

Architecture et Implémentation d'une Application Basée sur les Microservices pour la Résolution d'équations Polynomiales

Hmaidi Mohamed, Haraf Aymen, Seddik Othman, Ijjane Adnane

Université EMSI

Abstract

This report presents a modular microservices-based application designed to solve polynomial equations. Leveraging advanced techniques like numerical and analytical methods, the system uses Spring Boot for backend development and React for the frontend interface. The architecture integrates components such as REST APIs, gateways, and categorized microservices to enhance accuracy, scalability, and user accessibility. By incorporating advanced algorithms like Newton-Raphson, Bisection, and Cardan's formulas, the software efficiently computes polynomial roots. This innovative application demonstrates potential for both academic research and industrial applications, providing a scalable and extensible framework for solving complex mathematical problems.

Keywords: Spring Boot, React, Microservices, Polynomial Equations, Analytical Methods, Numerical Methods

Résumé

Ce rapport présente une application modulaire basée sur les microservices conçue pour résoudre des équations polynomiales. En utilisant des techniques avancées, telles que les méthodes numériques et analytiques, le système repose sur Spring Boot pour le développement backend et React pour l'interface frontend. L'architecture intègre des composants tels que des API REST, des passerelles, et des microservices catégorisés pour améliorer la précision, l'extensibilité et l'accessibilité des utilisateurs. En incorporant des algorithmes avancés, tels que Newton-Raphson, la Bisection et les formules de Cardan, le logiciel calcule efficacement les racines des polynômes. Cette application innovante montre un potentiel

pour la recherche académique et les applications industrielles, en offrant un cadre extensible pour résoudre des problèmes mathématiques complexes.

December 25, 2024

Contents

1 Introduction	2
2 Architecture du Système	2
3 Fonctionnalités Clés	3
4 Méthodes Utilisées	4
4.1 Méthodes Analytiques	4
4.2 Méthodes Numériques	4
5 Exemple Illustratif	4
5.1 Exemple : Polynôme Cubique	4
5.2 Exemple : Méthode de la Bissection	4
5.3 Exemple : Méthode de Newton-Raphson	5
5.4 Exemple : Méthode de Laguerre	5
5.5 Exemple : Méthode de Muller	5
6 Technologies Utilisées	6
7 Impact	6
8 Conclusion	7

1. Introduction

Ce projet fournit une solution extensible pour résoudre des équations polynomiales en utilisant des méthodes numériques et analytiques. L'architecture repose sur des microservices pour garantir la scalabilité et la modularité.

2. Architecture du Système

L'architecture de l'application repose sur plusieurs composants interconnectés

:

- **API REST Client** : Une interface utilisateur frontale développée avec React pour envoyer des requêtes.
- **Passerelle Spring Cloud Gateway** : Point d'entrée unique qui distribue les requêtes aux microservices appropriés.
- **Microservices Catégorisés** : Services autonomes pour les méthodes numériques (Newton-Raphson, Bisection, etc.) et analytiques (Formules quadratiques, cubiques).
- **Conteneurisation** : Tous les microservices sont isolés et déployés dans des conteneurs Docker pour assurer une gestion simplifiée des dépendances et une portabilité.

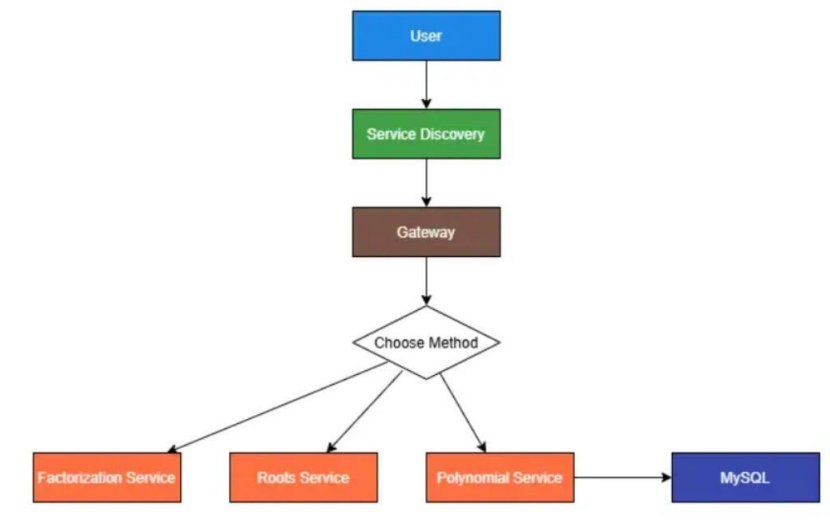


Figure 1: Schéma de l'architecture du système.

3. Fonctionnalités Clés

- **Interprétation intelligente** : Utilisation des LLMs pour interpréter des équations complexes.
- **Précision améliorée** : Calcul avec des tolérances fines.
- **Génération de solutions optimales** : Propose des alternatives et interprétations supplémentaires.

- **Conteneurisation** : Gestion simplifiée des microservices via Docker, permettant un déploiement rapide et cohérent.

4. Méthodes Utilisées

4.1. Méthodes Analytiques

- **Méthode de Newton-Raphson** : Une approche itérative pour trouver les racines en utilisant les dérivées.
- **Méthode de la Bissection** : Division répétée de l'intervalle jusqu'à convergence.
- **Méthode de Laguerre** : Une méthode efficace pour trouver des racines complexes en itérant sur un polynôme donné.

4.2. Méthodes Numériques

- **Formules de Cardan et Ferrari** : Pour les polynômes cubiques et quadratiques.
- **Méthode des Points Fixes** : Itération $g(x)$ pour approcher les racines.

5. Exemple Illustratif

Cas d'utilisation : Résolution du polynôme $x^2 - 4 = [-2, 2]$.

5.1. Exemple : Polynôme Cubique

Résolution du polynôme $x^3 - 6x^2 + 11x - 6 = 0$.

1. L'utilisateur entre le polynôme via l'interface React.
2. La passerelle dirige la requête vers le microservice concerné.
3. Les racines sont calculées et affichées sur l'interface utilisateur : $x = 1, x = 2, x = 3$.

5.2. Exemple : Méthode de la Bissection

- Polynôme : $f(x) = x^2 - 4$.
- Intervalle initial : $[1, 3]$.
- Calcul : La racine trouvée est $x = 2$ après plusieurs itérations.

5.3. *Exemple : Méthode de Newton-Raphson*

- Polynôme : $f(x) = x^2 - 2$.
- Point initial : $x_0 = 1.5$.
- Calcul : Convergence vers $x \approx 1.414$ (racine carrée de 2).

5.4. *Exemple : Méthode de Laguerre*

- Polynôme : $f(x) = x^3 - x - 1$.
- Point initial : $x_0 = 1.5$.
- Calcul : Convergence vers $x \approx 1.324$ (racine réelle).

5.5. *Exemple : Méthode de Muller*

- Polynôme : $f(x) = x^3 - x - 1$.
- Points initiaux : $x_0 = 1, x_1 = 1.5, x_2 = 2$.
- Calcul : Convergence vers $x \approx 1.324$ (racine réelle).

1. L'utilisateur entre le polynôme via l'interface React.
2. La passerelle dirige la requête vers le microservice concerné.
3. Les racines sont calculées et affichées sur l'interface utilisateur : $x = -2, x = 2$.

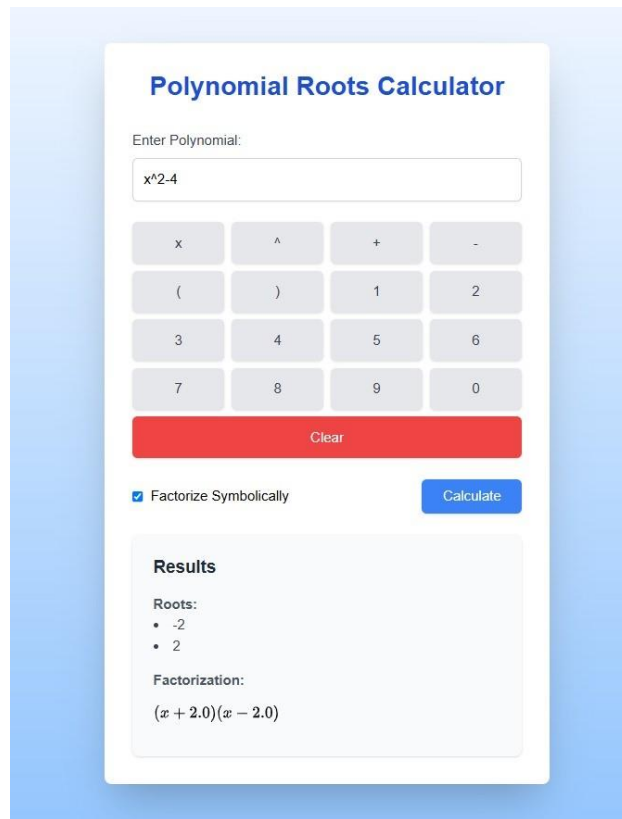


Figure 2: Exemple d'exécution : Résolution d'un polynôme cubique.

6. Technologies Utilisées

Composant	Technologie Utilisée
Interface Utilisateur	React
Passerelle API	Spring Cloud Gateway
Microservices	Spring Boot
Base de Données	MYSQL
Langage de Programmation	Java, TypeScript
Communication Interservices	REST API
Gestion des Dépendances	Maven
Environnement de Déploiement	Docker
Gestion de la Configuration	Netflix Eureka

Table 1: Tableau des technologies utilisées dans le projet.

7. Impact

L'application permet :

- Une meilleure exploration des équations complexes.
- L'intégration de nouvelles méthodes analytiques et numériques, comme Laguerre.
- Une utilisation de la conteneurisation pour un déploiement efficace.
- Une utilisation potentielle dans des cadres académiques et industriels.

8. Conclusion

Ce projet illustre l'efficacité des microservices dans la résolution d'équations polynomiales. L'architecture modulaire garantit une extensibilité pour des méthodes futures.