

# TP Calcul Numérique

Nicolas BOUTON

November 21, 2020

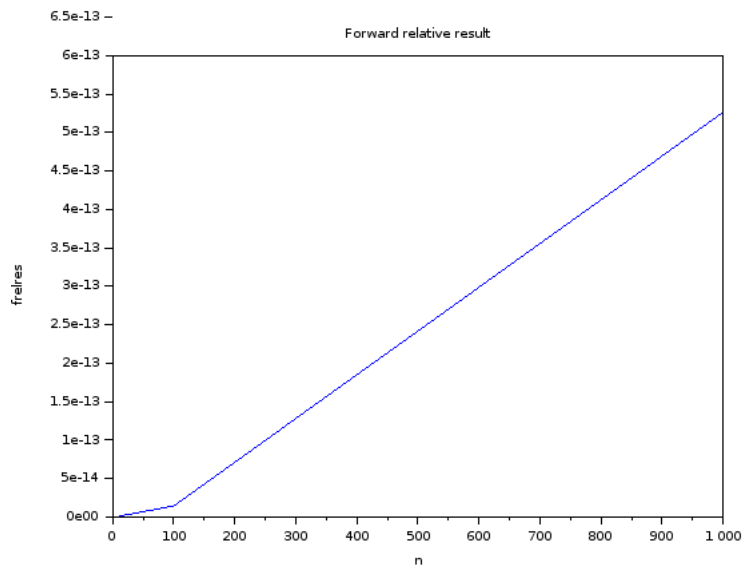
## Exercice 2

Nous remarquons 3 chose :

- Augmentation du conditionnement avec la taille
- Il faut considéré l'erreur arrière entaché du conditionnement pour avoir une approximation des résultats
- Nous manipulons des algorithmes avec certaines complexité qui peuvent être algorithmique ou bien de mémoire

Graphiques :

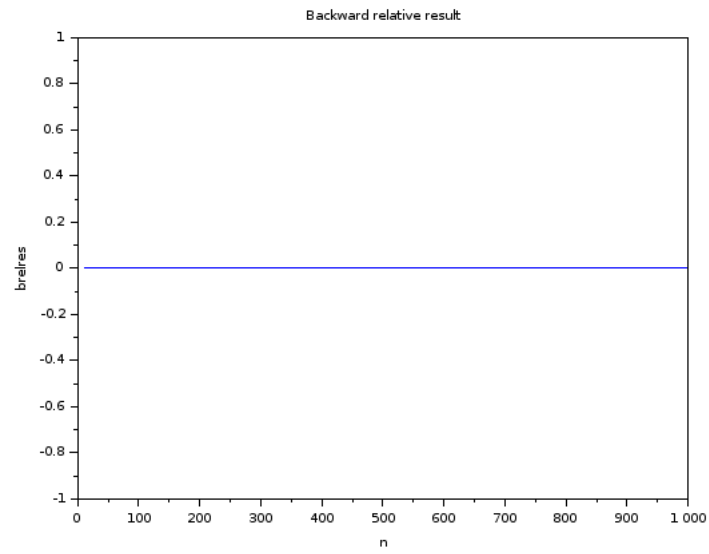
Pour l'erreur relative avant :



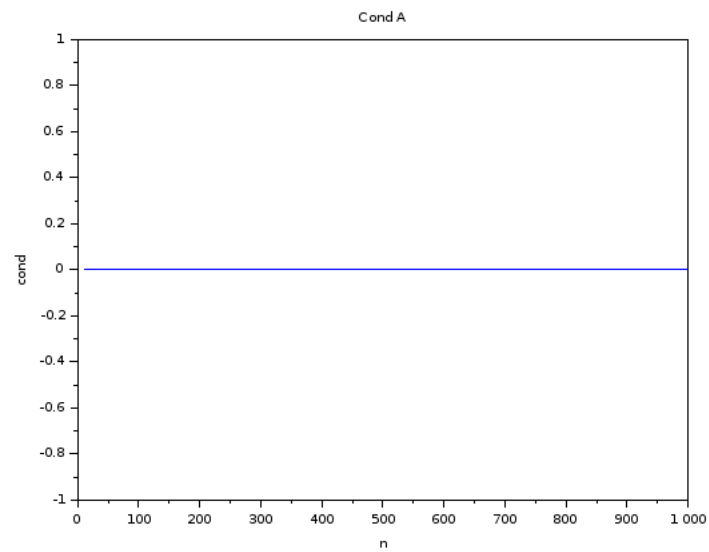
On voit que l'erreur relative avant est très sensible suivant la taille de la matrice.

Graphiques :

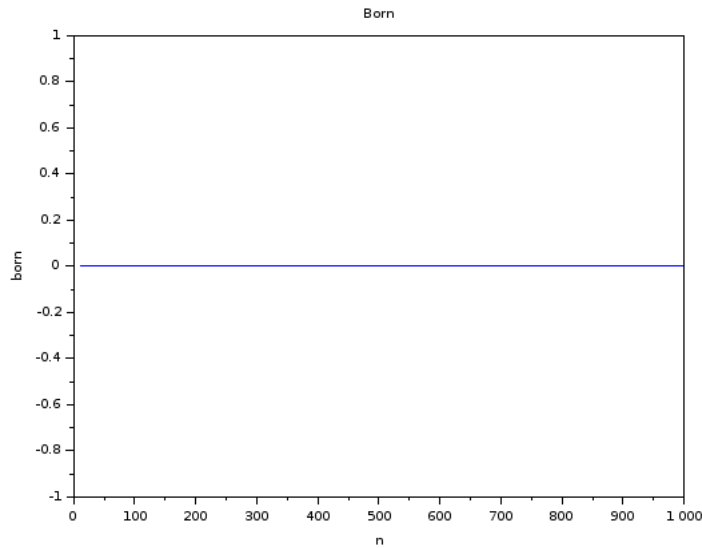
Pour l'erreur relative arrière :



Pour le conditionnement de A :



Pour les bornes :



Nous voyons que pour les autres graphiques la tailles n'influe pas.

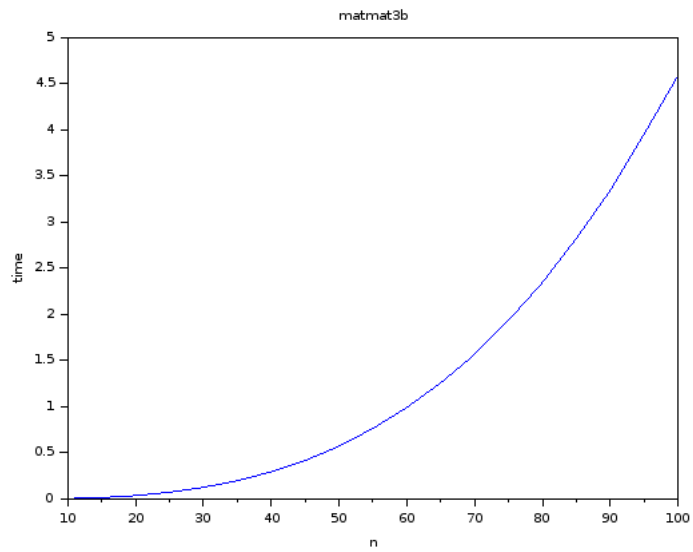
Le code source se trouve dans **exo2.sci**.

### Exercice 3

Le produit **matrice x matrice** à une complexité cubique. Nous remarquons dans un premier temps que l'appel au fonction de **Scilab** pour des calculs **vecteur x vecteur** ou bien **vecteur x matrice** au lieu de déroulé nous même les bloucles est beaucoup plus performant. Il est d'autant plus performant si la taille des matrices est grandes. **Scilab** appelle des noyaux de calcul optimisé tel que **blass**.

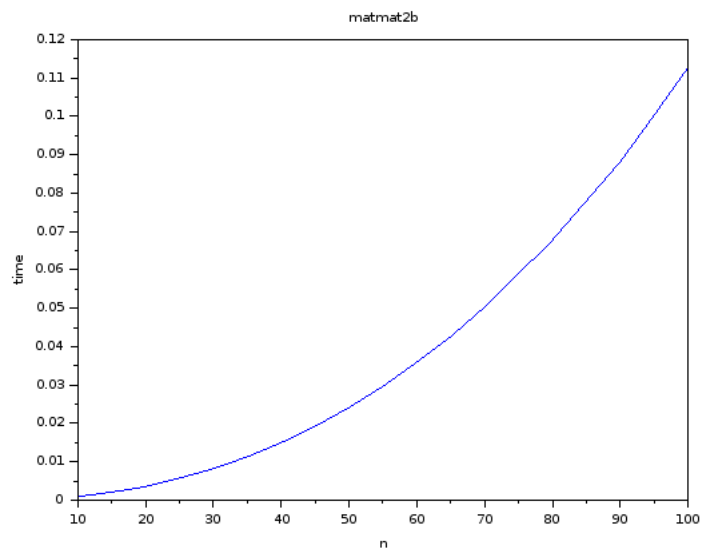
Graphiques :

Pour la fonction **matmat3b** :



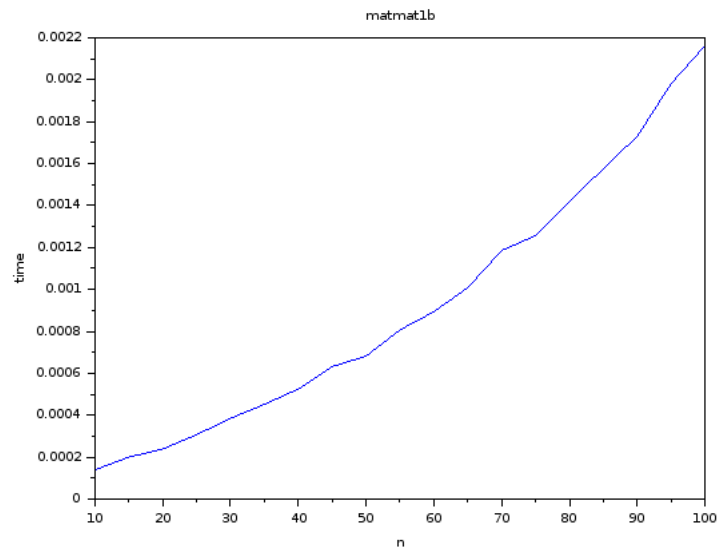
On voit que la fonction suit une complexité cubique.

Pour la fonction **matmat2b** :



On voit que la fonction suit une complexité quadratique.

Pour la fonction **matmat1b** :



On voit que la fonction suit une complexité linéaire.

- Le code des fonction se trouve dans **opmat.sci**.
- Le code de test se trouve dans **exo3.sci**.

## Exercice 4

Bah ça marche ...

## Annexe

Dépot github : [https://github.com/Sholde/CN/partie\\_2](https://github.com/Sholde/CN/partie_2)