

**LISTE DES CAPACITES NUMERIQUES AU PROGRAMME**

<b>BCPST1</b>	<b>1. Incertitudes-types composées</b>	Simuler, à l'aide d'un langage de programmation ou d'un tableur, un processus aléatoire – simulation de Monte-Carlo – permettant de caractériser la variabilité de la valeur d'une grandeur composée
	<b>2. A/B</b>	Tracer, à l'aide d'un langage de programmation, le diagramme de distribution des espèces d'un ou plusieurs couple(s) acide-base
	<b>3. A/B</b>	Déterminer la valeur du point isoélectrique d'un acide aminé
	<b>4. Cinétique</b>	A partir de données expérimentales, tracer l'évolution temporelle d'une concentration, d'une vitesse volumique de formation ou de consommation, d'une vitesse volumique de réaction
	<b>5. Cinétique</b>	A partir de données expérimentales, déterminer les ordres partiels, la constante de vitesse et l'énergie d'activation
	<b>6. Cinétique</b>	Tracer l'évolution des concentrations par résolution numérique de l'équation différentielle
<b>BCPST2</b>	<b>7. Régression linéaire</b>	Simuler un processus aléatoire de variation des valeurs expérimentales de l'une des grandeurs simulation de Monte-Carlo – pour évaluer l'incertitude sur les paramètres du modèle
	<b>8. Cinétique</b>	A l'aide d'un langage de programmation, tracer, dans le cas de deux actes élémentaires successifs, l'évolution des concentrations par résolution numérique du système d'équations différentielles et mettre en évidence les conditions d'application de l'approximation de l'état quasi-stationnaire
	<b>9. Cinétique</b>	Etablir un système d'équations différentielles et le résoudre numériquement, à l'aide d'un langage de programmation, afin de visualiser l'évolution des concentrations au cours du temps et mettre en évidence les situations de contrôle cinétique ou thermodynamique
	<b>10. Précipitation</b>	Déterminer les conditions optimales pour séparer deux ions par précipitation sélective
	<b>11. Mécanique</b>	Mettre en évidence le non-isochronisme des oscillations d'un oscillateur anharmonique
	<b>12. Mécanique des fluides</b>	Résoudre, à l'aide d'un langage de programmation, l'équation différentielle vérifiée par la vitesse, en utilisant une modélisation fournie du coefficient de traînée $C_x$ en fonction du nombre de Reynolds, dans le cas de la chute d'une bille sphérique dans un fluide newtonien

## **EXEMPLES**

1. Compléter le programme (méthode Monte-Carlo) :

[https://colab.research.google.com/drive/1koEmj4oNeFHcWtd1ME-b\\_DxCp0rznGbR](https://colab.research.google.com/drive/1koEmj4oNeFHcWtd1ME-b_DxCp0rznGbR)

3. Compléter le programme (méthode de la dichotomie et fonction bisect) :

<https://colab.research.google.com/drive/1JoFnMbPgQrGz-BO8RPFVOmlaIsMxERH8>

6. Compléter le programme (méthode Euler) :

<https://colab.research.google.com/drive/18ahtsqPYSrHWXfMcMP3LQYdtJGgLEeD>

7. Expliquer le programme par bloc (régression linéaire et incertitudes) :

<https://colab.research.google.com/drive/1yji8kQKRwJUWArbYQiCcdrcuG3zD6Mbo>

9. Compléter le programme (fonction odeint) :

[https://colab.research.google.com/drive/1AtZwlfOvbHG-2bYiY0XfqmDaRI9R5z\\_](https://colab.research.google.com/drive/1AtZwlfOvbHG-2bYiY0XfqmDaRI9R5z_)

10. Expliquer le programme par bloc (en retrouvant notamment les expressions données dans les deux boucles) :

[https://colab.research.google.com/drive/1r0TAKHa1jBqtr\\_7BaS2K\\_aj2Y0VaWFT5](https://colab.research.google.com/drive/1r0TAKHa1jBqtr_7BaS2K_aj2Y0VaWFT5)

11. Compléter le programme (fonction odeint) :

<https://colab.research.google.com/drive/1DPZHGCrsPnO7vn68iECPSRmp2UCI5EdF>

12. Compléter le programme pour un régime de Newton turbulent (fonction odeint) :

[https://colab.research.google.com/drive/1MbgttqP\\_r2S\\_14r6ZwiNuf3Gp4RIYvd\\_](https://colab.research.google.com/drive/1MbgttqP_r2S_14r6ZwiNuf3Gp4RIYvd_)