



DIRO
IFT 2425

DEVOIR N° 3

Intégration des Trapèze/Romberg & Erreurs Numériques

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle.

[http : //www.iro.umontreal.ca/~mignotte/IFT2425/](http://www.iro.umontreal.ca/~mignotte/IFT2425/)

E-mail : mignotte@iro.umontreal.ca

Estimation de π par Intégration Numérique

De nombreuses formules, de physique, d'ingénierie et bien sûr de mathématiques, impliquent π , qui est une des constantes les plus importantes des mathématiques. π est défini comme le rapport constant entre la circonférence d'un cercle et son diamètre. Cette définition géométrique, la première historiquement et très intuitive, permet de trouver une approximation de π grâce au calcul intégral en le définissant simplement comme étant quatre fois l'aire d'un quart de disque de rayon 1 ($\mathcal{A} = \pi \cdot R^2 = \pi \cdot 1^2$), i.e., en posant l'intégrale suivante :

$$\pi = \int_0^1 4 \sqrt{1-x^2} dx \quad (1)$$

La vraie valeur de cette intégrale est $\pi \approx 3.14159265358979323846264338$.

Méthode des Trapèzes

1- Implémenter un programme (utilisant des nombres DOUBLES) permettant de calculer une valeur approchée de π en utilisant la méthode d'intégration des Trapèzes sur l'Eq. (1). Afficher à la console, le nombre d'intervalles, l'estimée de π et l'erreur (avec 15 chiffres significatifs après la virgule).

Trouver le plus petit nombre d'intervalles permettant d'avoir (expérimentalement) une estimation de π avec une précision de 10^{-8} .

Méthode de Romberg

2- Implémenter un programme (utilisant des nombres DOUBLES) permettant de calculer une valeur approchée de π en utilisant la méthode de Romberg avec 1024 intervalles (i.e., ou 1025 points divisant uniformément le domaine d'intégration $[0; 1]$) sur l'Eq. (1).

On rappelle que la méthode de Romberg utilise la méthode des Trapèzes et autant d'extrapolations

de Richardson qu'il est possible de faire avec ses 1024 intervalles.

Réfléchissez, à partir des éléments donnés dans le cours (cf. Chap. 1 : "Propagation des erreurs numériques"), quelles pourraient être la ou les raisons numériques qui pourraient gêner une plus grande précision de cette estimation comparativement à la précision que l'on devrait obtenir théoriquement.

Méthode Des Suites Itératives d'Archimedes

L'histoire de l'estimation de π , qu'on peut retracer grâce aux écrits disponibles, est divisé en trois parties : la période antique durant laquelle π a été étudié géométriquement, l'ère classique, aux alentours du XVIIe siècle, où les outils du calcul intégral ont permis des avancées dans la connaissance de l'estimation du nombre π , et enfin la période des ordinateurs et des méthodes d'analyse numériques où π a pu être estimé avec des millions de décimales (avec des méthodes d'intégration numériques comme celle de la question précédente).

C'est durant la période de l'antiquité, et dans le traité d'Archimède (287 à 212 avant notre ère) intitulé "DE LA MESURE DU CERCLE" que l'on peut lire une des tout premières méthodes, inventée par Archimedes, pour estimer efficacement une valeur de π . Cette méthode consiste tout simplement à encadrer le périmètre du cercle par une borne inférieure donnée par le périmètre du polygone inscrit et une borne supérieure donnée par le périmètre du polygone circonscrit à un cercle de rayon $1/2$ (cf. Fig. 1).

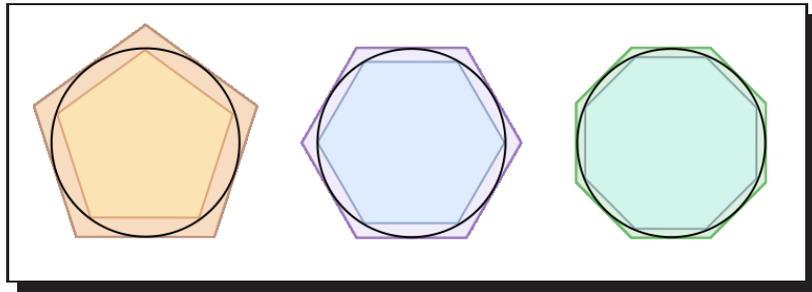


FIG. 1 – Un encadrement de π peut être estimé en calculant les deux périmètres associés au polygone (possédant n cotés) inscrits et circonscrits dans le cercle de rayon $1/2$. Une estimation est ensuite d'autant plus précise que n augmente.

Plus précisément, Archimedes construit une suite itérative qui, à la première itération, encadre la valeur de π en partant d'hexagones inscrits et circonscrits dans le cercle (définissant ainsi un intervalle de confiance $\pi \in [\pi_{min}, \pi_{max}]$, avec π_{min} et π_{max} désignant respectivement le périmètre du polygone inscrit et circonscrit. A chaque itération, la suite itérative d'Archimedes donne une estimation de π (i.e., le milieu de cet intervalle de confiance) en doublant le nombre de cotés des polygones inscrits et circonscrits [1].

Ce raisonnement conduit à la suite itérative $r_{n+1} = g(r_n)$ suivante, qui peut s'écrire avec deux expressions mathématiquement équivalente (en multipliant le numérateur et le dénominateur par l'expression conjuguée, $\sqrt{r_n^2 + 1} + 1$) :

	$r_0 = \frac{1}{\sqrt{3}}$
1 ^{ère} forme : ►	$r_{n+1} = \frac{\sqrt{r_n^2 + 1} - 1}{r_n}$
2 ^{ième} forme : ►	$r_{n+1} = \frac{r_n}{\sqrt{r_n^2 + 1} + 1}$
Dans les 2 cas :	$\pi \approx 6 \times 2^n \times r_n$

3– Implémenter sur ordinateur (en utilisant des nombres DOUBLES), 30 itérations de ces deux suites itératives en affichant à la console le numéro d’itération, l’estimée de π et l’erreur (avec 15 chiffres significatifs après la virgule). Dans le cas où l’estimée à l’itération n n’est plus un nombre (nan : not a number), arrêter la boucle itérative (avec par exemple l’instruction : IF ISNAN(*estimée*) BREAK ;

Afficher sous forme de graphe² (cf. Fig. 2), l’erreur $\epsilon = \ln|\pi - r_n|$ (en ordonnée) entre l’estimée et la vraie valeur de π que l’on fait en fonction du nombre d’itération (en abscisse). On considèrera qu’un pixel en abscisse correspond à 1 itération.

Essayer aussi de comprendre, à partir des éléments donnés dans le cours (cf. Chap. 1 : “Propagation des erreurs numériques”), pourquoi une des deux méthodes itératives est plus stable numériquement et plus précise que l’autre.

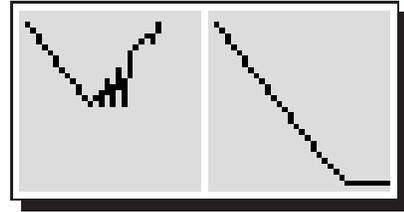


FIG. 2 – Évolution du log de l’erreur en fonction du nombre d’itération pour la première et la deuxième suite.

Références

[1] Wikipédia.

1

Remise & Rapport

Vous devez rendre électroniquement le(s) programme(s) fait en C avant la date de remise spécifiée dans le fichier *barème* situé sur la page web du cours. Pour la remise électronique, utilisez le programme *remise* (*man remise* pour plus de détails) pour remettre votre code dans le répertoire TP<Numéro du Tp>. N’oubliez pas d’inscrire vos noms, courrier électronique en commentaire en haut du fichier .c remis. Les noms des programmes à remettre devront avoir le format suivant *Tp<Numero du Tp>-IFT2425-<Numero de la question>.c*. Les programmes devront se compiler et s’exécuter sur Linux tel qu’indiqué dans le barème.

2

Conseils Pratiques

Utiliser le programme que je vous donne sur ma page web en initialisant la variable `FLAG_GRAPH=1` et remplir le tableau 2D nommé `Graph2D` dans le programme (initialement remplie de blanc) comme une feuille de papier dessin ou vous tracerez l’erreur ϵ de la suite des estimations du calcul de π , pour différentes valeurs de n , en considérant que n est l’abscisse et que l’erreur est l’ordonnée. Le programme que je vous donne sauvera cette image au format PGM. Vous pourrez ensuite la lire avec les logiciels `DISPLAY`, `xv` ou `GIMP` disponible sur Linux.