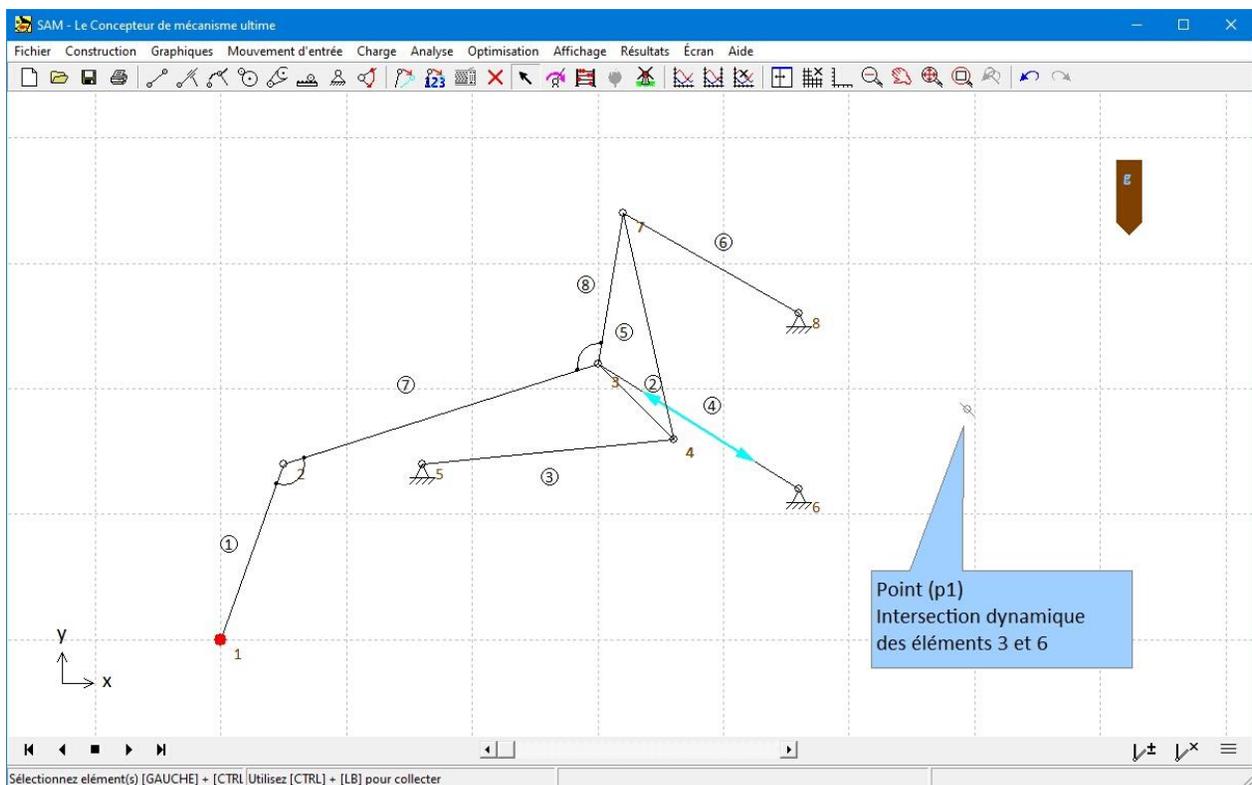


## Quelles nouveautés dans SAM 8.3?

### Points dynamiques

La possibilité de construire des points via l'intersection de lignes et même l'intersection d'axes est devenue dynamique. Lors du déplacement des objets utilisés dans la création d'un point, les coordonnées xy de ce point sont adaptées automatiquement. Cela vaut pour la situation où un objet est déplacé dans la position de référence, mais aussi lors de l'exécution d'une animation.

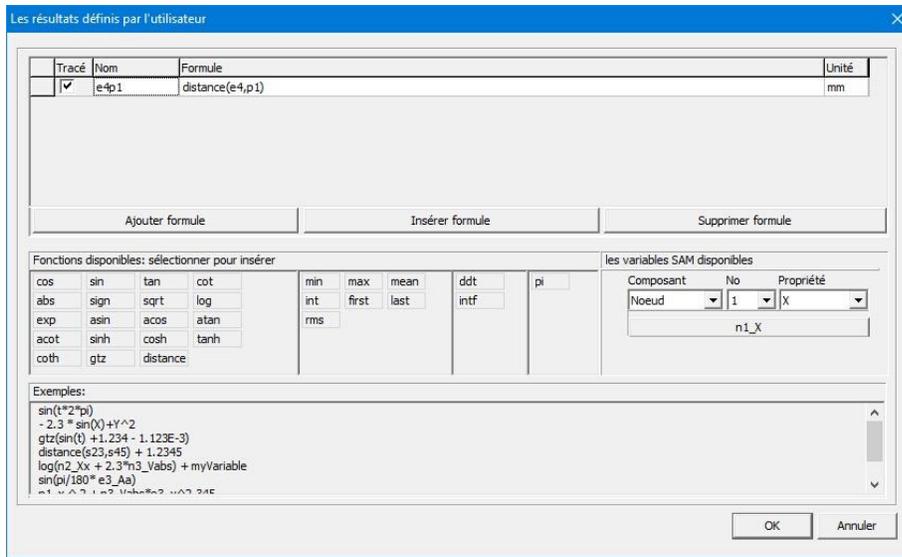
De cette manière, on peut par exemple créer le point p1 comme intersection de la poutre 3 et de la poutre 6, qui constitue le centre de rotation momentané du triangle formé par le coupleur triangulaire. Lorsque le mécanisme est animé et passe d'une position à la suivante, la position du point p1 s'adapte automatiquement.



**Le point p1 est créé comme une intersection des deux lignes (infinies) correspondant aux éléments de poutre 3 et 6. Le point est dynamique et se déplace au cours d'une animation.**

### Distance

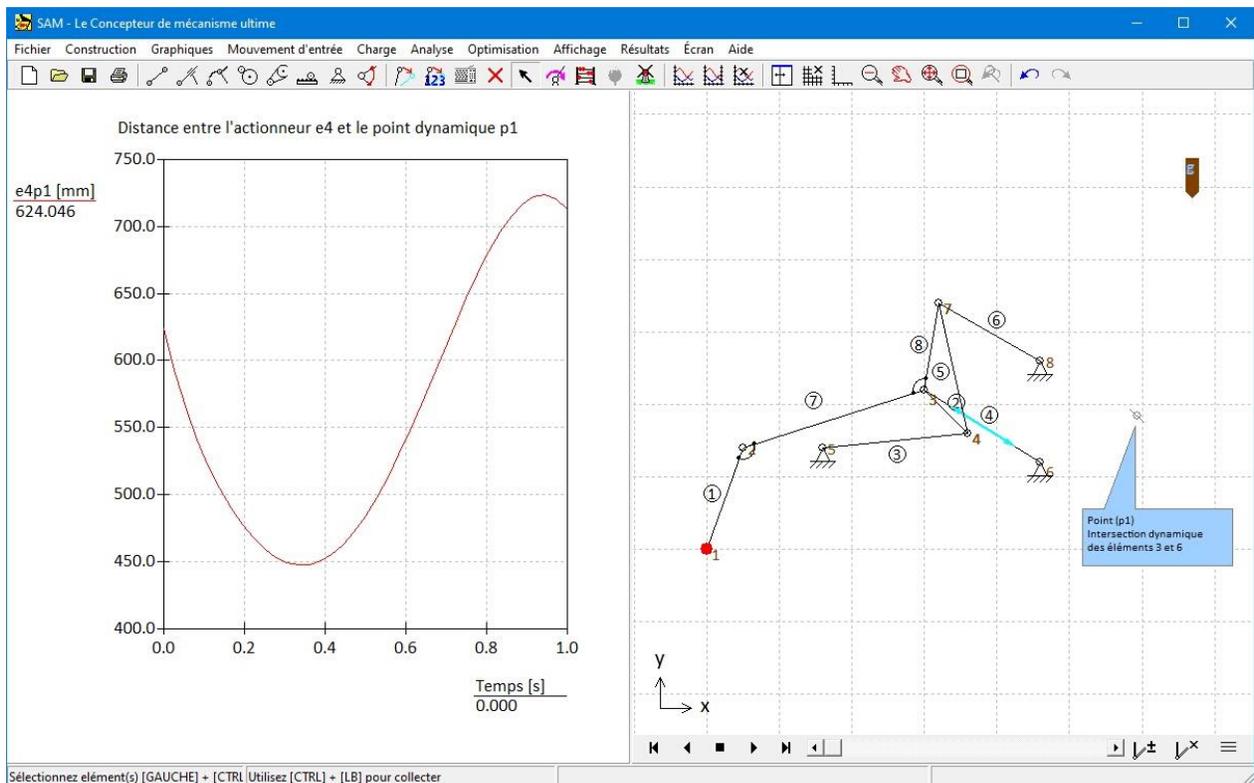
Le module de post-traitement (Résultats > Défini par l'utilisateur...) a été étendu avec une nouvelle fonction : distance(objet1,objet2), dans laquelle un objet peut être soit un nœud, soit un élément, soit un point ou une forme.



Définition de la distance entre l'élément e4 et le point p1.

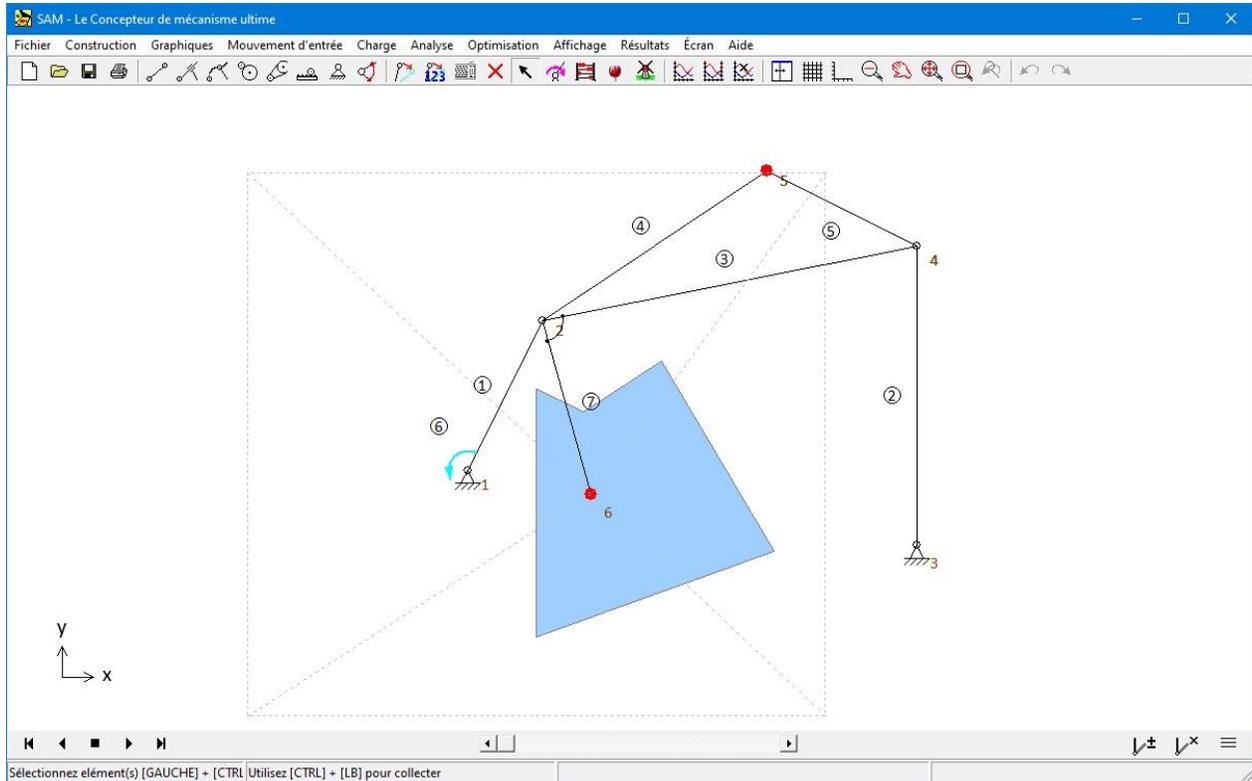
L'utilisation de cette fonctionnalité est multiple :

1. Tracer la distance entre deux objets en fonction du temps. Dans l'exemple précédent, il peut être utile de tracer la distance entre le cylindre moteur (élément e4) et le point dynamique p1 afin de contrôler si cette distance est toujours positive et n'atteint pas zéro, car dans cette situation le mécanisme serait bloqué .

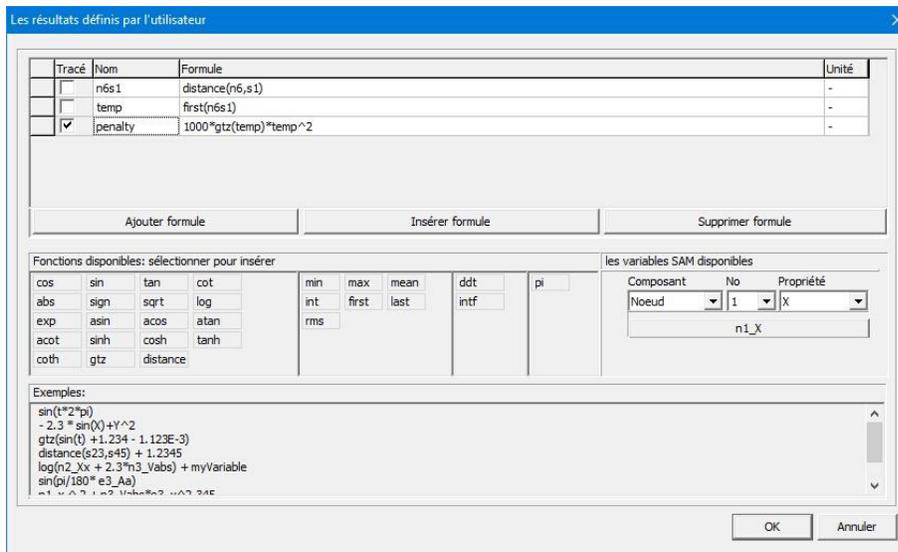


La distance entre l'élément e4 et le point p1 en fonction du temps est tracée à côté du mécanisme

2. Lors d'une optimisation, la position xy d'un nœud peut être un paramètre d'optimisation. En utilisant la fonction de distance, on peut définir une fonction de pénalité pour l'optimisation afin de contraindre l'emplacement xy de ce nœud à n'importe quel polygone.



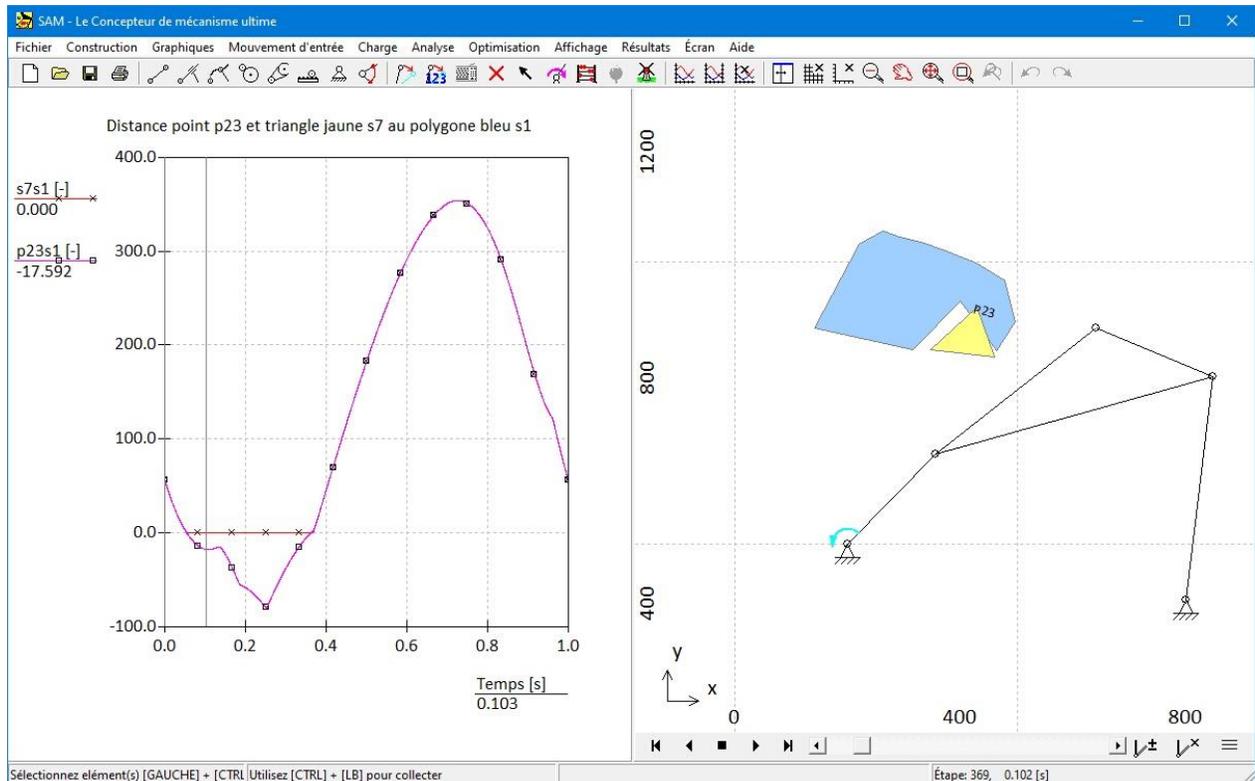
**Exemple d'optimisation dans laquelle le nœud n6 est un paramètre de conception mais doit être contraint à la zone définie par le polygone (forme s1).**



**Définition de la fonction de pénalité pour contraindre le nœud n6 au polygone s1. Dès que le nœud n6 se trouve à l'extérieur du polygone dans la position initiale du mécanisme, la pénalité est supérieure à 0.**

### 3. Détection de collision entre objets

La fonction de distance peut être utilisée pour détecter une collision entre des objets et, selon le type d'objets, même la profondeur de pénétration (ou distance de rétraction minimale pour éviter une collision). Dans l'exemple suivant, le polygone bleu est une forme stationnaire  $s1$ , tandis que le triangle jaune (forme  $s7$ ) est attaché au coupleur. Dans le tracé on voit la variable  $s7s1$ , qui correspond à la distance entre les deux formes et la variable  $p23s1$ , qui correspond à la distance entre le point  $p23$  sur le triangle et le polygone bleu. Dans le cas de deux formes en collision, la distance est définie sur zéro, tandis que dans le cas d'un point ou d'un nœud se déplaçant dans une forme, la distance positive et négative (distance négative = profondeur de pénétration) est indiquée dans le tracé.



Exemple de détection de collision via l'utilisation de la fonction de distance.

#### Copier/Coller (Ctrl-C/Ctrl-V) pour les graphiques/formes

On peut maintenant utiliser Ctrl-C et Ctrl-V pour copier puis coller n'importe quelle forme.