaction and the compaction with the curved punch can proceed without any element distortion or convergence issues.

## 4. Results

Results for Von Mises stress and relative density distributions are shown in the following images.

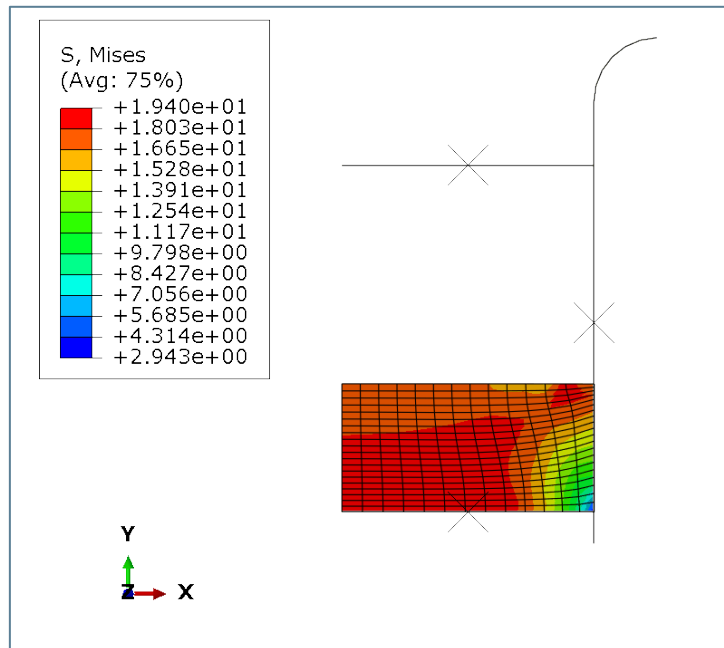


Fig 4.1: Final Mises stress distribution in the tablet

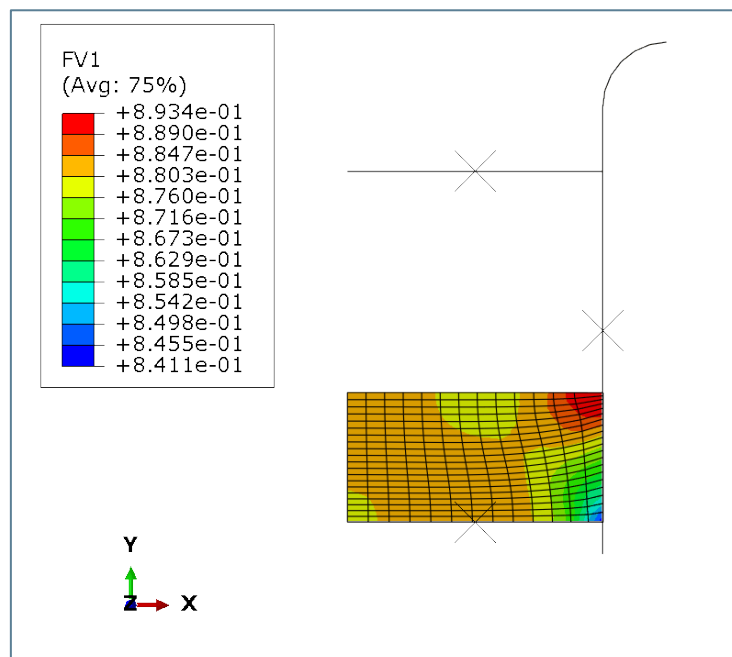


Fig 4.2: Final relative density distribution in the tablet

To understand the results better, recall the configuration of the model set up. It is an axisymmetric model and the axis of revolution is aligned with the left side of the tablet,

## Best Practices

parallel to the die, and an axisymmetric constraint is applied there. This implies that the center of the tablet is on the left side, and the perimeter of the tablet is aligned with the die.

It can be seen from the results that the relative density is mostly uniformly distributed in the tablet (within 2%). This shows that the compaction occurred satisfactorily. The sharpest change in relative density variation occurs around the perimeter of the tablet. That also coincides with maximum variation of Von Mises stress. The perimeter thus can be a potential region for tablet defects to occur. However, it can be observed that the variation of relative density is still within 5%, along the perimeter, which can be considered satisfactory. Changes in process parameters can possibly bring this variation further down.

## 5. References

- SIMULIA User's Guide 2020
- Cunningham J.C., Sinka I.C., Zavaliangos A., 2004, "Analysis of tablet compaction. I. Characterization of mechanical behavior of powder and powder/tooling friction," Journal of Pharmaceutical Sciences, Vol 93, No. 8, pp. 2022-39
- Ahmad Baroutaji, Keith Bryan, Mustafa Sajjia, Sandra Lenihan, 2017, "Mechanics and Computational Modeling of Pharmaceutical Tableting Process," Reference Module in Materials Science and Materials Engineering. pp. 1–10, Elsevier
- Lianghao Han, James Elliott, Serena Best, Ruth Cameron, A.C. Bentham, A. Mills, G.E. Amidon and B.C. Hancock, 2008, "Numerical Simulation on Pharmaceutical Powder Compaction," 5th International Conference on Physical and Numerical Simulation of Material Processing (ICPNS 07)

## 6. Document History

Document Revision	Date	Revised By	Changes/Notes
1.0	08/21/2020	Shashwat Sinha	Original