Série 3: structure de contrôle `if-else`

Comment tester l'égalité en virgule flottante ?

Lien avec le MOOC Initiation à la Programmation (en C++)
Lien avec ICC-Théorie en complément du MOOC

Cette semaine un complément est apporté sur la contradiction entre le message du cours et certains éléments des solutions fournies sur le MOOC concernant le test d'égalité sur des nombres en virgule flottante. Quelques éléments concrets sont apportés à partir des connaissances vues la semaine passée.

Exercices du MOOC

- **Document Tutoriel « résolution d'un polynome de degré 2» niveau 0 :**
  - Distinction des 3 cas avec if-else

Remarque sur l'approche proposée dans ce tutoriel et dans la correction de certains autres exercices du MOOC: la solution du tutoriel réalise un test *d'égalité* sur la valeur d'un nombre à virgule flottante dans 2 cas alors que le cours *souligne que c'est quelque chose à éviter* ! Examinons chacun des deux cas :

- **Test immédiatement après la lecture d'une valeur** : La valeur (double précision) du coefficient **a** est testée vis-à-vis de la valeur **0.0** immédiatement après sa lecture. Dans ce cas particulier le test d'égalité est ok car zéro est exactement représenté. De même c'est ok pour toute autre valeur du domaine couvert en *double précision* si la **variable** testée est elle-même en *double précision*.
  - Cette approche est incorrecte si la **variable** testée est en *simple précision (float)* et que la **valeur** testée n'est pas représentée exactement dans cette représentation (voir cours avec *un dixième*). Eviter d'utiliser des types différents dans le test !

- **Test du résultat d'un calcul :** Un tel test d'égalité **doit effectivement être évité** à cause des approximations faites sur les nombres manipulés. De plus les erreurs absolues qui en découlent sont *amplifiées* par chaque opération réalisée.

  - **Alors comment faire ?**

    1. ***Valeur minimum de la tolérance*** τ : La précision ε de la représentation vous donne l'erreur absolue maximum qui pourrait être faite sur la valeur théorique **V** que vous voulez tester. Dans le pire des cas, on a : $\varepsilon = \delta V/V$.
       Votre tolérance minimum τ doit être: $\tau > \delta V = \varepsilon * V$.
       Cette valeur n'est qu'un minimum que nous devons préciser en évaluant *l'influence des calculs effectués sur les données* dans le pire des cas. Cela est décrit ci-dessous dans une approche plus générale qui englobe aussi le cas des valeurs V construites avec la formule dénormalisée (ex : 0.0).

2. E**stimation de la valeur minimum de τ à partir des données:** on utilise la **précision ε** sur les données manipulées et sur tous les résultats intermédiaires. Il faut se fixer une valeur maximum possible pour ces données pour traiter le pire des cas.

   Pour cette estimation, il faut savoir que :

   a. La *précision d'un produit* est la *somme des précisions*
   b. La *précision d'une division* est la *somme des précisions*
   c. L'*erreur absolue d'une somme* est la *somme des erreurs absolues*
   d. L'*erreur absolue d'une différence* est la *somme des erreurs absolues*

   A titre d'exemple, pour le calcul du discriminant, si on suppose que les valeurs maximum de **a,b,c** peuvent atteindre une valeur de $\mathbf{10^6}$ alors on doit estimer τ pour <u>le pire des cas</u> où ces 3 coefficients ont la valeur de $\mathbf{10^6}$:

   - La précision de la représentation **double** est : $\varepsilon = \mathbf{10^{-15}}$
   - L'erreur relative maximum sur le premier terme X1= b^2 est $\mathbf{2*\varepsilon}$
   - L'erreur relative maximum sur a*c est $\mathbf{2*\varepsilon}$
   - L'erreur relative maximum sur le second terme X2=4*a*c reste $\mathbf{2*\varepsilon}$ car 4 est représenté exactement en binaire
   - L'erreur absolue maximum sur le premier terme (avec b valant 10^6) vaut alors : $\delta X1 = (2*10^{-15})*(10^6*10^6)= 2*10^{-3}$
   - L'erreur absolue maximum sur le second terme (avec a et c valant 10^6) vaut alors : $\delta X2 = (2*10^{-15})*(4*10^6*10^6)= 8*10^{-3}$
   - L'erreur absolue sur le discriminant $\delta\Delta$ est la somme des erreurs absolues $\delta X1$ et $\delta X2$, c'est-à-dire : $2*10^{-3} + 8*10^{-3} = 10^{-2}$
   - *La tolérance τ doit donc être au minimum de* $\boldsymbol{10^{-2}}$

- **Document Exercices semaine 2 du MOOC**

  - **Exercice 4 : Expressions conditionnelles (niveau 2)**
    - Utilisation de l'opérateur et-logique
  - **Exercice 5 : expressions conditionnelles (niveau 2)**
    - Expression logique complexe
  - **Exercice 6 : expressions arithmétiques (niveau3)**
    - La solution du MOOC vous autorise de faire des tests d'égalité sur des calculs en virgule flottante… Gardez à l'esprit les calculs effectués ci-dessus pour avoir une idée des marges de validité des calculs.