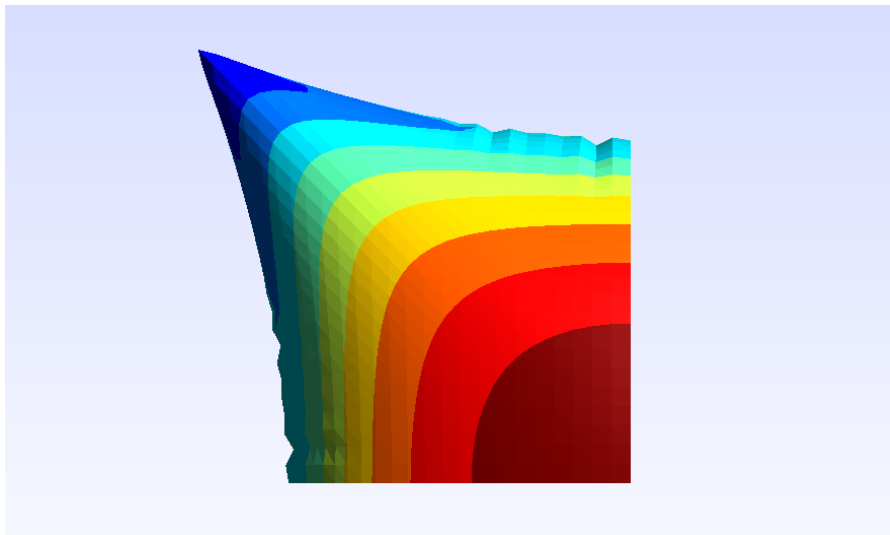


Tutoriel : Gonflage coussin

Etude de l'algorithme de relaxation dynamique avec un amortissement cinétique et ou visqueux

Gaëtan ROMAN

30/05/2017



Sommaire

I)	Introduction :	3
II)	Gonflage d'un coussin	3
a)	Création de la géométrie :	3
b)	Calcul :	4
c)	Visualisation :	7
III)	Analyse des différents amortissements	8
a)	Amortissement cinétique	9
b)	Amortissement visqueux	9
IV)	Conclusion	10

I) Introduction :

L'objectif de ce document est de parvenir à gonfler un coussin via un algorithme de relaxation dynamique. Nous utiliserons pour cela un amortissement cinétique et ou visqueux possédant différents paramètres afin d'étudier leurs comportement et leurs rapidités en temps de calcul. L'algorithme de relaxation dynamique est une technique principalement utilisé pour des structures souples (câbles, nappes etc ...) permettant la recherche de forme.

Ci-joint le dossier permettant de vous aider, expliqué en fin du tutoriel.

II) Gonflage d'un coussin

a) Création de la géométrie :

Dans un premier temps, il est nécessaire de créer la géométrie du coussin possédant un maillage de type triangle linéaire. Ce coussin sera plat donc un carré (2D) de dimension 250x250 mm avec 25 éléments selon sa longueur et sa largeur. Pour le créer nous utiliserons stamm avec les choix ci-dessous (figure 1) :

```

type d'elements : (1D, 2D, 3D) ? 2D
choix lu: 2D
type de decoupage :
rectangulaire           (reponse r ) ?
triangulaire           (reponse t ) ?
treilli rectangulaire  (reponse tr ) ?
treilli triangulaire   (reponse tt ) ?
rectangulaire axi-symetrique (reponse ra ) ?
triangulaire axi-symetrique (reponse ta ) ? t
choix de l'interpolation :
lineaire               (reponse li ) ?
quadratique complet   (reponse qc ) ?
cubique incomplet     (reponse ci ) ?
cubique complet       (reponse cc ) ? li
calcul de la position des noeuds :
de maniere exacte     (reponse e ) ?
de maniere aleatoire autour d'une position exacte (reponse a ) ?
aleatoire sur la geometrie sauf frontieres et symetries (reponse s ) ?
aleatoire sur la geometrie uniquement noeuds internes (reponse i ) ? e
type de geometrie :
rectangle              (reponse rec ) ?
parallelogramme       (reponse prl ) ?
plaque en helice      (reponse phe ) ?
cylindre              (reponse cyl ) ?
portion de cylindre   (reponse pcy ) ?
disque                (reponse disq ) ?
portion angulaire de disque (reponse a_disq ) ?
dome hemispherique complet (reponse dome ) ?
tranche horizontale de dome hemispherique (reponse p_dome ) ?
tranche angulaire de dome hemispherique (reponse a_dome ) ?
anneau                (reponse ann ) ? rec
dimension du rectangle: longueur x ? 250
largeur ? 250
nombre d'element dans la longueur ? 25
dans la largeur ? 25
nom du fichier de sortie ? : coussin
nom lue : coussin
un autre maillage ? (rep o ou n ) n

```

Utilisation d'éléments 2D

Eléments de type triangle

Une interpolation linéaire

Le calcul de la position des noeuds se fera de manière exacte

Notre coussin est un rectangle

Dimension de notre plaque

Nombre d'élément

Nom du fichier possédant le maillage en sortie

Figure 1: Stamm

Sous `hz_visuMail.pl` il est alors possible de visualiser le maillage obtenu avec la commande suivante « `hz_visuMail.pl nomfichier.her` ». Nous visualisons cela (figure 2) :

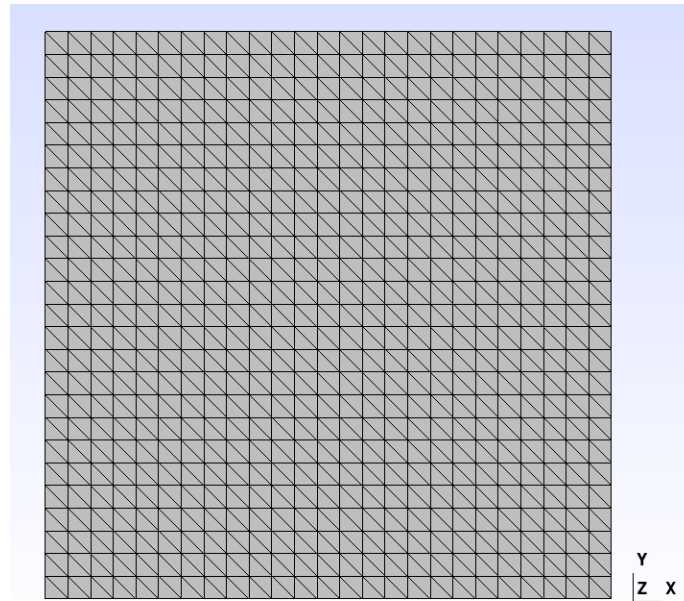


Figure 2: Maillage du coussin

On observe bien un maillage de type triangle linéaire. D'ailleurs comme demandé celui-ci possède 25 éléments selon sa longueur et sa largeur.

b) Calcul :

Nous allons expliquer la mise en donnée pour la mise en place de conditions de symétries pour 1/8 du coussin et l'ajout d'un effort de type pression. Le principe est le suivant (figure 3):

- Blocage de certains nœuds (voir figure 3)
- Application de la pression

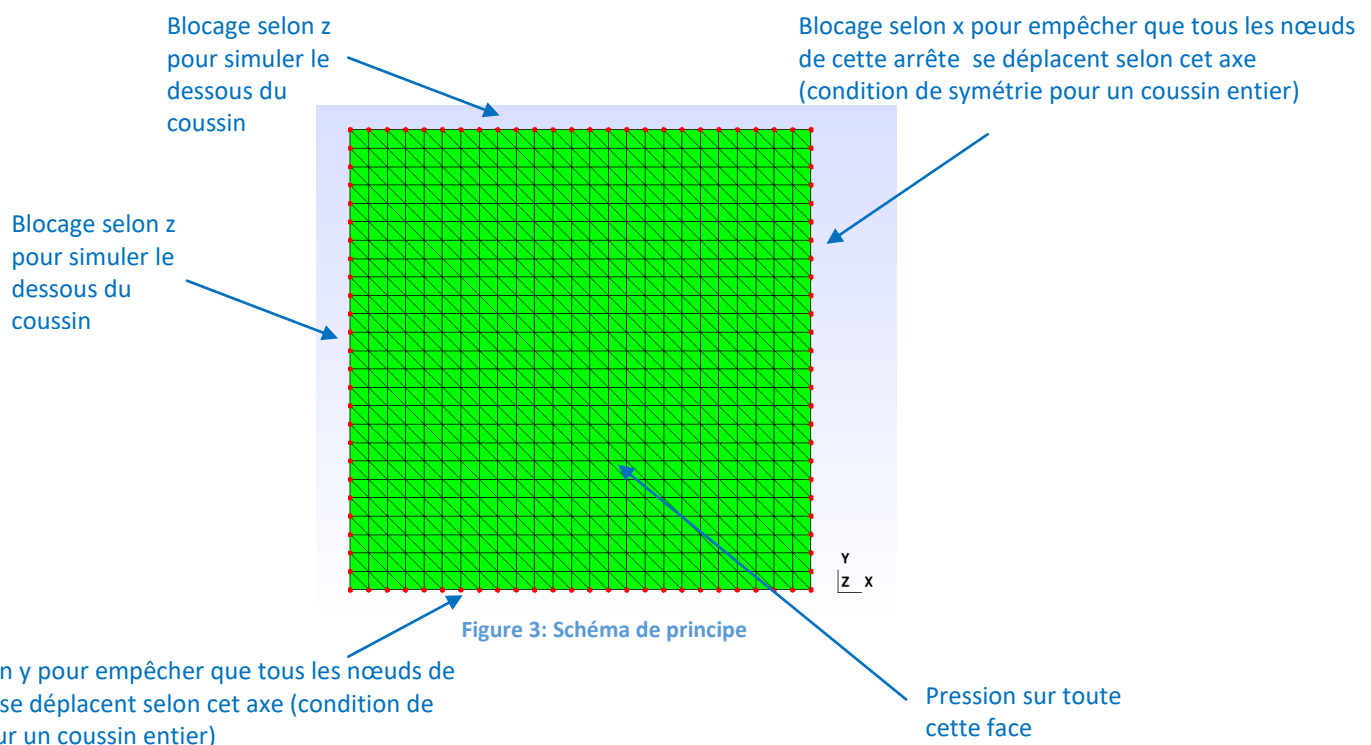


Figure 3: Schéma de principe

La position initiale du coussin est à plat. Lors de la pression et grâce aux conditions de symétries cela va permettre de simuler le gonflage du coussin. Ainsi la position finale sera lorsque la portion de ce coussin sera gonflé.

Explication de la mise en donnée :

```

#-----
# Definition of the dimension
#-----
dimension      3          ← Dimension de l'espace de
                           travail

#-----
# Level of comments: 1-10, the bigger, the more comments are
# shown in the Herezh++ terminal during the calculation
#-----
niveau_commentaire      3#7 ← Niveau de commentaire
                           variable

#-----
# Definition of the calculation type
#-----
TYPE_DE_CALCUL          Type de calcul ici relaxation
dynamique_relaxation_dyn # avec plus visualisation ← Mot clé : avec plus visualisation :
                           dynamique                               ⇒ permet de créer le fichier de sortie
                           permettant la visualisation des
                           résultats (fichier.Cvisu)

# ----- Specific parameters for this calculation type
PARAM_TYPE_DE_CALCUL

# ----- Different dynamic relaxation methods (choose one -uncomment-)
#< KDR1.algo
< KDR.algo              Ne pas modifier les paramètres de
#< oDR.algo              ces amortissements !!
#< mDR.algo
#< mdDR.algo            ← Les différents fichiers permettant
#< uqDR.algo            d'utiliser un amortissement cinétique et
#< pqDR.algo            ou visqueux

#-----
# Mesh definition
#-----
< TL_25x25.her         ← Maillage du coussin

#-----
# Material properties
#-----
choix_materiaux -----
#-----
# Elements' reference(s) | Name of material
#-----
E_to polymere         ← Nom du matériau

materiaux -----
#-----
# Name of material | Type of law of behavior ← Nom de la loi utilisé
#-----
polymere              ISOELAS2D_C      # isoelastic 2D plane stress

```

```

#-----
# Young's modulus | Poisson's ratio
#-----
125 0.41
#-----
# ----- Density (in ton/mm3)
masse_volumique -----
E_to 1e-09
# ----- Thickness (when using membrane/shell elements)
epaisseurs -----
E_to 0.27

#-----
# Loading
#-----
charges -----
#-----
# Reference faces | Loading type | Value |
#-----

# ----- Pressure (in MPa)
F_to PRESSION -0.015 TEMPS_MINI= -1.
#-----
# Boundary conditions
#-----

blocages -----
#-----
# References nodes | Blocking
#-----
# ----- Two sides encastred and two sides with symmetries
N_O UZ
N_S UY
N_E UX
N_N UZ

typecharge -----
#-----
# Type | Time |
#-----
# ----- Direct application
TYPE3

#-----
# Calculation and control settings
#-----

# ----- Control parameters
controle -----
#-----
# PARAMETER | VALUE |
#-----
#SAUVEGARDE 0 # If we plan to make a restart, put 1 to save
intermediate calculation increments

```

Propriété du matériau

Caractéristiques du matériau

Pression de 0,015MPa sur toute la face du coussin

Nos conditions de symétries

Pour prendre en compte un chargement il faut également indiquer un temps mini inférieur à 0 pour éviter l'exclusion du temps 0 (Cela est défini par le temps mini)

L'ensemble du chargement est imposé de manière constante dans le temps (voir courbe ci-dessous)

```
#DELTAatMAXI 1.          # Size limit for the time step
TEMPSFIN 1.              # Time at which calculation will end
DELTAat 1.              # Time step: in DR, arbitrarily fixed to 1
NORME E_cinetique/E_statique_ET_ResSurReact          # Convergence
criterion
PRECISION 1e-3          # Value for the convergence criterion
MAXINCRE 1              # Maximum number of increments: if we don't make an
incremental calculation, 1 is ok
ITERATIONS 4000        # Maximum number of iterations before considering
that it diverges
#RESTART 1200          # If we want to restar the calculation from a
particular increment

# ----- Display settings
para_affichage -----
#-----
# PARAMETRE      |      VALEUR      |
#-----
FREQUENCE_AFFICHAGE_INCREMENT 1          # Frequence of display of
increments (1 = show all)
FREQUENCE_AFFICHAGE_ITERATION 100        # Frequence of display of
iterations (25 = show one out of 25)
FREQUENCE_SORTIE_FIL_DU_CALCUL 1        # Frequence of output generation
(1 = at each increment)

resultats -----
#-----
# PARAMETRE | VALEUR |
#-----
COPIE 0

      _fin_point_info_ -----
```

En sortie du fichier.info avec le paramètre « *avec plus visualisation* » on récupère la déformée du coussin mais aussi les contraintes de mises. Ainsi sous gmsh nous pourrons visualiser les contraintes et le gonflage du coussin.

c) Visualisation :

Nous allons visualiser le résultat obtenu grâce à la mise en donnée précédente. Pour cela il suffit d'ouvrir le fichier « nom_deplace_Gmsh.pos » avec gmsh. Puis faire quelques modification dans les options comme ci-contre (figure 4):

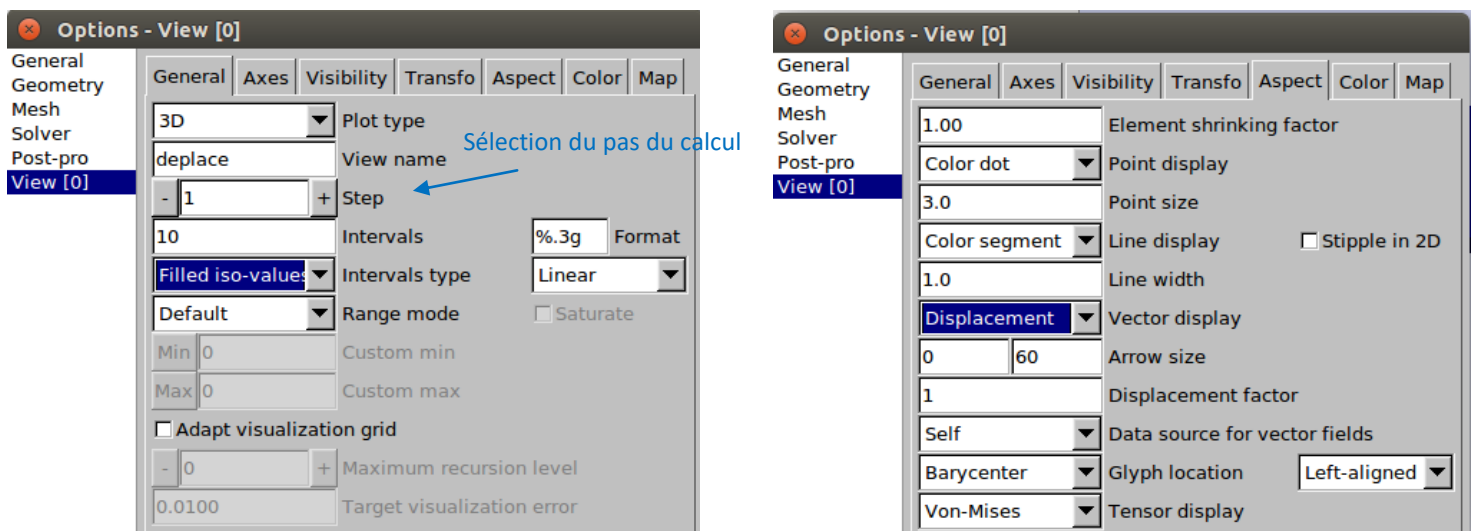
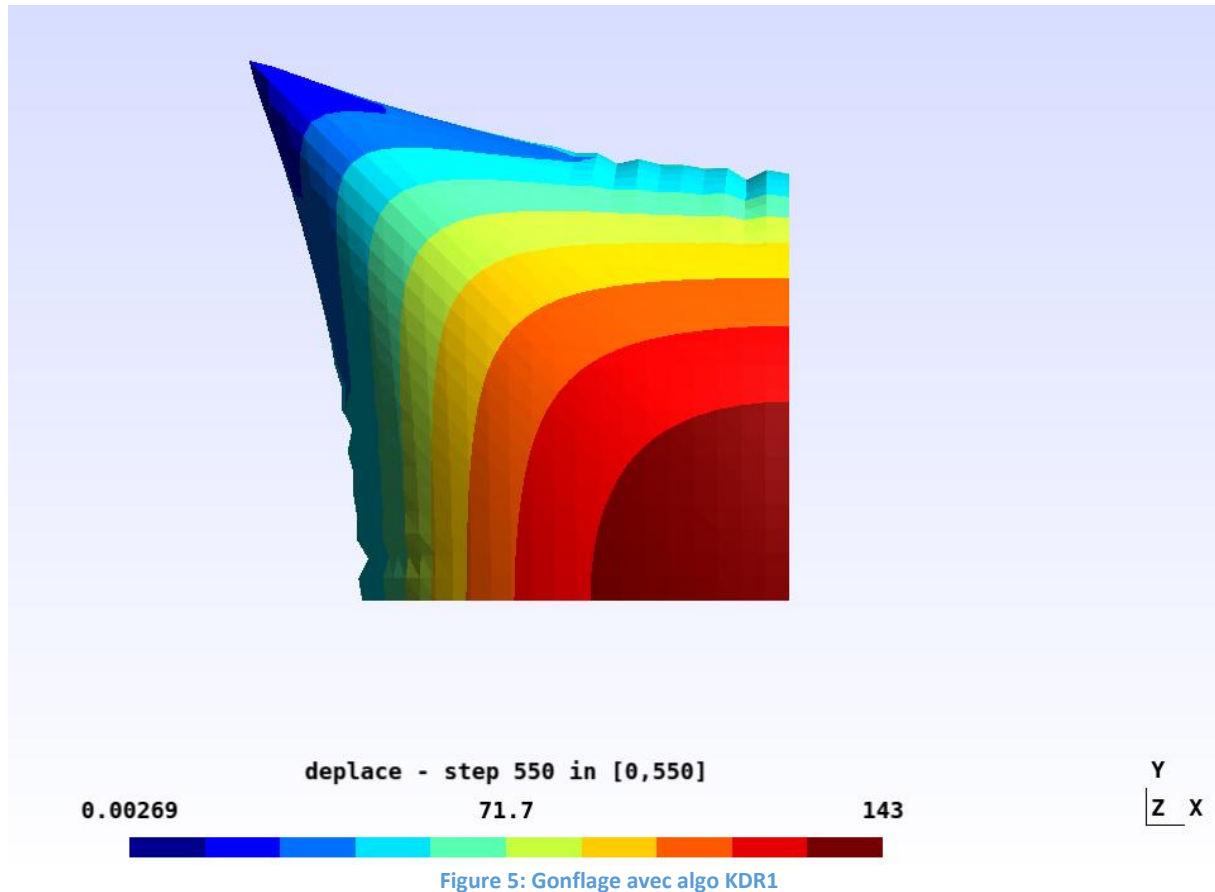


Figure 4: Réglage option



En faisant varier les pas il est possible d'observer le gonflage du coussin. On observe bien le comportement d'un coussin qui est gonflé, la forme observée est correcte. Ce résultat est donné par l'algorithme KDR1 (fichier en annexe).

III) Analyse des différents amortissements

Nous allons analyser les différents fichiers .algo pour observer la conséquence sur la visualisation mais aussi en temps de calcul.

Pour cela il faut exécuter les différents fichiers.algo un à un dans la mise en donnée du fichier.info. Puis garder les différentes déformées mais aussi les fichiers.cpu qui possèdent le temps du calcul.

a) Amortissement cinétique

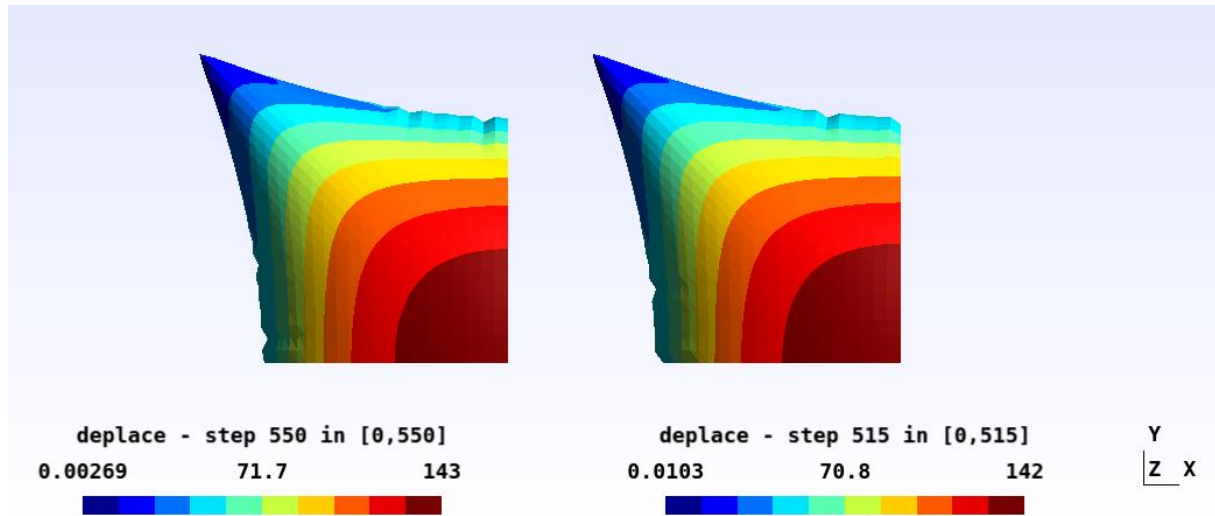


Figure 6: Algo KDR1 (gauche) - KDR (droite)

Les résultats sont à peu près identiques cependant pour celui de gauche on observe plus de plis. Le gonflage se fait très rapidement pour les deux cas puis celui-ci se stabilise. D'ailleurs on remarque que l'amortissement de l'algo KDR est légèrement plus rapide, celui-ci converge plus vite en 518 itérations alors que l'autre en 555 itérations. Cela est dû aux paramètres utilisés, un lambda plus faible permet d'accélérer le temps de calcul. Notons également que le choix d'un calcul de masse fictive à chaque itération (KDR1) ou seulement en début du calcul (KDR), a des conséquences importantes sur les temps de calcul.

b) Amortissement visqueux

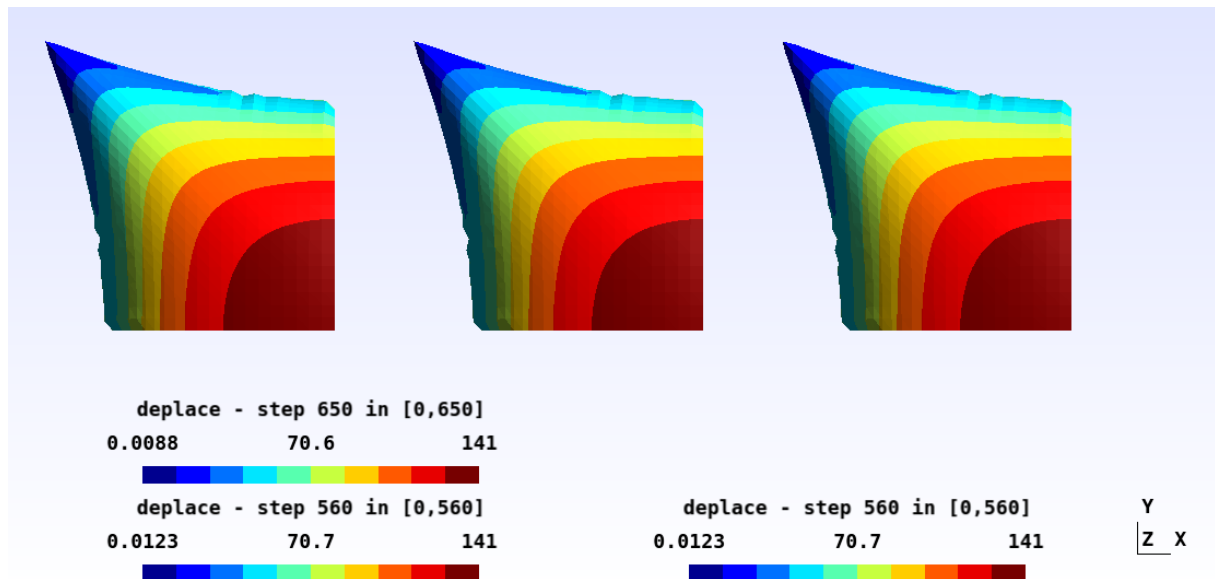


Figure 7: Algo mdDr (gauche) - oDR (milieu) - mDR (droite)

On observe que les résultats sont à peu près identiques. Dans chaque cas le gonflage se fait progressivement et non rapidement mais tout en étant stable. On remarque que l'amortissement de l'algorithme mdDR et oDR (convergence en 561 itérations) est plus rapide que celui de l'algorithme mDR (convergence en 654 itérations). Cette différence de rapidité s'explique essentiellement sur le lambda car pour le mDR celui-ci est légèrement plus grand. Cependant en comparant spécifiquement les paramètres de l'algorithme de relaxation dans les fichiers mdDR et oDR parfois les paramètres ne sont pas les mêmes mais n'influe pas sur le nombre d'itérations pour la convergence. Finalement on constate que c'est essentiellement le paramètre lambda qui influe.

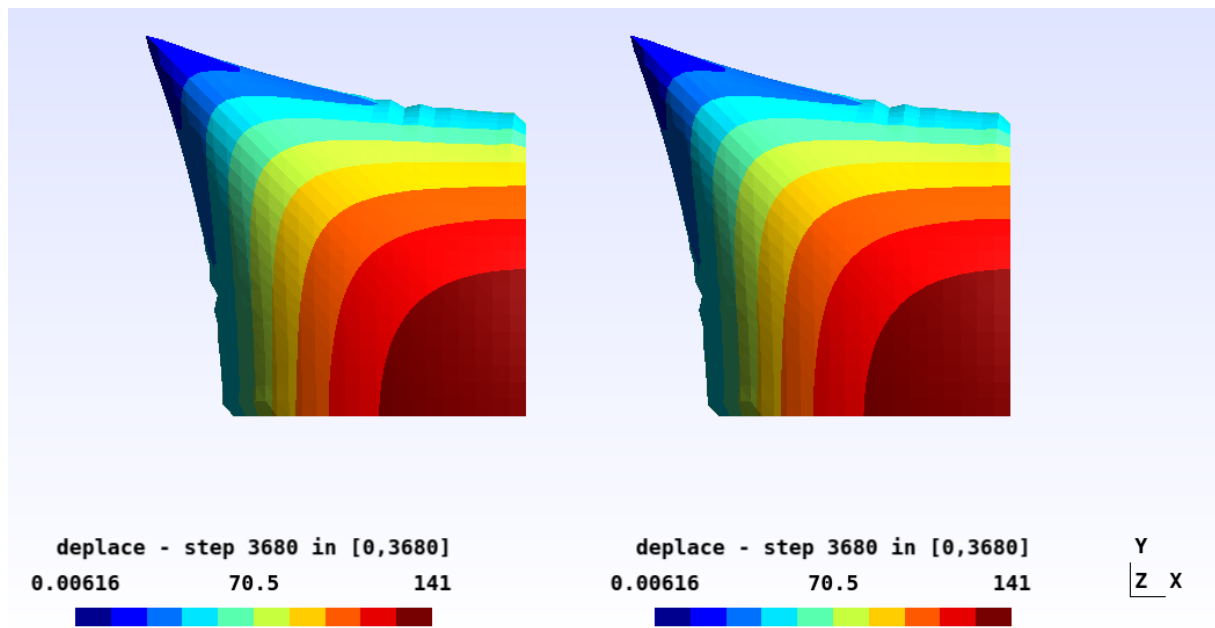


Figure 8: Algo uqDR (gauche) - pqDR (droite)

Sur la figure 8, ces 2 calculs convergent lentement (3685 itérations). D'ailleurs on remarque lors de la visualisation un gonflage très progressif avec très peu de pli donc très stable. Mais le temps de calcul est très important puisque le lambda est plus important que sur les 3 algorithmes précédents.

IV) Conclusion

On remarque que l'amortissement cinétique est plus rapide qu'un amortissement visqueux à conditions de prendre de bons paramètres. De plus pour les deux cas, lambda a un impact conséquent sur la rapidité du calcul mais aussi sur la stabilité du calcul. Il est aussi important de savoir que la rapidité du calcul est surtout liée aux paramètres du calcul mais aussi sur ce qu'on demande en sortie de calcul. En effet, on remarque pour chacun des fichiers.cpu au moins 15% du temps total en sortie de calcul ce qui n'est pas négligeable.

Le lambda est un paramètre d'ajustement permettant de pondérer la masse. Donc lambda est en grande partie responsable de la stabilité du processus d'amortissement. Puisque celui-ci va augmenter la valeur de la masse donc améliorer la stabilité. A noter que plus lambda est grand, plus le calcul est stable, mais à l'inverse plus l'amortissement est faible d'où une convergence vers la solution stable plus longue. Il s'agit donc de trouver un compromis.

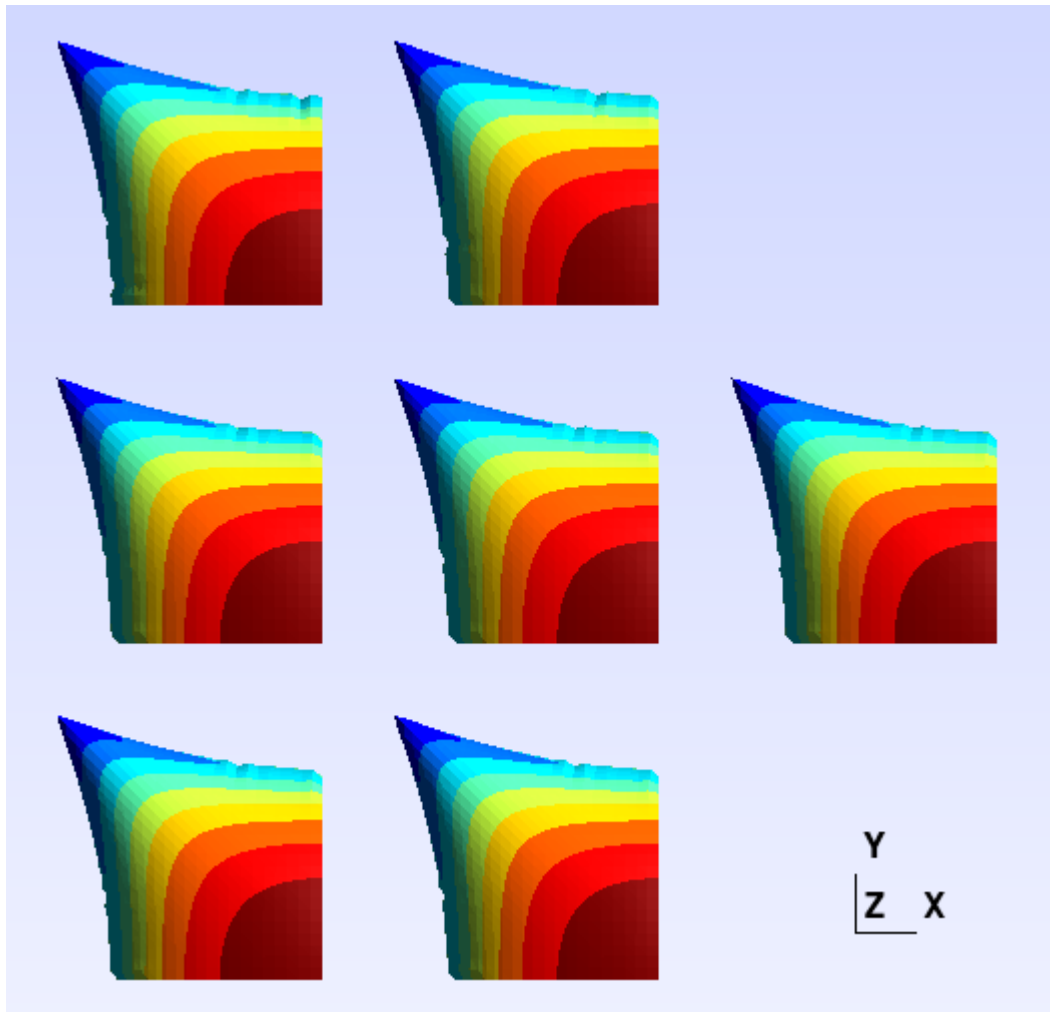


Figure 9: Les différents résultats

Dossier « Inflation-tests » :

- TL_25x25.her (maillage du coussin)
- Inflation_TL1 (mise en donnée du calcul)
- ⇒ KDR.algo | KDR1.algo (amortissement cinétique)
- ⇒ mDR.algo | mdDR.algo | oDR.algo | uqDR.algo | pqDR.algo (amortissement visqueux)
- Inflation_Tl1.Cvisu (sortie du calcul)
- Sous_dossier :
- ⇒ « résultat » : possède les différentes simulations
- ⇒ « temps » : possède les différents temps de calcul