

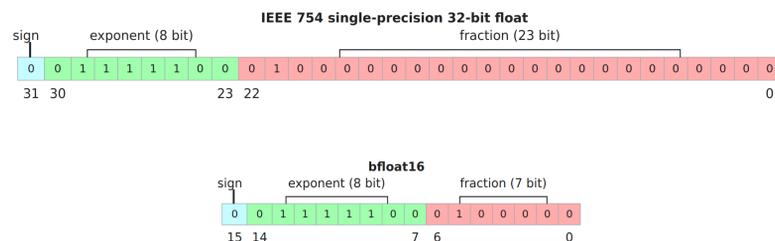
Présentation du projet

Le projet Météorix consiste en un lancement de nano-satellites hautes performances capables de détecter des météores, corps célestes composés de roches, métaux et glaces, directement depuis l'espace au moment de leur entrée dans l'atmosphère terrestre. L'idée est d'utiliser l'échauffement des météores avec l'atmosphère, qui crée des étoiles filantes et produit assez de lumière pour détecter ainsi le phénomène depuis un satellite. [2]



Bibliothèque MPFR et sa surcouche

Au cours de ce projet, nous avons principalement travaillé sur la bibliothèque de calcul multi-précision MPFR, ainsi que sur sa surcouche, MPREAL. Ces bibliothèques sont particulièrement intéressantes pour notre projet puisqu'elles permettent de simuler le fonctionnement du code lorsque ce dernier sera implémenté dans le nano-satellite. [1, 3]



Un exemple d'application : la bibliothèque MPFR permet de simuler des nombres à virgule flottante de précision arbitraire. On peut donc travailler avec des flottants au format bfloat16, comme sur certaines architectures avancées. Ce format est particulièrement intéressant car il diffère du float32 uniquement par la taille de sa mantisse. On peut donc passer de l'un à l'autre par troncature ou bourrage. Remplacer les float32 par des bfloat16 dans certains calculs permet ainsi d'augmenter la bande passante, sans que le coût de conversion de type soit rédhibitoire mais au prix d'une précision plus faible.

Algorithme de HornSchunck

L'algorithme de HornSchunck est une méthode permettant une estimation du flux optique sous plusieurs hypothèses considérées : petits déplacements, illumination constante et déplacements similaires dans un même voisinage (contrainte de lissage).

Division

Une des opérations les plus coûteuses lors de l'exécution de l'algorithme de HornSchunck est la division. Pour essayer de remédier à cela, nous avons envisagé d'autres solutions pour diviser a par b :

- $r = 1/b$ en demi-précision, puis multiplication de a et r ,
- $r = 1/b$ en demi-précision, multiplication de a et r , puis une itération de l'algorithme de Newton-Raphson.

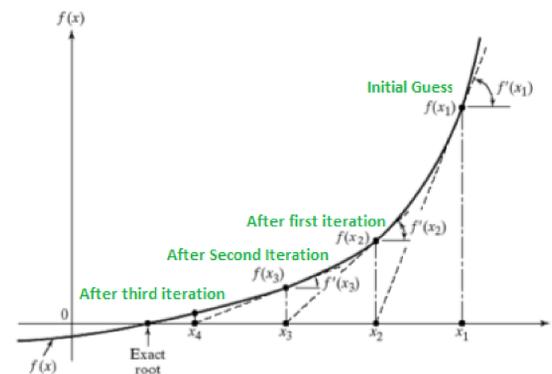
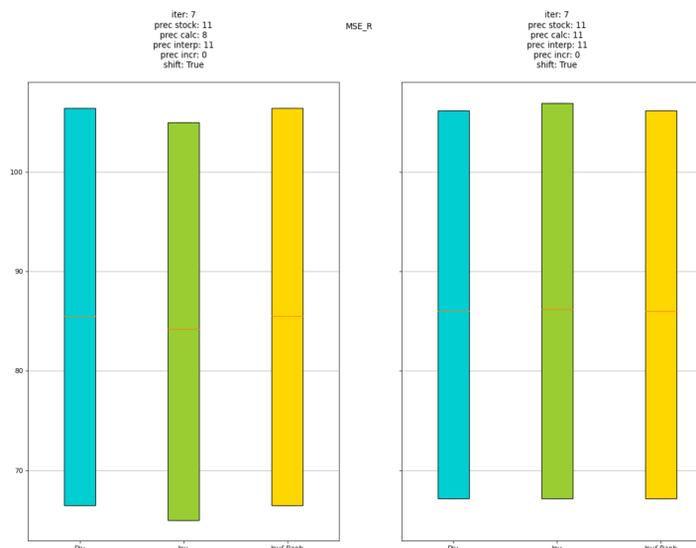


Fig. 4: Illustration de la méthode de Newton Raphson

L'algorithme de Newton-Raphson est une méthode permettant de trouver rapidement la racine d'une fonction réelle. Sa convergence étant quadratique, cette méthode est régulièrement utilisée et préconisée. Dans cette étude, l'algorithme nous permet d'obtenir les derniers bits de précisions manquants suite à l'inversion en demi-précision.



Les boxplots (sans moustache) représentent la répartition des erreurs quadratiques moyennes des réels sur la séquence *coastguard*. On s'attend à obtenir un résultat meilleur (plus faible en moyen) sur la division que sur l'inversion ou l'inversion et méthode de Newton-Raphson. On remarque qu'à gauche ce n'est pas le cas : nous pensons que cela est dû aux faibles paramètres de précision.

Séquence d'images

Le programme calcule le flot optique sur toute une séquence d'images. Pour nos recherches, nous avons à notre disposition une séquence d'images *coastguard*, composée de 300 images de taille 352x288. Dans un premier temps, nous avons utilisé le code écrit pour être appliqué aux images 150 et 151 de la séquence. Une fois nos avancées validées sur ces images, nous avons modifié le code afin de pouvoir traiter de manière exhaustive l'ensemble de la séquence.



Fig. 6: Images 150 et 151 de la séquence

Le flot optique permet de mettre en évidence les mouvements en analysant la position des pixels de la deuxième image par rapport à la première.

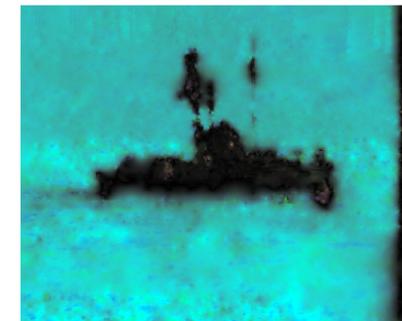


Fig. 7: flot optique des deux images précédentes

En noir sont représentés les pixels qui n'ont pas bougés par rapport à l'image précédente. En bleu, les pixels sont décalés vers la gauche.

References

- [1] Documentation MPFR. <https://www.mpfr.org/mpfr-current/mpfr.pdf>.
- [2] Nicolas Rambaux et al. "Meteorix: a cubesat mission dedicated to the detection of meteors and space debris". In: 2019.
- [3] Site officiel MPFR. <https://www.mpfr.org/>.