

Portfolio de Compétences

BUT SD2 VCOD

Analyse d'un Modèle de Calcul Numérique

Le cas du programme xp

Auteur : Aimad Hamdaoui

Compétence : VCOD - Mettre en œuvre un modèle statistique (Niveau 1)

Date : 23 juin 2025

Table des matières

| | | |
|----------|--|----------|
| 1 | Introduction et Contexte | 2 |
| 2 | Mise en Œuvre et Analyse du Modèle | 2 |
| 2.1 | Comprendre l'intérêt de planifier le recueil des données | 2 |
| 2.2 | Appréhender les difficultés et les limites de la collecte | 2 |
| 2.3 | Comprendre l'impact du type de données sur la modélisation | 2 |
| 2.4 | Apprécier les limites de validité et d'application d'un modèle | 3 |
| 2.5 | Réaliser l'importance d'une procédure de test | 3 |
| 3 | Conclusion | 3 |

1 Introduction et Contexte

Ce document analyse un programme C, nommé `xp`, conçu pour calculer la fonction exponentielle (e^x). L'objectif est de démontrer les compétences du bloc VCOD (Valorisation et Communication des Données) de niveau 1, en particulier la capacité à "mettre en œuvre un modèle statistique" ou, dans ce cas, un modèle mathématique.

Nous considérons l'algorithme du programme (basé sur la série de Taylor) comme le **modèle**, et son exécution comme la **collecte de données** permettant de valider et d'analyser ce modèle.

2 Mise en Œuvre et Analyse du Modèle

Cette section détaille comment le projet répond aux différents critères de la compétence VCOD de niveau 1.

2.1 Comprendre l'intérêt de planifier le recueil des données

Dans le contexte de ce projet, le "recueil des données" consiste à exécuter le programme avec une entrée contrôlée pour obtenir un résultat. La planification a été simple mais cruciale :

- **Choix de l'entrée** : Une valeur simple comme '4' a été choisie pour le test. Cette valeur est assez grande pour que les limites de précision d'une calculatrice standard deviennent visibles.
- **Exécution du programme** : Le binaire `xp` a été exécuté dans un environnement contrôlé (shell Docker) pour garantir la reproductibilité des résultats.

Cette planification a permis de générer une donnée de sortie précise, qui est au cœur de l'analyse.

2.2 Appréhender les difficultés et les limites de la collecte

La principale "difficulté" ou "limite" rencontrée n'est pas dans la collecte elle-même, mais dans la comparaison des données. En exécutant `./xp 4`, le programme a produit un résultat d'une grande précision.

```
1 iut2402941@docker:~/exp2/inst$ ./xp 4
2 54.5981500331443
```

Listing 1 – Exécution du programme `xp` et récupération de la sortie.

La limite apparaît lorsqu'on compare cette sortie avec celle d'une calculatrice standard, qui affiche généralement 54.59815003. Cela met en évidence la **limite de précision** des outils de calcul courants par rapport à un algorithme spécialisé. Le modèle implémenté dans le programme `xp` surpassé cette limite.

```
iut2402941@docker:~/exp2/inst$ ./xp 4
54.5981500331443
```

FIGURE 1 – Capture d'écran de l'exécution et de la sortie du programme `xp`.

2.3 Comprendre l'impact du type de données sur la modélisation

Le programme `xp` utilise très probablement des types de données à virgule flottante de haute précision en C (comme `long double`). Ce choix est fondamental :

- Il permet de stocker un grand nombre de chiffres significatifs.
- Il conditionne la précision finale du modèle mathématique implémenté.

Le modèle lui-même, la série de Taylor pour e^x ($e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$), est une somme infinie. En pratique, l'algorithme calcule un nombre fini de termes. La capacité du type de données à gérer de très petits et de très grands nombres sans perte de précision est donc directement liée à la **fiabilité du modèle**.

2.4 Apprécier les limites de validité et d'application d'un modèle

Le modèle (l'algorithme) est très performant, mais il a des limites :

- **Limites de validité** : Pour des valeurs d'entrée très grandes, le nombre de termes à calculer pour atteindre une bonne précision pourrait devenir prohibitif, entraînant des temps de calcul longs ou des dépassements de capacité (*overflow*).
- **Conditions d'application** : Le modèle est conçu pour des entrées numériques. Il ne fonctionnerait pas avec d'autres types de données.

L'analyse de la sortie montre que pour une entrée comme '4', le modèle est bien dans son domaine de validité et d'application optimal.

2.5 Réaliser l'importance d'une procédure de test

La démarche entreprise constitue une **procédure de test** fondamentale pour valider une hypothèse.

- **Hypothèse** : Le programme `xp` calcule la fonction exponentielle de manière correcte et avec une précision supérieure à celle d'une calculatrice standard.
- **Test** : Exécution du programme avec une valeur connue ('4').
- **Validation** : Comparaison du résultat ('54.5981500331443') avec une valeur de référence ('54.59815003...').

Le test confirme l'hypothèse. La différence de précision observée n'invalidé pas le résultat, au contraire : elle démontre la supériorité du modèle implémenté, validant ainsi sa conception et sa mise en œuvre.

3 Conclusion

L'analyse du programme `xp` a permis de mettre en pratique l'ensemble des compétences de niveau 1 de la ressource VCOD. En traitant un algorithme comme un modèle, nous avons planifié une collecte de données (exécution), analysé ses limites (précision), compris l'impact des choix techniques (types de données) et appliqué une procédure de test simple mais efficace pour valider une hypothèse. Ce projet illustre parfaitement comment les principes de la science des données peuvent s'appliquer à l'analyse de logiciels et d'algorithmes.