



Université
de Rennes



Fondation
Université de Rennes

Suivi de Thèse en Modélisation Hydrogéologique

Documentation de Code et Avancées Scientifiques

Version Brouillon – Document de Travail

Auteur : Bastien Boivin

Email (pro) : bastien.boivin@univ-rennes.fr

Email (perso) : bastien.boivin@proton.me

Directeur de thèse :

Jean-Raynald de Dreuzy, Directeur de recherche CNRS, Géosciences Rennes

Co-directeur de thèse :

Luc Aquilina, Professeur des universités, Géosciences Rennes

Partenaire industriel :

Jean-Yves Gaubert, Directeur du pôle R&D, Eau du Bassin Rennais

Rennes, 28 avril 2025

Chapitre 0 Table des matières

Chapitre 1 Introduction

1.1 Objectif de la thèse

Cette thèse vise à développer des modèles hydrogéologiques avancés pour caractériser et prédire les flux d'eau souterraine dans le bassin rennais. Les objectifs principaux incluent l'intégration des données de terrain, la modélisation multi-échelle des écoulements, et l'évaluation de l'impact des changements climatiques sur les ressources en eau.

1.2 Contexte scientifique

La compréhension des flux hydrogéologiques est essentielle pour la gestion durable des ressources en eau. Dans le contexte du bassin rennais, les interactions entre eaux souterraines et eaux de surface sont particulièrement complexes en raison de la géologie hétérogène et des pressions anthropiques croissantes. Ce projet s'inscrit dans une démarche globale d'amélioration des connaissances scientifiques sur les systèmes aquifères fracturés.

1.3 Exemples de code moderne

Le code suivant montre un exemple simple de classe Python avec coloration syntaxique adaptée :

Une petite classe Python

```
1 class Compteur:
2     def __init__(self, start=0):
3         self.count = start          # <- 'self' est color
4     def increment(self):
5         self.count += 1
6         print(f"Valeur : {self.count}")
7
```

```
8  # Usage
9  if __name__ == "__main__":
10     c = Compteur()
11     for _ in range(3):
12         c.increment()
```

Ce code simple illustre l'utilisation d'une classe Python avec initialisation et méthode d'incrément. La colorisation syntaxique met en évidence les différents éléments du code pour une meilleure lisibilité.

??putlabel