

CV5 – Comment modéliser un ressort cylindrique de compression paramétré – Partie 2 – Optimisation

📁 **cv5**

👤 - 🕒 14h00

Suite de l'article consacré à la modélisation du ressort. Dans l'article précédent, nous avons vu qu'il n'était pas possible d'imposer une longueur à l'Hélice construite avec une loi de type S. On peut par contre lui imposer le nombre de spires mais combien en faut il pour que cette partie fasse précisément la hauteur souhaitée (ici 1/4 de la hauteur libre du ressort).

[\(Voir l'article précédent\)](#)

Je vais vous montrer comment ne pas tâtonner pour trouver le bon nombre de spires en utilisant un...

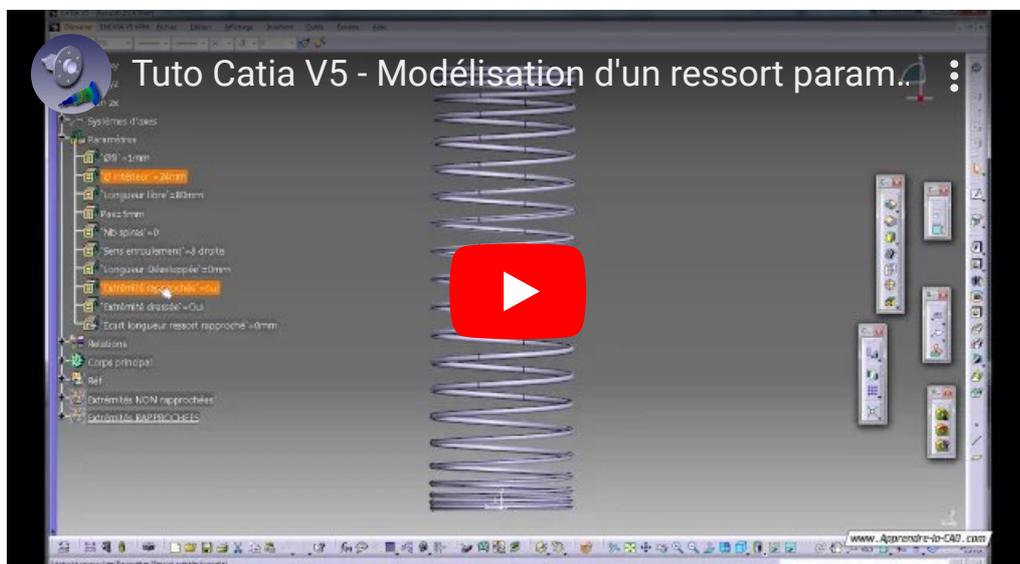
Sommaire [\[Cacher\]](#)

1 [Algorithme d'optimisation afin d'obtenir la longueur souhaitée du ressort.](#)

1.0.1 [Le problème est posé.](#)

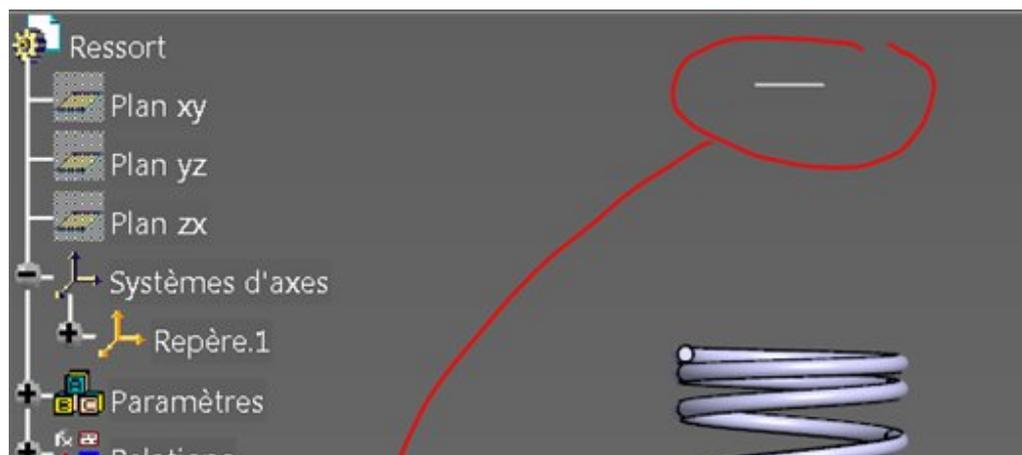
- 1.0.2 Des éléments pour l'optimisation
- 1.0.3 L'optimisation
- 1.0.4 Automatiser la chose avec l'outil "Réactions"
- 1.0.5 Conclusion:

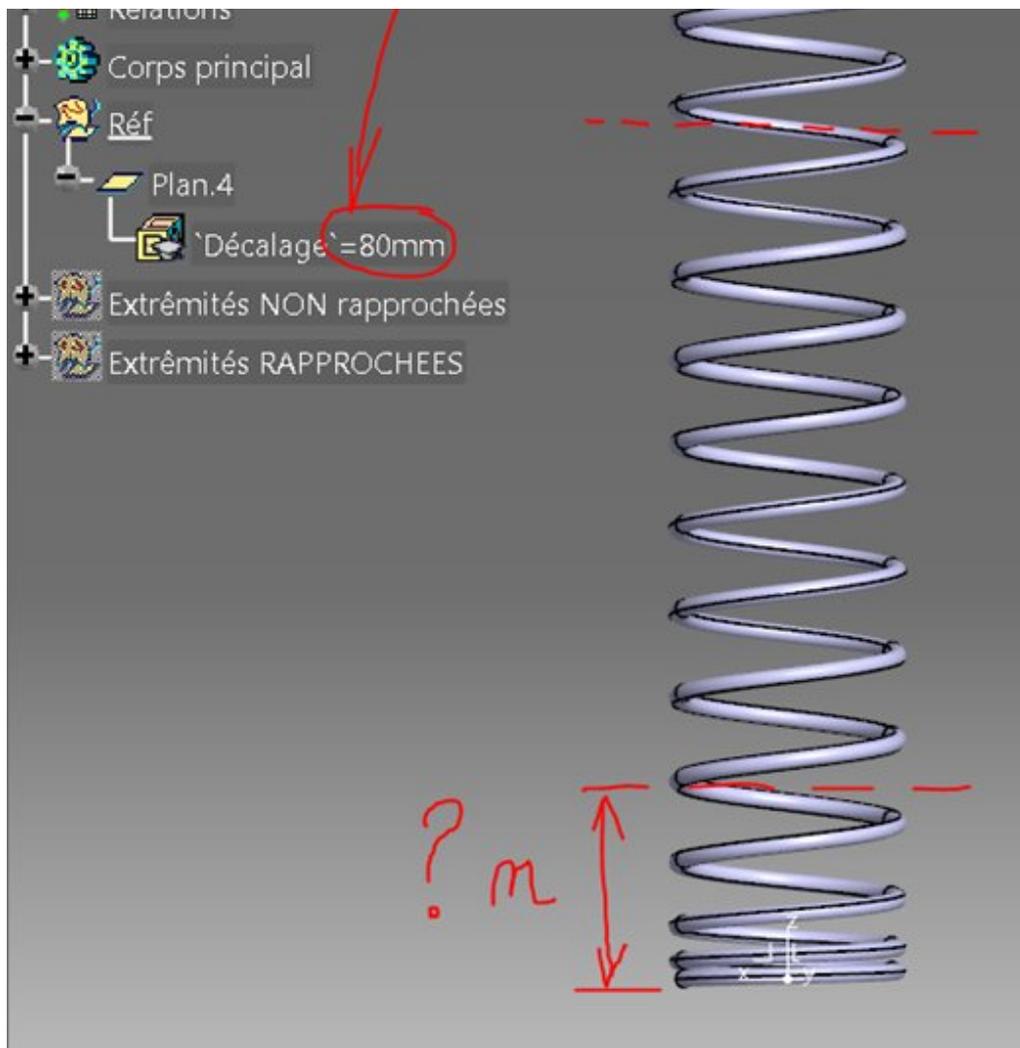
Algorithme d'optimisation afin d'obtenir la longueur souhaitée du ressort.



Le problème est posé.

Nous récupérons le ressort de l'article précédent. Nous avons vu qu'il était trop court (voir l'image ci-dessous).





Il faut donc trouver la juste valeur de n pour que la longueur totale du ressort fasse 80 mm (la longueur libre imposée).

Je rappelle que le ressort est fait en 3 parties avec une partie centrale à pas constant d'une longueur égale à la moitié de la longueur libre et deux extrémités à pas variable (loi de type S) dont la longueur doit donc faire $1/4$ de la longueur libre totale.

Des éléments pour l'optimisation

Avant de créer la routine d'optimisation, il faut un peu réfléchir sur la stratégie à adopter.

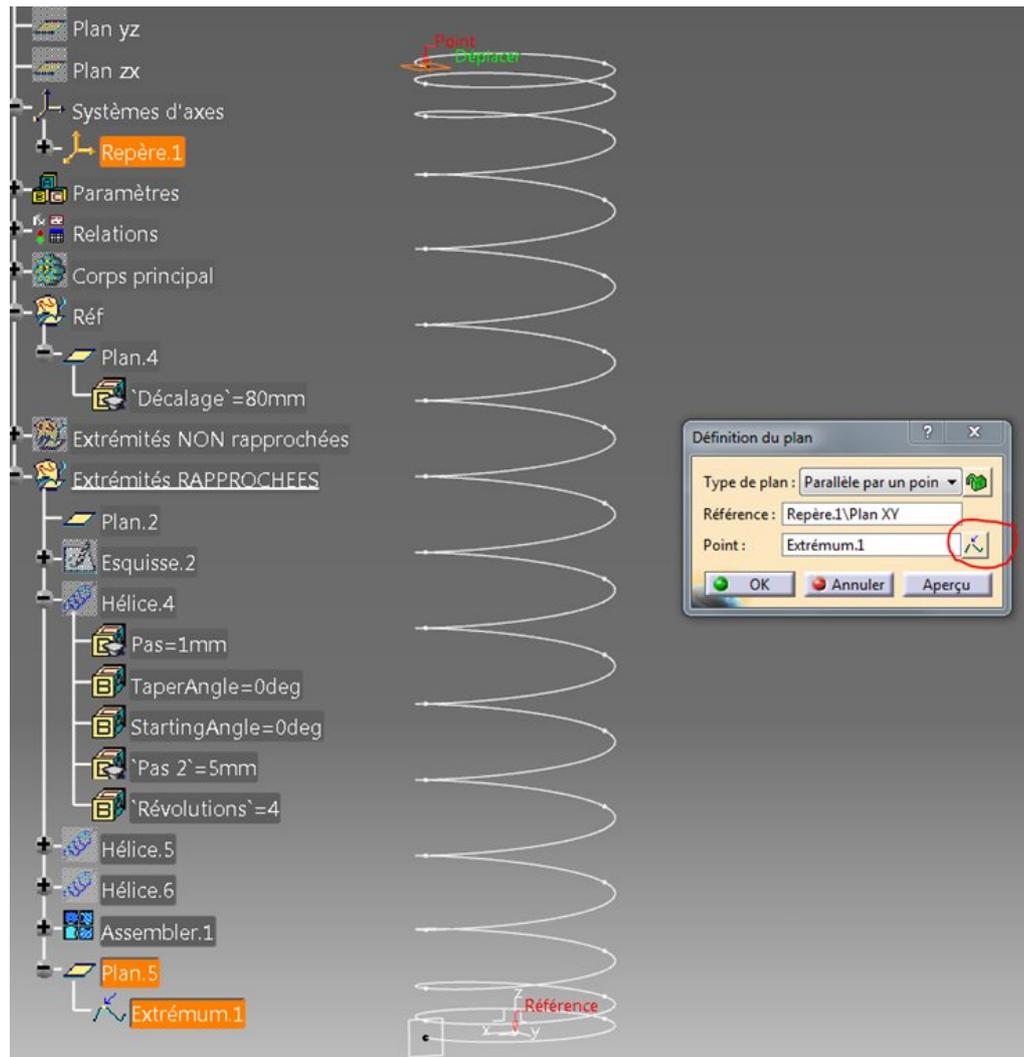
La cible, vous l'avez compris est d'avoir un ressort dont la face supérieure atteigne le "plan.4". Or il existe un écart entre les deux. Nous allons alors pouvoir dire que l'on souhaite:

- soit minimiser la distance entre le dessus du ressort et le "plan.4"

- soit donner comme contrainte quelque chose comme “la hauteur du ressort doit atteindre la valeur du paramètre ‘longueur libre’ ”

Dans les deux cas, il nous faut quelque chose que nous n’avons pas:
La position du dessus du ressort.

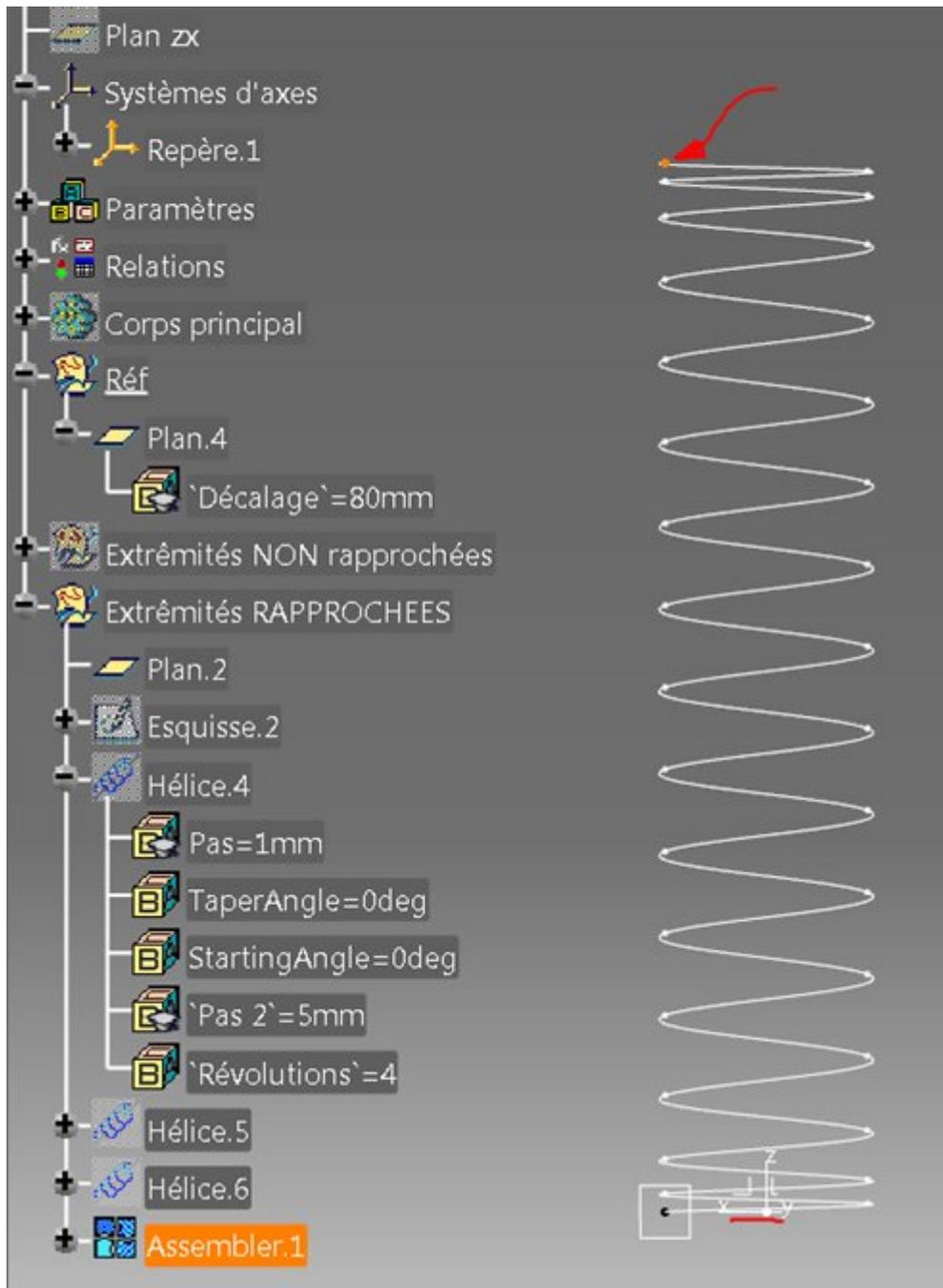
Nous allons donc tout de suite créer un plan parallèle à XY (toujours celui du trièdre) passant par le point extrémité de la courbe guide de ce ressort.



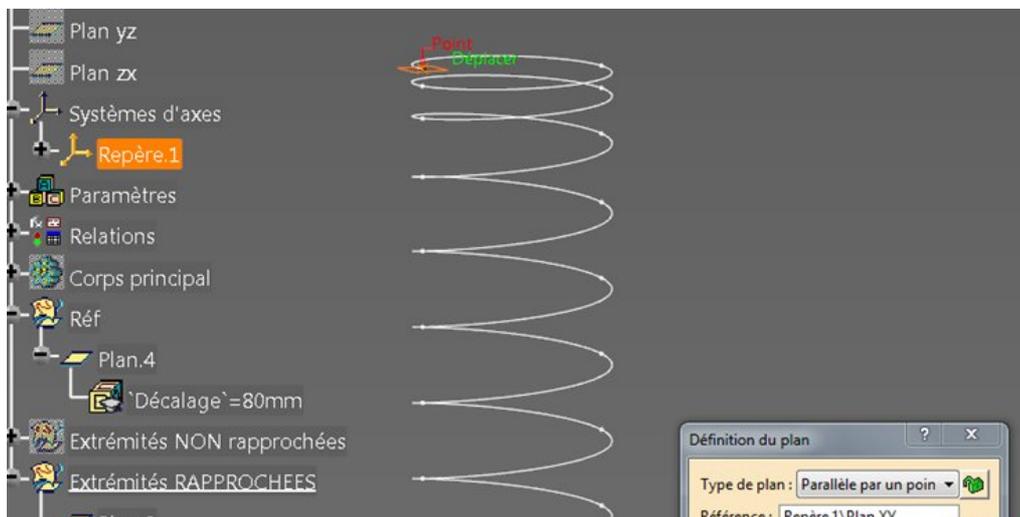
On ajoute cela au set géométrique “Extrémité RAPPROCHÉES”.
Plan parallèle par un point. Pour le point, on fait un « bouton droit » dans le champ à remplir et on choisit “Créer l’extrémum”

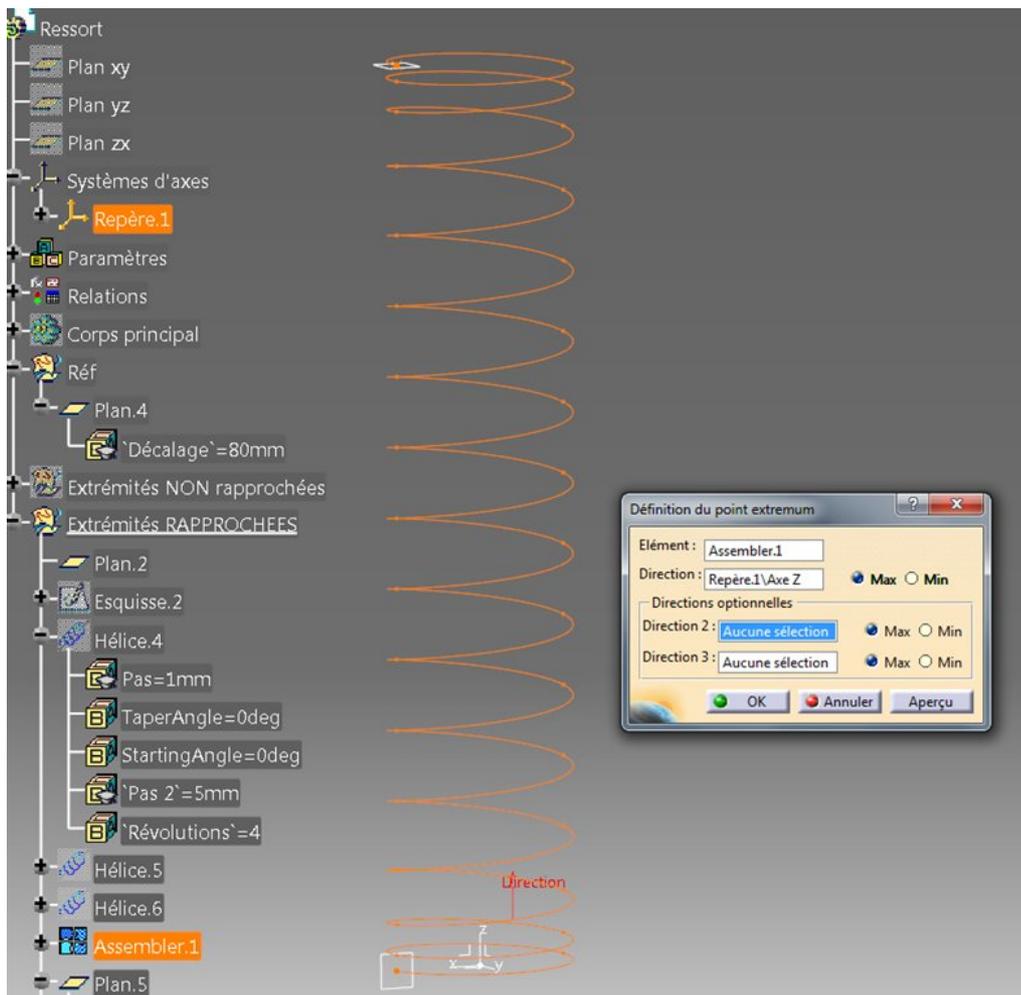
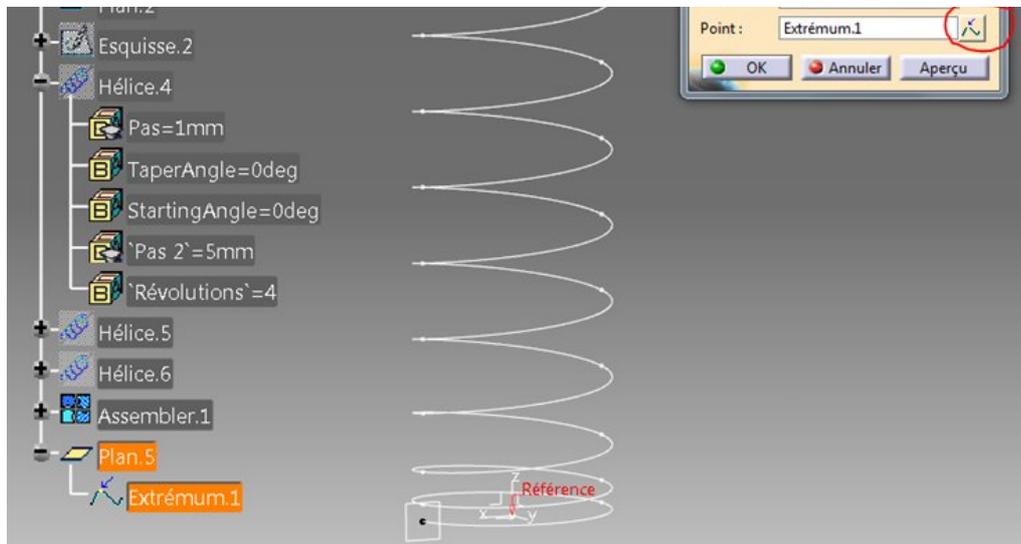
On obtiendra alors cela au final





avec comme point de référence (en agrégation) l'extrémum de la courbe dans la direction Z max du trièdre.





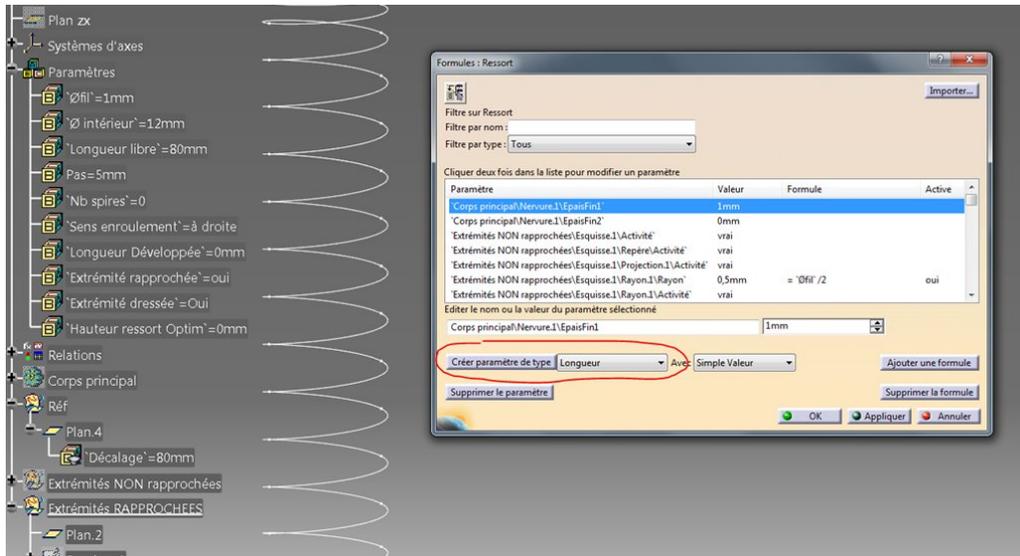
Nous avons le plan passant par le sommet de l'hélice.

Nous allons alors maintenant créer un paramètre de type longueur qui mesurera en permanence la distance entre ce plan (plan.5) et le plan.4

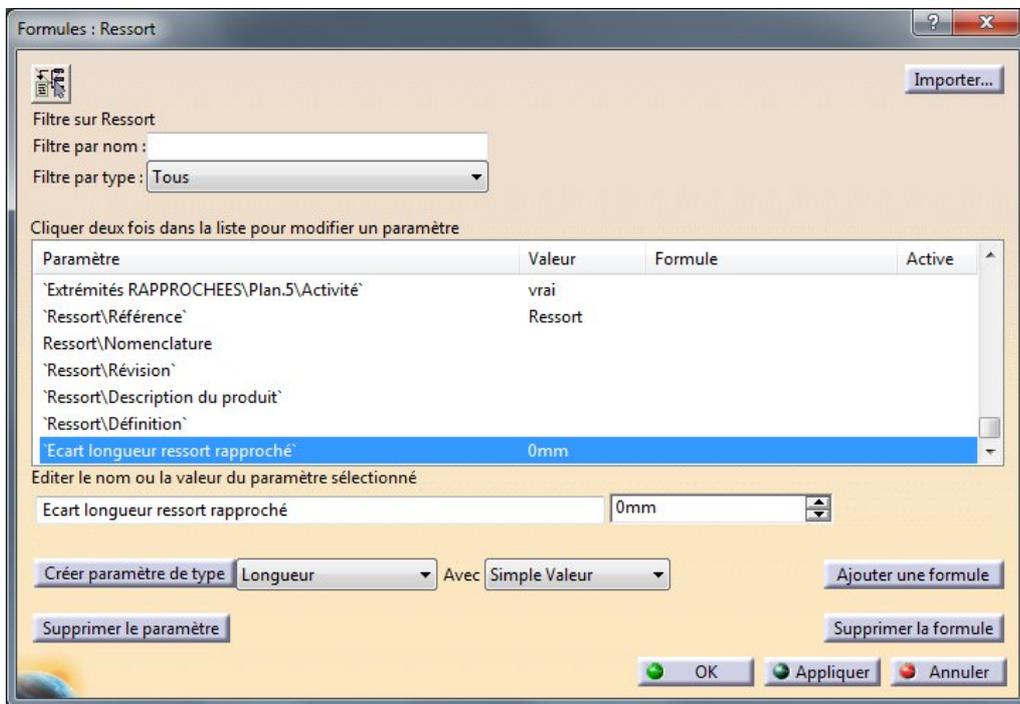
Créer un paramètre de type longueur.

On le renomme



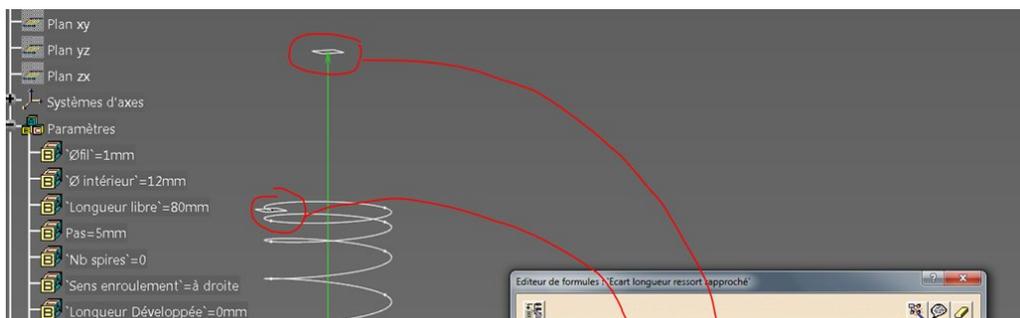


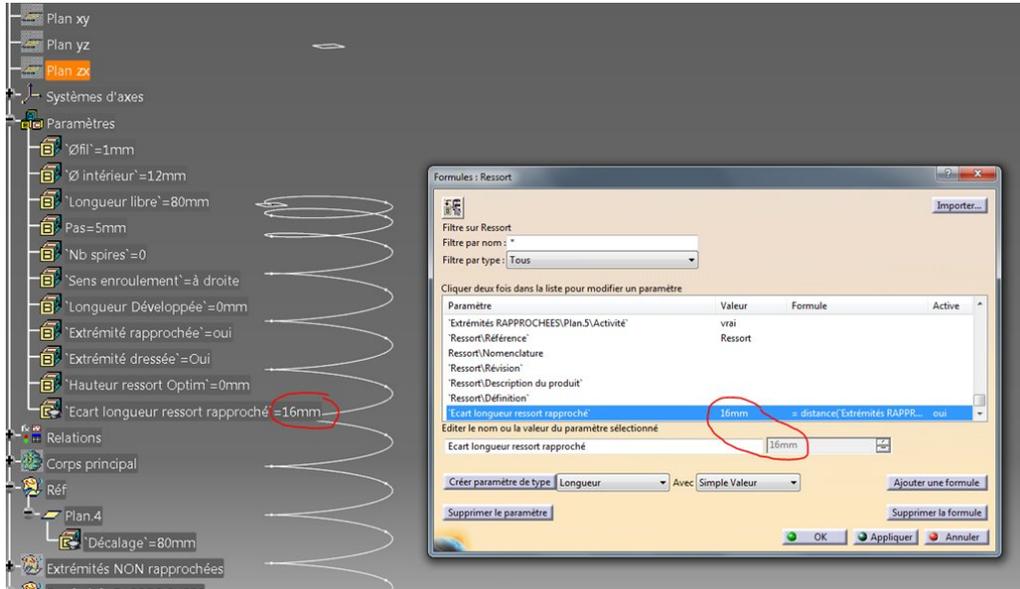
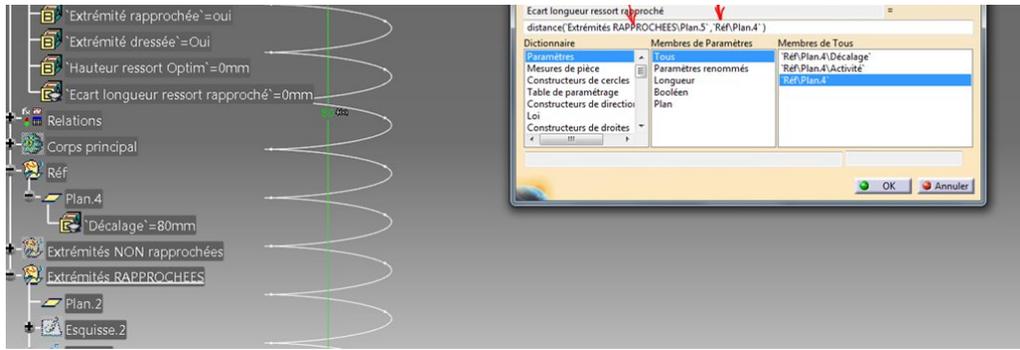
et on ajoute une formule



Le mot clé distance se trouve dans « mesure » (première colonne. Ne pas confondre avec « Mesure de pièce) puis on double clic sur le premier plan puis sur le deuxième (cela se remplit tout seul de chaque côté de la virgule entre les parenthèses)..

On a alors la valeur souhaitée (ici 16 mm entre les deux plans).



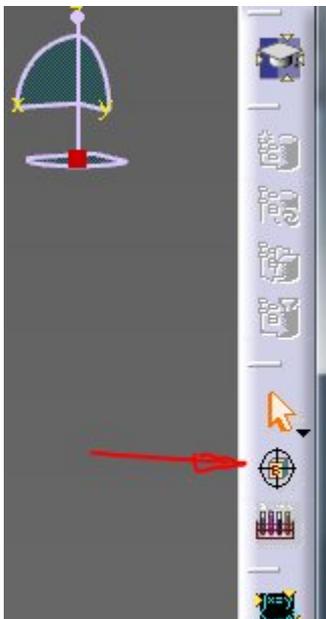


Nous allons maintenant faire en sorte que cet écart fasse 0 mm.

L'optimisation

Allons dans l'atelier Product Engineering Optimizer et prenons l'outil d'optimisation.

Lançons la chose. Nous obtenons ceci.





Dans cet onglet (« problème ») nous allons remplir certains champs en partant du haut.

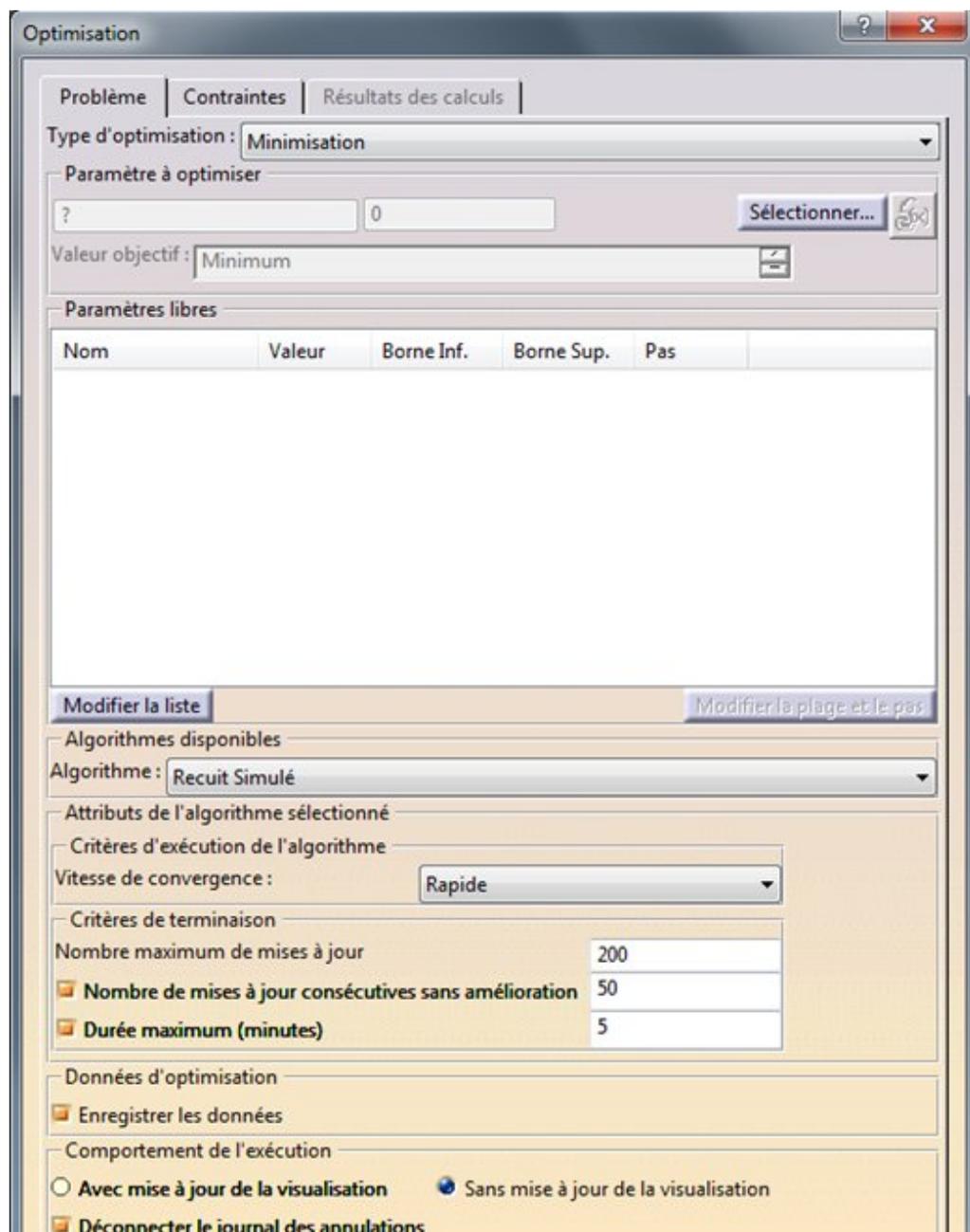
Choisissons ici une “minimisation” puisque nous avons défini notre stratégie comme étant la minimisation de l’écart (distance) entre les deux plans.

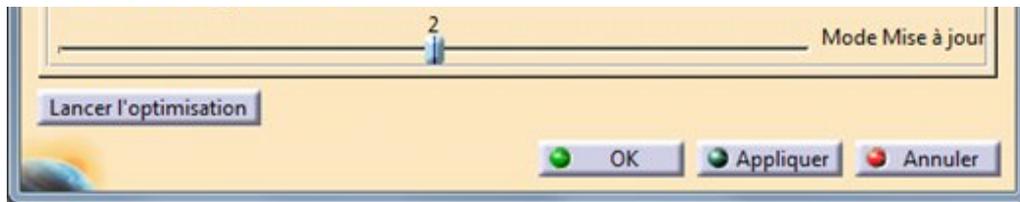
Paramètre à optimiser ... Sélectionner.

A votre avis ?

Oui, c’est le paramètre utilisateur que nous avons créé juste avant (la mesure de distance).

Plutôt que de le sélectionner dans la liste, faites le dans l’arbre c’est plus facile. Puis Ok !



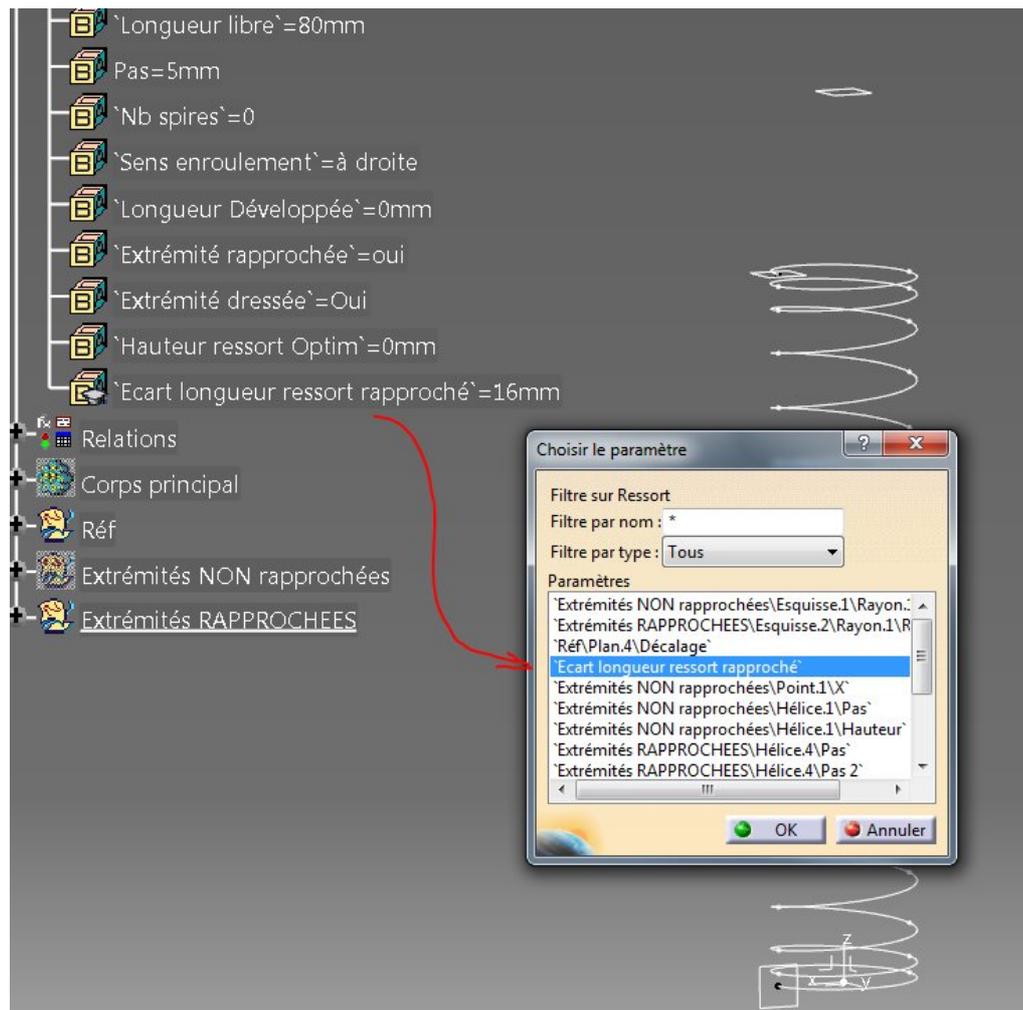


C'est déjà bien. C'est ce que l'on veut minimiser.

Oui mais en jouant sur quels potentiomètres (sur quelles variables) ?

Ce sont les paramètres libres.

Cliquons sur modifier la liste.

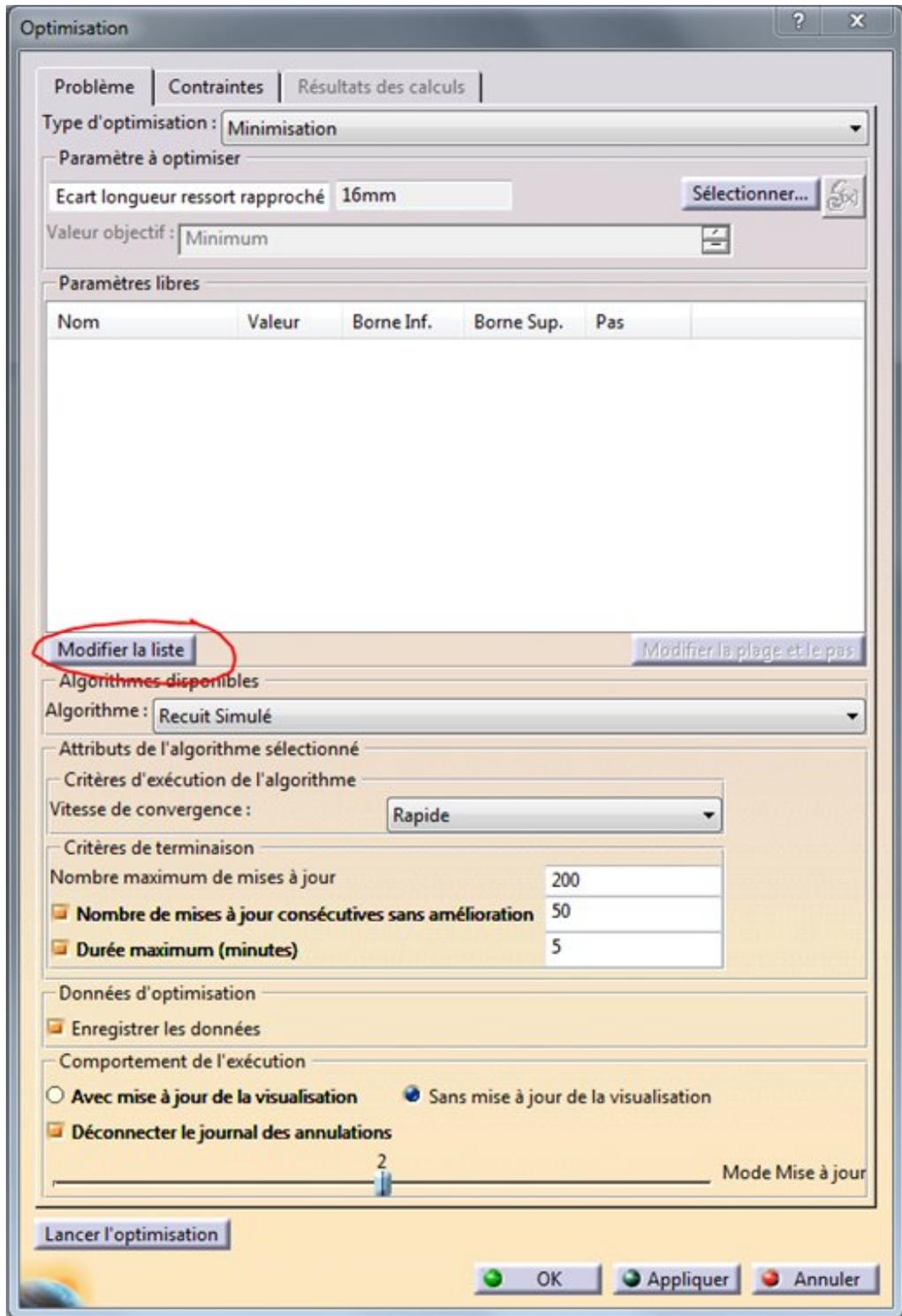


Sur quoi doit-on jouer ?... Sur le nombre de spires de l'extrémité du ressort bien sûr !

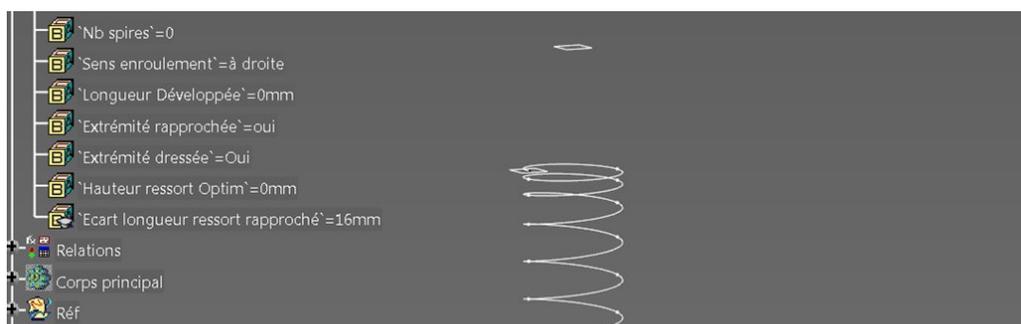
De la même façon, il suffit de cliquer dans l'arbre sur le paramètre « révolutions » de Hélice.4 pour qu'il soit mis en évidence dans la liste.

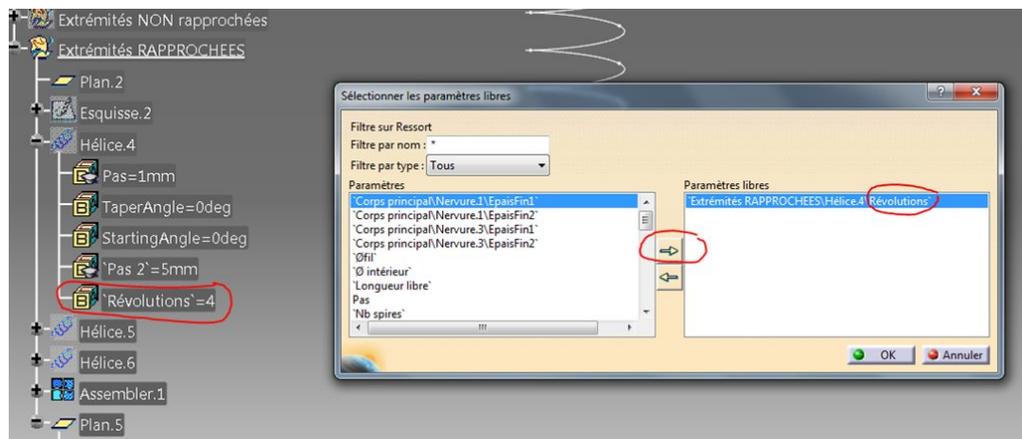
Il faut ensuite le passer à droite avec la flèche.

Rappel: Par construction, l'autre extrémité à pas variable possède le même nombre de spires. Ils sont liés par une formule (égalité). On ne pourrait de toute façon pas l'ajouter à la liste.



Après un ok, on a un élément dans la liste. Il n'en faut pas plus ici. Si l'on sélectionne cet élément de la liste, on peut alors changer son pas et les bornes.



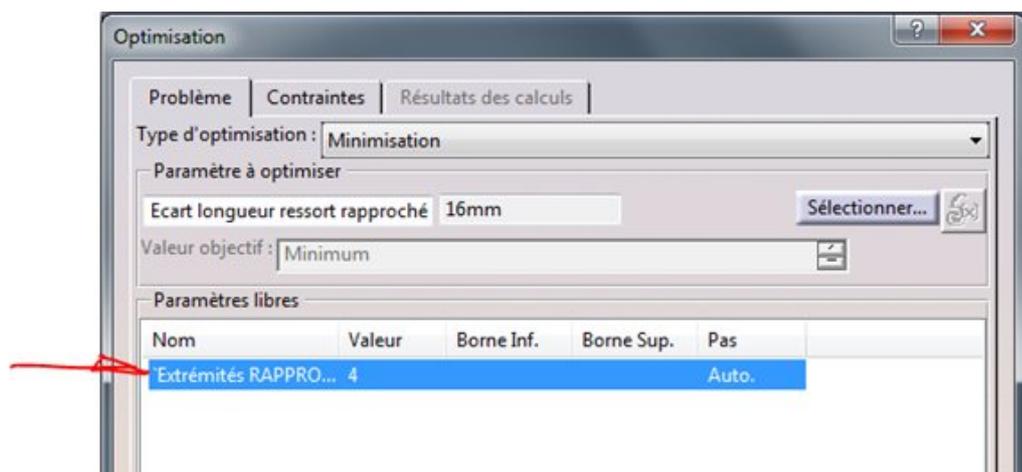


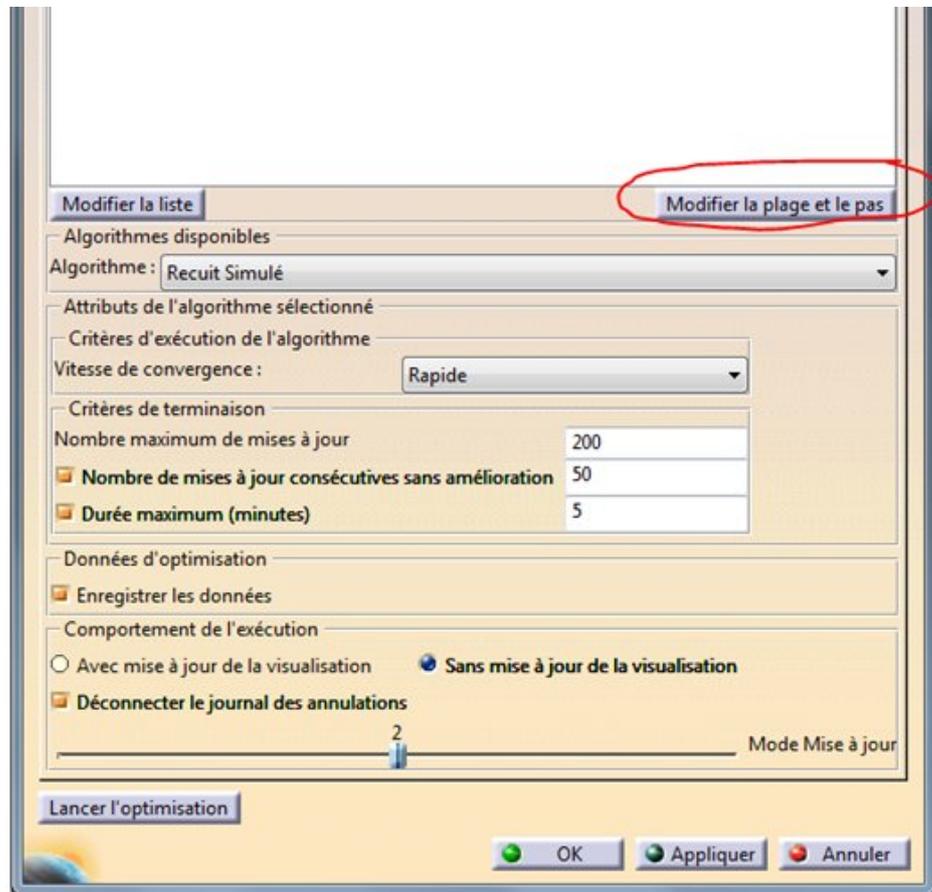
Il est toujours bon de mettre des bornes aux paramètres car l'algorithme va faire varier (trop) largement les valeurs des paramètres libres et cela va conduire rapidement à des échecs de mise à jour. Des coups pour rien qui vous feront perdre du temps finalement.

J'ai choisi ici d'interdire les essais avec des valeurs inférieures à 2 spires. Je n'impose par contre rien d'autre car je ne sais pas dire si avec par exemple un ressort d'un mètre de long avec un pas très petit on n'aurait pas finalement 100 ou 200 spires progressives. Idem pour le pas. Je cherche à me rapprocher le plus possible de la longueur libre imposée. Si je donne un pas de 0.1 spire, je m'approcherai du résultat avec une précision moindre que si je donne un pas de 0.001 spire.

Je laisse donc soin à la machine de faire varier le pas et de s'approcher au mieux du résultat. Cela donne une forte précision (fraction décimale) du nombre de spires. J'accepterai alors des valeurs un peu peaufinées comme 2.1124547 à la place de 2.1 spires.

Mais pour ne pas être trop long, nous choisirons alors de limiter le temps de calcul à 1 minute au lieu de 5.





Pour le reste, je garde ces réglages : Algorithme du recuit simulé avec une convergence rapide.

J'ai réduit la durée maximale de calcul: 1 minute.

Je n'enregistre pas les données car ce n'est pas utile d'avoir après coup une analyse de la convergence dans ce cas.

Je laisse la mise à jour de la visualisation car c'est plus parlant. On ne le voit finalement dynamiquement que sur le filaire (pas sur le solide).

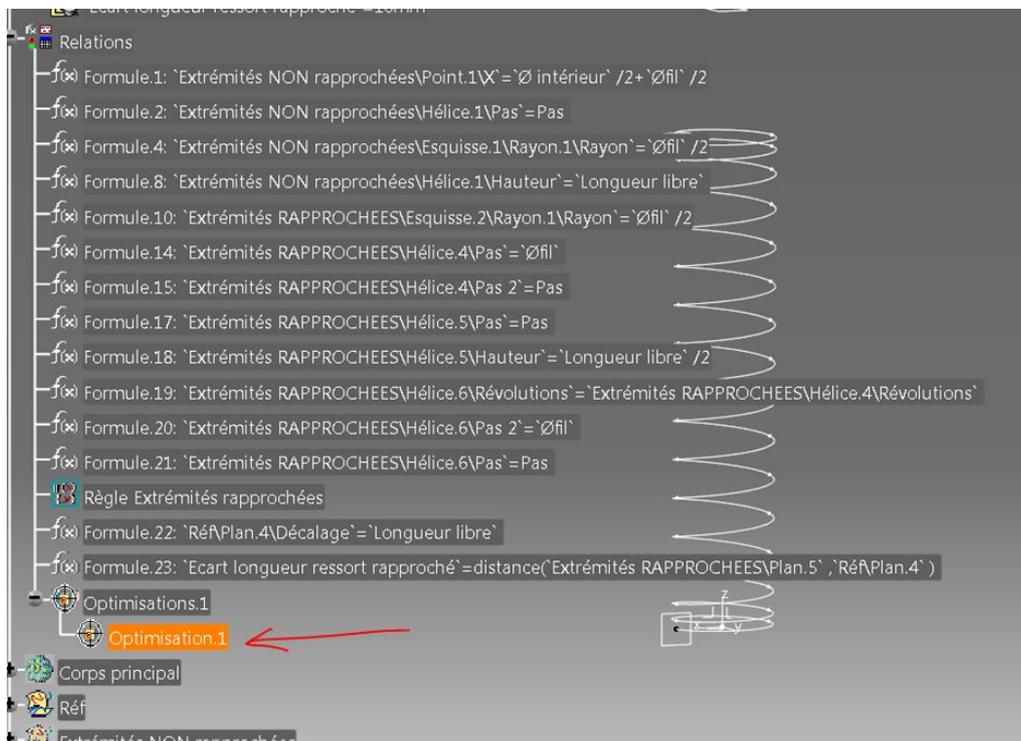
Et je n'oublie pas de faire « Appliquer » ou « Ok » pour que tout soit conservé dans l'arbre.

Oui, si j'annule ou si je lance directement l'optimisation et que cela « plante », j'aurai perdu tous les réglages et il faudra recommencer !

Au fait, pour retrouver cette optimisation, il faut aller chercher dans l'arbre dans Relations. Un double clic pour ré-ouvrir la boîte de dialogue.

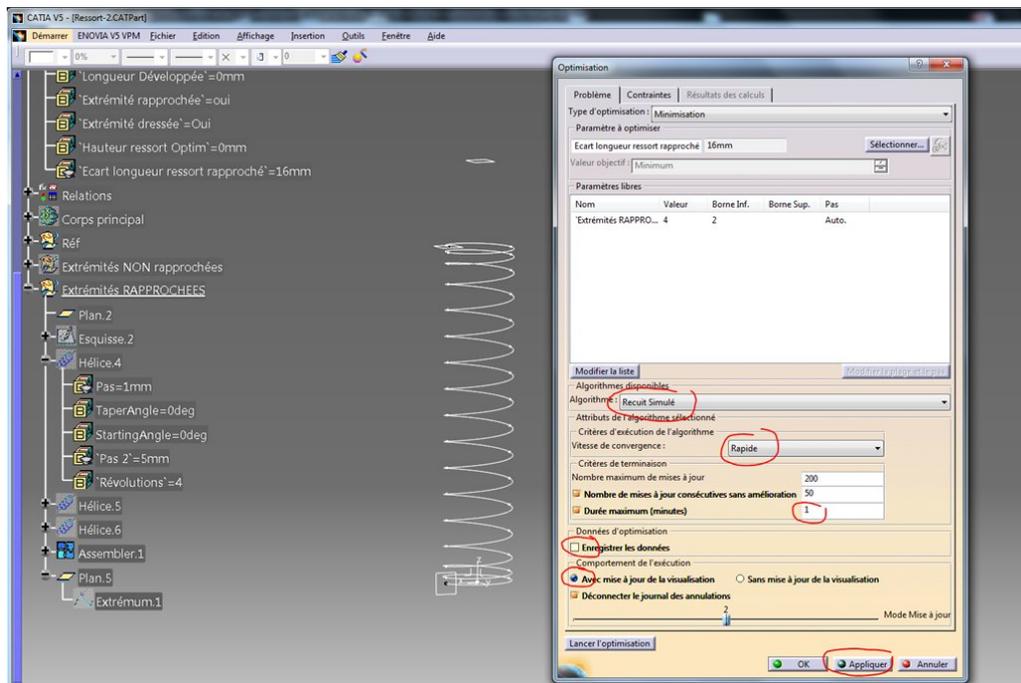
Attention: Si l'on lance l'outil d'optimisation une nouvelle fois, cela créera dans l'arbre une nouvelle Optimisation.





Pensez à sauvegarder d'abord votre modèle ;)

On lance l'optimisation.



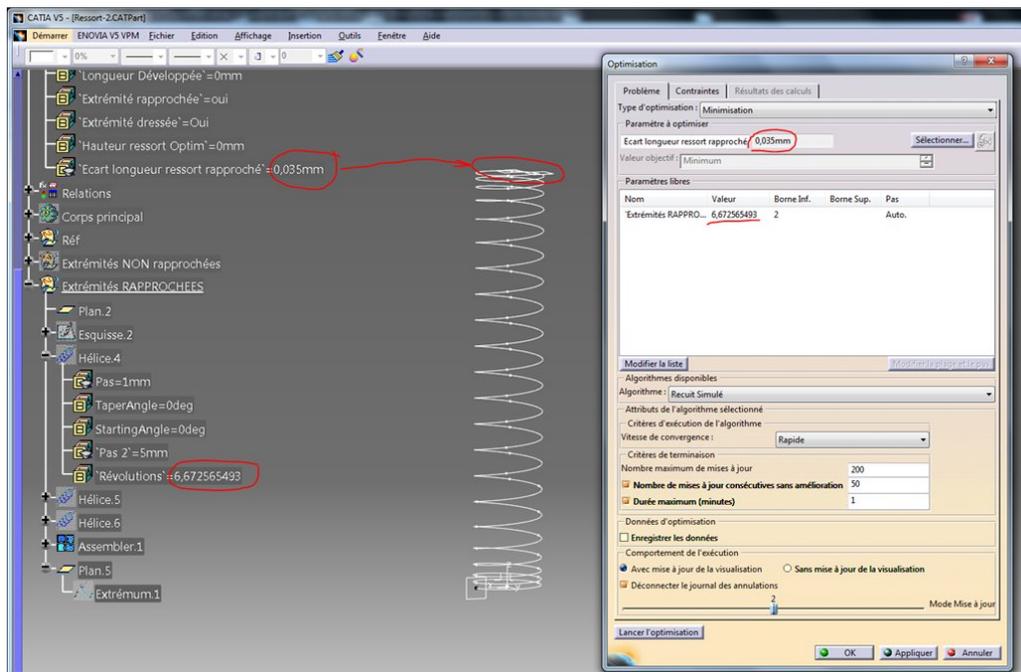
Cela ne prend que quelques secondes et les deux plans se retrouvent très proches l'un de l'autre.

ici 35µm mais dans certains cas (on peut d'ailleurs relancer l'optimisation) on obtient des valeurs inférieures au µm.

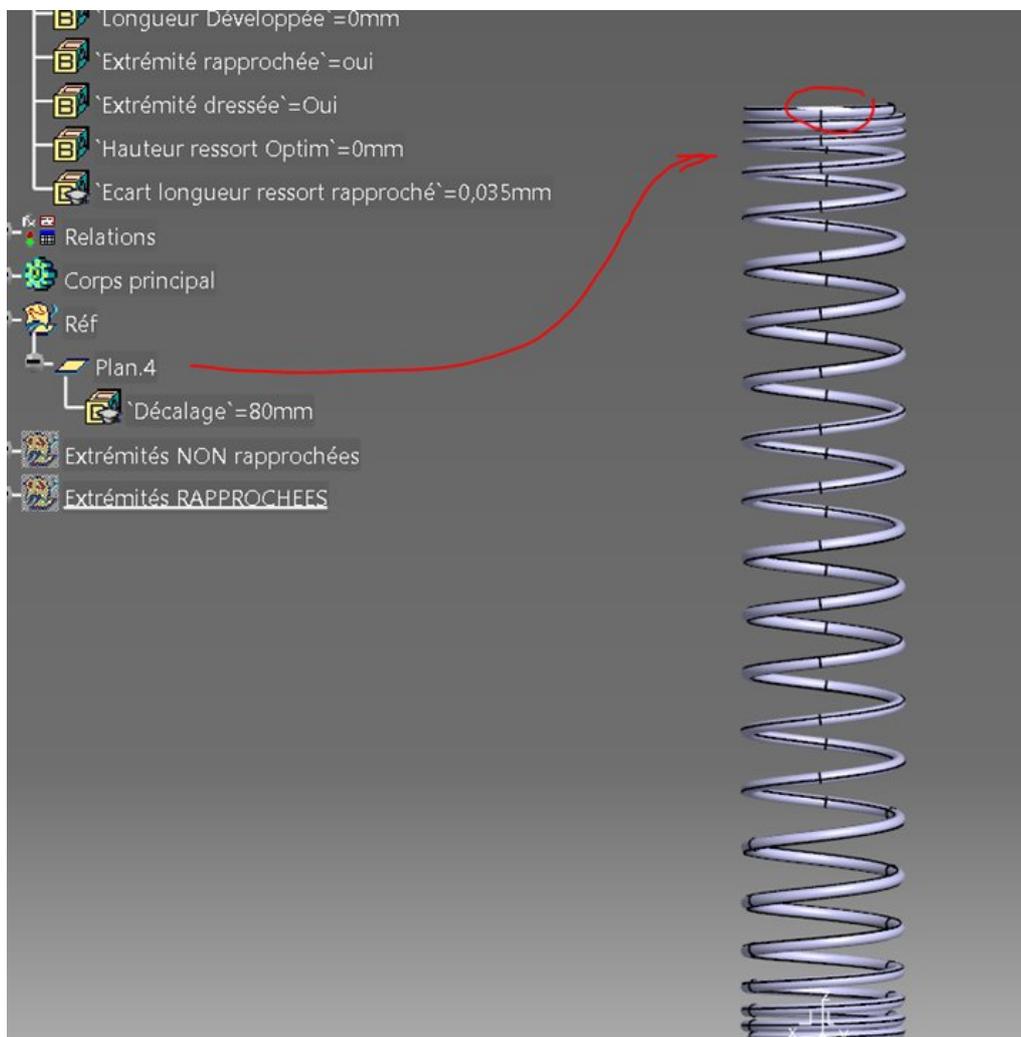
Le nombre de spires idéal serait alors de 6,672565493 ! Vous l'auriez

trouvé à la main ?

En tout cas, notre ressort a la bonne longueur libre.



Cela fonctionne très bien mais il faut penser à le faire à chaque fois que l'on modifie un paramètre en entrée.

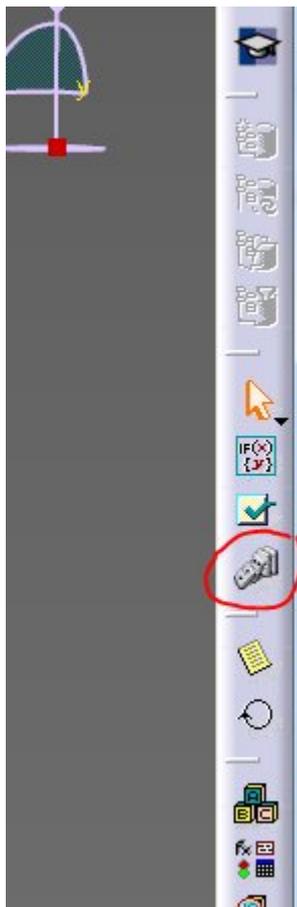


Automatiser la chose avec l'outil "Réactions"

Tout cela c'est bien beau, mais ce qui serait bien, c'est que cela se mette en route lorsque je choisis un ressort de type "extrémités rapprochées" dans l'arbre ou bien lorsque je change un des paramètres dimensionnels.

Oui car je risquerais d'oublier de lancer l'optimisation et cela serait vite astreignant d'aller chercher chaque fois dans l'arbre l'outil d'optimisation.

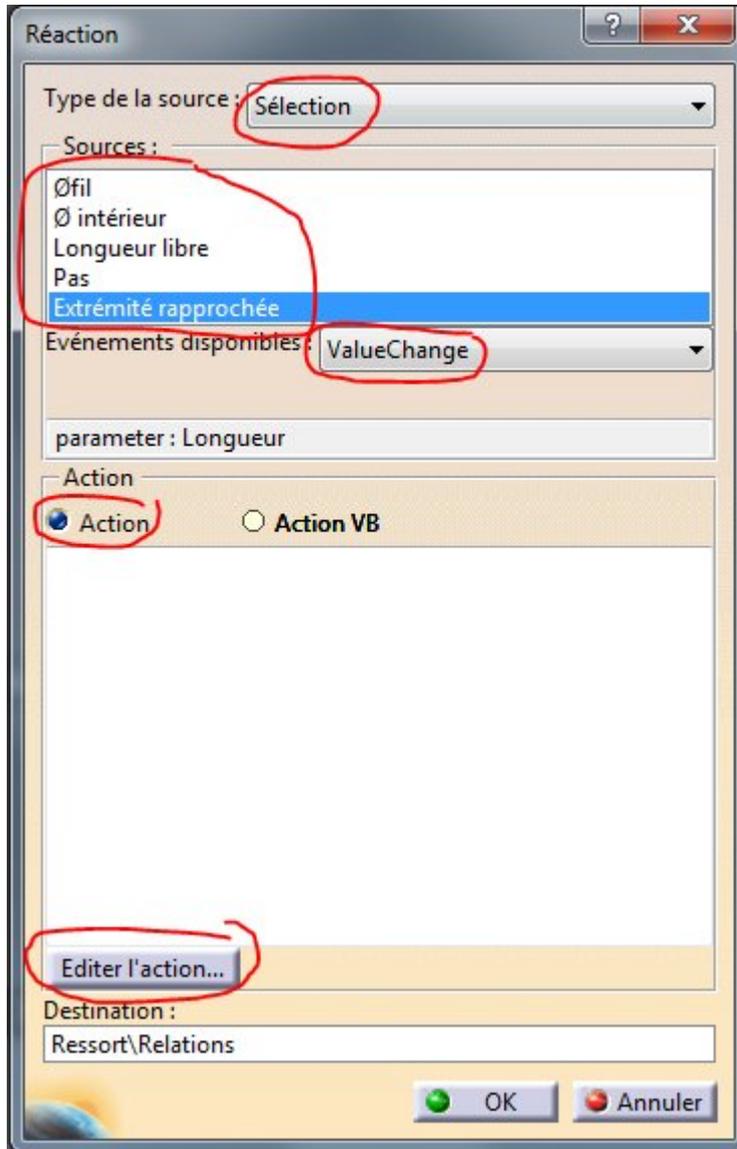
Nous allons faire cela grâce à l'outil "Réactions" de l'atelier Knowledge Advisor.



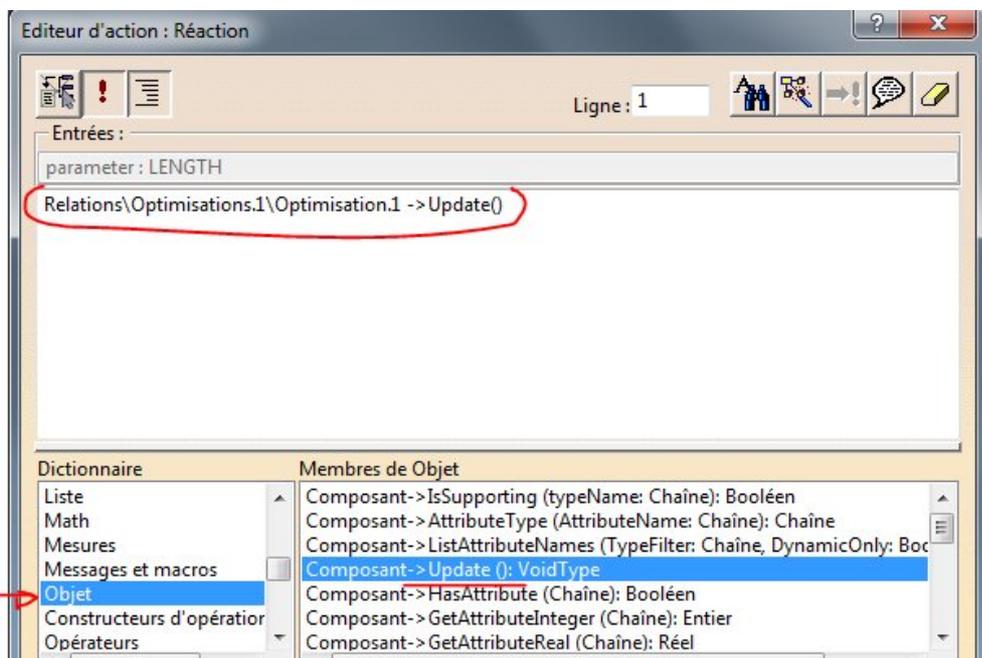
Dans cette nouvelle fenêtre nous allons dire:

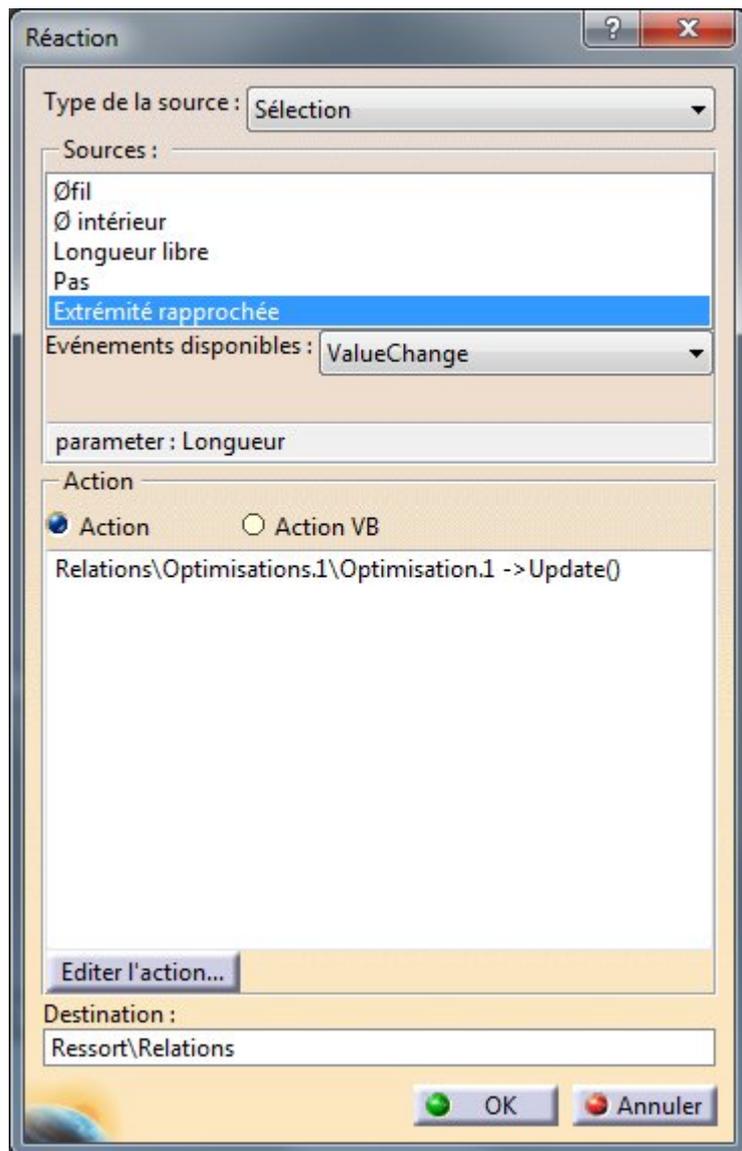
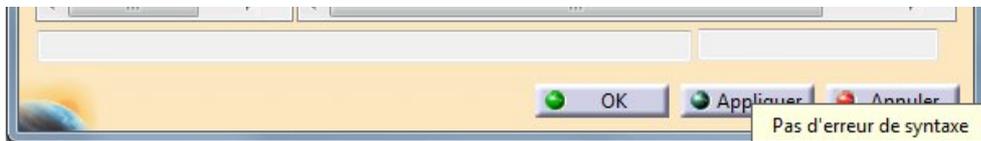
Si l'un des paramètres choisis dans cette liste (il suffit de les désigner dans l'arbre) est modifié (value change)...

Alors l'action suivante survient.



La mise à jour de l'optimisation ! (on clique d'abord sur l'optimisation dans l'arbre puis dans la partie gauche de la liste « Objet » puis « Composant->Update() »)





Et c'est tout. Ok !

On retrouve cela dans l'arbre sous Relations.

Ainsi, à chaque fois que vous modifiez un des paramètres d'entrée, Catia lance l'optimisation et recalcule le nombre de spires des extrémités pour que le ressort à extrémités rapprochées ait la bonne longueur libre.

Cela demande à chaque fois une poignée de secondes et cela fait penser à un plantage de la machine car il n'y a pas d'état d'avancement. Toutefois cela fonctionne très bien (patience)

Conclusion:

Dans cet article, vous avez pu voir une méthode pour atteindre une cible donnée par l'utilisation d'un algorithme d'optimisation (minimisation).

Vous avez également vu comment il était possible de déclencher cette optimisation grâce à un évènement (ici le changement de valeur d'un paramètre utilisateur).

Dans le [prochain article](#) nous verrons comment dresser les extrémités du ressort et comment récupérer la longueur développée.

Si cet article vous a plu, criez le sur les réseaux sociaux et si vous avez des questions, n'hésitez pas à me laisser un commentaire ci-dessous, j'y répondrai.

.

 No Tag

PREVIOUS POST

NEXT POST

No responses yet

Laisser un commentaire

Votre adresse e-mail ne sera pas publiée. Les champs obligatoires sont indiqués avec *

Commentaire *

Nom *

E-mail *

Site web

Enregistrer mon nom, mon e-mail et mon site dans le navigateur pour mon prochain commentaire.

LAISSER UN COMMENTAIRE

Search ...

Search



Articles récents

Rhino #4 – GrassHopper ou Python, sélection d'un élément par son nom

Rhino #2 – l'indispensable manipulateur (Gumball) de Rhinocéros

Rhino #1 – C'est quoi Rhinocéros 3D ?

Rhino #3 – Une cafetière Bialetti sur Rhinocéros – Part 1

Rhino #5 – Une cafetière Bialetti sur Rhinocéros – Part 2

Commentaires récents

Akrim dans CV5-Utilisez-vous les lois ?

PSX59 dans Quel logiciel 3D pour mon modèle?

STEFANOVIC dans Quel logiciel 3D pour mon modèle?

Vince PSX dans CV5 – Comment faire un moletage partiel en 2 étapes par copie optimisée ?

Vince PSX dans Tutoriel débutant – Premier assemblage

Archives

juin 2019

mai 2019

avril 2019

mars 2019

février 2019
janvier 2019
décembre 2018
novembre 2018
octobre 2018
septembre 2018
août 2018
juillet 2018
juin 2018
mai 2018
avril 2018
mars 2018
février 2018
janvier 2018
décembre 2017
novembre 2017
octobre 2017
septembre 2017
août 2017
juillet 2017
juin 2017
mai 2017
avril 2017
mars 2017
février 2017
janvier 2017
décembre 2016
novembre 2016
octobre 2016
septembre 2016
août 2016
juillet 2016
juin 2016
mai 2016

[avril 2016](#)
[mars 2016](#)
[février 2016](#)
[janvier 2016](#)
[décembre 2015](#)
[novembre 2015](#)
[octobre 2015](#)
[septembre 2015](#)
[août 2015](#)
[juillet 2015](#)
[juin 2015](#)

Catégories

[Calcul](#)
[CV5](#)
[Evolve](#)
[Fusion 360](#)
[Non classé](#)
[Python](#)
[Rhino](#)
[Usinage](#)
[Vb.Net](#)
[ZW3D](#)

Méta

[Connexion](#)
[Flux des publications](#)
[Flux des commentaires](#)
[Site de WordPress-FR](#)

