Tangram
Revista de Educação Matemática

DOI: 10.30612/tangram.v6i3.16950

Interdisciplinarité et difficulté d'apprentissage des méthodes numériques en programmation

Interdisciplinaridade e dificuldade de aprendizagem de métodos numéricos em programação

Interdisciplinarity and difficulty in learning numerical methods in programming

Rahim Kouki

Université de Tunis el Manar
Institut Préparatoire aux Etudes d'Ingénieurs
Tunis, Tunísia
E-mail: rahim.kouki@ipeiem.utm.tn
0000-0002-8664-731X

Soumaya Derragi

Université de Tunis el Manar
Institut Supérieur aux Etudes scientifiques et Techniques
Tunis, Tunísia
E-mail: soumaya.darragi@ipest.ucar.tn
0009-0002-7791-1825

Résumé : Les algorithmes numériques font l'objet d'un enseignement explicite dans les classes préparatoires à l'entrée aux écoles d'ingénieurs. Le caractère interdisciplinaire de ces derniers, permet de construire un pont entre le langage et l'action et met l'accent sur l'utilité d'une certaine rigueur scientifique expérimentée. Notre recherche s'inscrit dans le cadre d'une démarche réflexive avec une prise de conscience centrée sur les difficultés liées à l'implémentation de la méthode d'Euler comme algorithme numérique pour la résolution des équations différentielles. L'exploration d'un milieu théorique fondé sur les méthodes numériques d'approximation, que nous avons menée, nous a permis de dégager différents types d'obstacles didactique rencontrés lors de l'implémentation de la solution numérique d'ordre sémiotique, organisationnel ou encore psychologique. Elle a aussi dénoté d'une certaine complexité syntaxique des relations de récurrences à analyser dans un aspect sémantique. Nous nous plaçons alors dans le cadre de la théorie des champs conceptuels développée dans les travaux de (Vergnaud, 1990) croisée à la notion de registres développé par (Duval, 1993) afin d'analyser ces difficultés dans une dimension syntaxique/sémantique (Kouki, 2018).

Mots clés : Réflexivité. Algorithme numérique. Obstacle didactique e syntaxique/sémantique.

Resumo: Algoritmos numéricos são objeto de ensino explícito em aulas preparatórias para ingresso em escolas de engenharia. O caráter interdisciplinar destas últimas, permite construir uma ponte entre linguagem e ação e enfatiza a utilidade de um certo rigor científico experimental. A nossa pesquisa insere-se numa abordagem reflexiva com uma tomada de consciência centrada nas dificuldades relacionadas com a implementação do método de Euler como algoritmo numérico para a resolução de equações diferenciais. A exploração de um ambiente teórico baseado em métodos numéricos de aproximação, que realizamos, permitiu identificar diferentes tipos de obstáculos didáticos encontrados durante a implementação da solução numérica de ordem semiótica, organizacional além de psicológico. Também denotou certa complexidade sintática das relações de recorrência a serem analisadas sob o aspecto semântico. Situamo-nos então no quadro da teoria dos campos conceptuais desenvolvida na obra de (Vergnaud, 1990) cruzada com a noção de registos desenvolvida por (Duval, 1993) para analisarmos essas dificuldades numa dimensão sintática/semântica (Kouki , 2018).

Palavras-chave: Reflexividade. Algoritmo numérico. Obstáculo didático e sintático/ semântico.

Abstract: Numerical algorithms are explicitly taught in preparatory classes for entering engineering schools. The interdisciplinary character of the latter allows building a bridge between language and action and emphasizes the usefulness of a certain experimental scientific rigor. Our research is part of a reflective approach with an awareness centered on the difficulties related to the implementation of Euler's method as a numerical algorithm for solving differential equations. The exploration of a theoretical environment based on numerical approximation methods, which we carried out, allowed us to identify different types of didactic obstacles encountered during the implementation of the semiotic, organizational and psychological numerical solution. It also denoted a certain syntactic complexity of the recurrence relations to be analyzed under the semantic aspect. We are therefore situated within the framework of the theory of conceptual fields developed in the work of (Vergnaud, 1990) crossed with the notion of registers developed by (Duval, 1993) to analyze these difficulties in a syntactic/semantic dimension (Kouki , 2018).

Keywords : Reflexivity. Numeric algorithm. Didactic and syntactic/semantic obstacle.

Recebido em
21/04/2023
Aceito em
25/09/2023

INTRODUCTION

L'enseignement de l'informatique a été introduit dans le système éducatif tunisien très tôt (Année scolaire 1992-1993). Deux niveaux scolaires étaient concernés par cette matière à l'époque, à savoir la 3ème année et la 4ème année secondaire scientifique et économie. Il s'agissait d'une matière optionnelle. Faute de matériel, l'expérience a démarré dans quelques lycées. Quelques années plus tard, la matière informatique a été généralisée pour toutes les sections (3e année et 4e année). Depuis 2003, on a introduit dans les trois niveaux du deuxième cycle de l'enseignement de base la matière informatique.

À partir de la réforme de 2005, bientôt (16 ans), la matière informatique est devenue obligatoire pour toutes les sections du lycée et du collège. Les programmes de la matière informatique se sont enrichis davantage. Le contenu des cours étant plus adapté à la spécialité choisie par l'élève c'est-à-dire la manière dont l'informatique a été enseignée dépendait étroitement de la filière (littéraire ou scientifique) car c'est en fonction de cette dernière que le choix fut sur l'apprentissage de l'informatique en tant qu'outil ou en tant qu'objet d'enseignement.

Dans les sections scientifiques, le choix de la Tunisie était fondamental : l'informatique est une discipline scientifique et technique enseignée à tous, dès les enseignements scolaires, par des spécialistes reconnus comme tels et ceci sans oublier sa complémentarité avec l'utilisation de l'informatique des TIC dans les différentes disciplines qui doit se développer et se poursuivre.

Quant à la résolution de problèmes au lycée, elle est principalement focalisée sur des problèmes manipulant des chaînes de caractères et des tableaux statiques, un type de problème riche en itérations, les problèmes numériques sont rarement avancés, ce qui peut expliquer en partie, ce que nous remarquons comme difficulté rencontrée chez nos étudiants dans ce type d'applications.

Un deuxième constat est que les élèves au lycée, ayant étudié l'algorithmique et la programmation, ont « un déjà là », les notions de bases sont déjà introduites aussi bien d'un point de vue syntaxique via le langage Pascal que d'un point de vue

sémantique via les problèmes résolus. Ceci sous-entend aussi que des schèmes existent déjà pour une catégorie de problèmes.

Les bacheliers sont en effet les « inputs » des universités et malgré qu'un enseignement de deux ans de l'algorithmique et de la programmation, les élèves arrivent au cursus universitaire, avec des lacunes immenses en matière de démarche de résolution de problème, un constat général est que les élèves apprennent plus ce que comprennent ! Les algorithmes de recherche et de tri en sont le meilleur exemple.

En classes préparatoires les concepts de bases de la programmation (variable, structure de contrôle, sous-programme…) sont repris en première année avec le langage de programmation Python avec son typage dynamique et sa syntaxe simple de haut niveau. L'enseignement des algorithmes numériques est mis en accent dans la réforme des programmes de 2015 dans le programme d'informatique[1]. Cependant, cette partie demeure absente dans les programmes de mathématiques[2] dans la nouvelle réforme ce qui dénote d'un décalage frappant entre les programmes des deux disciplines. Dans les programmes de mathématiques le mot algorithme demeure absent en deuxième année et figure seulement à deux reprises dans les programmes de première année une fois avec l'algorithme d'Euclide en arithmétique et une autre avec le Pivot de Gauss dans le calcul matriciel et résolution des systèmes linéaires, toutes autres méthodes numérique telle que la résolution approchée d'équations différentielles ou les méthodes d'intégrations demeurent absentes. Voilà une deuxième raison menant au choix de ce domaine d'investigation.

Le principe d'itération consiste l'un des objets didactique les plus étudiées, l'absence de control de l'aspect dynamique d'un algorithme et la façon dont il se comporte en fonction des données qu'on lui fournit en entrée est l'une des premières difficultés à affronter de la part des étudiants, en ne considérant que son aspect statique c'est-à-dire la suite finie d'instructions que nous avons sous les yeux, il est parfois difficile de comprendre son comportement dynamique (Rogalski & Lagrange, 2017).

[1] http://www.ipein.rnu.tn/useruploads/files/t-prog-informatique.pdf
[2] http://www.ipeit.rnu.tn/sites/default/files/Prog%20Maths%20MP%202%C3%A8me%20ann%C3%A9e.pdf

Pour apporter des éléments de réponses, nous avons opté de travailler sur des algorithmes numériques, plus particulièrement sur la résolution numérique des équations différentielles par la méthode d'Euler explicite, nous rappelons dans ce qui suit le concept d'algorithmes numérique en mettant l'accent sur son interdisciplinarité puis nous présentons l'objet de notre étude à savoir les équations différentielles.

ALGORITHME NUMERIQUE ET INTERDISCIPLINARITE

Un algorithme numérique consiste à utiliser l'ordinateur (ou un calculateur) pour la résolution d'un problème réel. Les étapes à suivre afin d'élaborer cet algorithme numérique sont généralement les suivantes : nous partons d'un problème réel physique ou mécanique, auquel nous cherchons une modélisation dans un formalisme mathématique, c'est ici qu'un choix d'une méthode de résolution a lieu. Cette méthode est mise en pratique par un programme informatique afin de résoudre le problème en question. Il est donc évident que face à cette situation l'étudiant fait appel à ses triples compétences : celles des sciences physiques d'un côté, des mathématiques de l'autre, et de l'informatique, par ailleurs. Il est confronté à trouver un compromis « cognitif » entre l'ensemble des connaissances dont il dispose dans ces différentes disciplines, parfois complémentaires, parfois contradictoires, nous retrouvons alors la notion de différents champs conceptuels mis en relation (Vergnaud, 1989). Il est également dans un état de métacognition car il devrait être en mesure d'élaborer une solution générique résolvant ce problème mais aussi tout type de problème semblable, une étape de dévolution est alors nécessaire.

L'enseignant d'informatique, quant à lui, il est aussi confronté à un double problème de transposition et devrait être en connaissance parfaite du phénomène physique, sa modélisation mathématique avec la syntaxe mathématique et enfin la programmation de la méthode choisie pour la résolution du problème dans une syntaxe informatique, le retour sur l'expérience de l'enseignant jouera ainsi un rôle primordial dans ce processus d'apprentissage. Voir figure1 (Daniel, 2021)

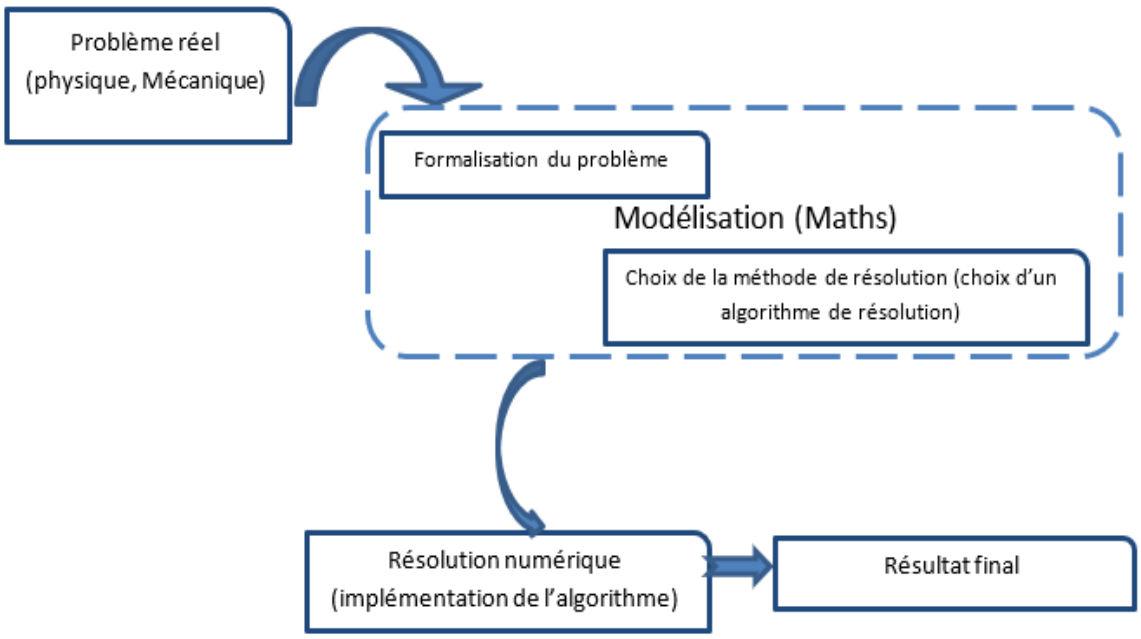


Figure 1. Du problème au Programme.

En observant ce double problème de transposition didactique, et la mise en relation de plusieurs disciplines, il est clair que ce schéma est soumis à plusieurs sortes d'erreurs : au premier niveau –physique- on pourrait avoir des erreurs de mesures (réelles) ensuite on pourrait avoir des erreurs de modélisation –mathématiques- de discrétisation, de facteurs négligés, formules approchées…ou tout simplement erreurs liées à la programmation même. (Daniel, 2021)

Dans notre travail de recherche nous menons une étude qualitative via des tests d'évaluations et nous nous intéressons principalement à l'interprétation des erreurs liées à la programmation d'un point de vue syntaxique et/ou sémantique. Ainsi l'interprétation des erreurs se fera dans ces deux points de vue et nous exploiterons ceci afin des dégager les obstacles cognitifs auxquels sont confrontés les étudiants.

Méthode d'Euler comme méthode de résolution d'équations différentielles : Au cœur de l'interdisciplinarité

De nombreux phénomènes physiques se modélisent naturellement à l'aide d'équations différentielles pour lesquelles on ne dispose pas de solutions analytiques. En mathématiques, ces équations ont également leur intérêt propre, et étudier le comportement qualitatif de solutions est nettement plus aisé si on peut visualiser une approximation raisonnable de celles-ci numériquement.

Au sens de (Vergnaud, 1989) nous remarquons déjà que la notion d'équations différentielles interfèrent principalement avec deux champs conceptuels : celui des maths et celui des sciences physiques et que la recherche d'une solution approximative numérique nous emmène aux différents registres de représentation au sens de Duval tel que les registres algébriques, graphique mais aussi de la programmation en informatique.

Nous rappelons ici la méthode d'Euler pour la résolution des équations différentielles

Lorsqu'il n'est pas possible de résoudre analytiquement une équation (E), c'est-à-dire de trouver une expression explicite de la solution f de (E) qui vérifie f(a)=y0, on met en œuvre un tracé approché de la courbe intégrale sur l'intervalle I= [a,b]. Pour h suffisamment petit, on peut assimiler la courbe de f à sa tangente avec l'approximation suivante :

Si y est solution alors pour tout $t_0 \in$ R, si h est suffisamment petit on a

$$\frac{(y(t_0+h)-y(t_0))}{h} \approx y'(t_0)$$

ce qui donne $(t_0+h) \approx y(t_0)+h*y'(t_0)$ soit (puisque y est solution de (E))

$$y(t_0+h) \approx y(t_0)+h*F(t_0,y(t_0))$$

Mais alors si h suffisamment petit est fixé (h est appelé pas du schéma) on obtient de façon itérative une suite de valeurs approchées y_k de (t_0+kh) pour $k \in$ N de la façon suivante : (Julien, 2019-2020) (figure 2)

$$y_0 = y0$$

$$\forall k \in N, y_{k+1} = y_k + hF(t_0 + kh, y_k)$$

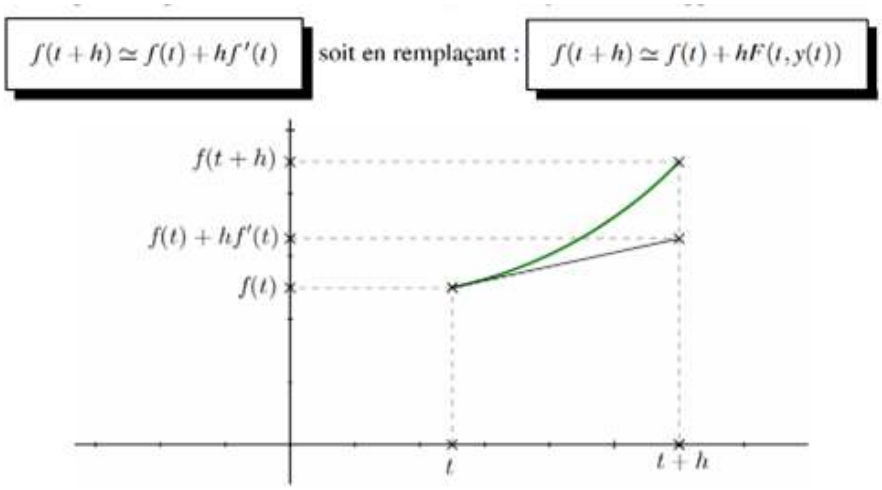


Figure 2. Méthode d'Euler explicite

En chaque point de la subdivision, on approchera donc $f(t_i)$par y_i défini par:

$\forall i \in [\![0, n-1]\!], y_{i+1} = y_i + h f'(t_i) = y_i + hF(t_i, y_i)$ avec $y_0 = y(t_0) = a$.

On approche alors f sur [a,b] par une fonction affine par morceaux et la courbe intégrale de f par une ligne polygonale. L'algorithme s'effectue de proche en proche puisque la position d'un point au temps ti ne nécessite que la connaissance de la position au temps ti-1 (Julien, 2019-2020)

On se donne donc une condition initiale le point qui est en y_0 à l'instant t_0

$$y_1 = y_0 + hF(t_0, y_0)$$

$$y_2 = y_1 + hF(t_1, y_1)$$

$$\ldots\ldots$$

$$y_n = y_{n-1} + hF(t_{n-1}, y_{n-1})$$

Il est clair que l'algorithme suit un processus itératif et que lorsque la structure itérative manipule des variables, ou change les états du contenu des variables dynamiquement. La compréhension du changement d'état des variables dans un processus itératif hors contrôle constitue un double défi : un problème lié à la syntaxe de type « y=y+1 » et sa présentation, et le deuxième problème lié à la sémantique que porte cette syntaxe, nous analyserons ces difficultés d'un point de vue syntaxique/sémantique.

Notre démarche tente, sans introduire un formalisme excessif, d'apporter des éléments de réponses aux questions fondamentales qui se posent en algorithmique et en programmation :

Quelles sont ces principales difficultés rencontrées en programmation et pourquoi certaines difficultés persistent ?

Quel est l'effet de la syntaxe de la représentation mathématique/Physique sur la syntaxe informatique ?

En quoi cette syntaxe entrave les réflexions et le raisonnement des étudiants ?

MÉTHODOLOGIE

Le but de notre étude didactique est donc, de toucher de près la nature des dites difficultés se basant sur une analyse des champs conceptuels inspirée de (Vergnaud, 1990) croisé avec le concept de registre (Duval, 1993).

Le travail expérimental, consiste en une évaluation à destination des élèves des classes préparatoires de l'IPEST de deuxième année portant sur la résolution d'une équation différentielle de premier ordre de la forme $ay'(x) * + by(x) + c = 0$ au moyen d'un algorithme numérique implémentant la méthode d'Euler, dans un paradigme objet en présentant la relation de récurrence inspirée du domaine de la physique c'est-à-dire dans une syntaxe du domaine de la physique.

A travers cette démarche expérimentale, nous essayons d'expliquer l'impact de la nature de la relation de récurrence d'un point de vue syntaxique sur sa représentation en informatique ainsi que le sens logique auquel porte l'étudiant (Tarski, 1923-1944).

Cette étude qualitative a été effectué en deux temps :

Une première partie théorique où l'étudiant est amené à faire une trace d'exécution à la main des premières itérations pour des valeurs de a, b et c donnés afin de dégager les solutions numériques y1, y2 et y3 pour les instants t1, t2 et t3, ceci révèle d'une compréhension de la méthode numérique dans son aspect mathématique

Une deuxième partie expérimentale où l'étudiant devrait passer à l'implémentation de la fonction de résolution dans la syntaxe du langage Python.

L'idée derrière, est simple, si l'étudiant est capable de calculer à la main les premières itérations ceci révèle d'une assimilation de l'algorithme numérique c'est-à-dire de la sémantique dans le cadre des mathématiques, par contre si cette compréhension n'est pas suivie d'une implémentation correcte ceci signifie que le problème est purement informatique lié à la syntaxe de la programmation ou au paradigme de programmation proposé à savoir la programmation orientée objet.

Dans le cas contraire nous pourrons conclure que l'obstacle est purement mathématique la méthode d'Euler n'a pas été bien assimilée, une réponse à l'une des questions de recherche sur la difficulté rencontrée par l'étudiant en situation d'apprentissage sur l'hypothèse que l'étudiant serait incapable d'implémenter une relation de récurrence si cette dernière n'a pas été assimilée dans le cadre mathématique.

Encore une fois, nous retrouvons ici cet aspect d'interdisciplinarité qui réapparait entre mathématique, et informatique.

Enfin, afin d'analyser les copies des élèves, nous avons procédé à une catégorisation de leurs réponses s'appuyant sur les dimensions sémantique/syntaxiques. Nous avons comparé leurs travaux par rapport à notre analyse à priori des procédures envisageables. Nous avons dégagé ainsi les principaux obstacles didactiques et difficultés auxquelles ils ont été confrontés au moment de la résolution des problèmes. L'étude des difficultés que nous envisageons, dans ce travail, est effectuée dans un sens sémantique/syntaxique au sens de (Kouki, L'articulation des dimensions syntaxiques et sémantiques en algèbre du secondaire, 2018) (Kouki & Griffiths, 2021). Nous dégagerons également les obstacles d'ordre interdisciplinaire aussi bien organisationnel qu'épistémologique.

Nous présentons dans ce qui suit le travail expérimental effectué auprès des étudiants de deuxième année avec la représentation de la relation d'Euler présente

dans la littérature sous deux formes $y_{(t+dt)} = y_t + h\,F(t, y_t)$ inspiré du domaine de la physique.

INVESTIGATION EXPERIMENTALE

C'est une évaluation de 30 mn faisant intervenir une soixantaine d'étudiants de deuxième année. Lors de cette évaluation, nous avons rappelé la méthode d'Euler pour la résolution d'équations différentielles de premier ordre à coefficients réels. Le premier objectif était de savoir si cette méthode a été bien assimilée mathématiquement, mais aussi de tester la capacité des étudiants à généraliser cette méthode en y ajoutant un niveau d'abstraction et ce en utilisant la POO enseignée en deuxième année à l'aide du langage de programmation Python, l'idée est de savoir si l'étudiant a acquis l'expérience nécessaire au bout de deux ans de passer d'une implémentation procédurale directe sur un problème donné à une modélisation et la présentation d'une solution générique, cette maturité de réflexion dénotera d'un certain niveau de métacognition et d'abstraction atteint par l'étudiant.

L'objectif est double :

La méthode d'Euler a-t-elle été bien assimilée en première année ?

L'étudiant est-il capable de généraliser son raisonnement de façon abstraite ? Un nouveau paradigme de programmation à savoir la programmation orientée objet pose-t-il problème dans une démarche de transposition didactique d'un concept mathématique ?

L'étudiant est amené à créer une classe equa_diff permettant d'instancier des objets de types équations différentielles à coefficients réelles de la forme $ay' + by + c = 0$. La relation de récurrence présentée est celle utilisé dans les cours de physique.

RESULTATS ET INTERPRETATIONS

Nous présentons dans ce qui suit un tableau récapitulant les résultats des copies par question des étudiants suivi par un histogramme illustrant la répartition des réponses.

Tableau 1

Résultats des élèves par question

	Déclaration de la classe et son constructeur	Méthode de représentation (affichage)	Evaluation à la main			Implémentation de la méthode d'Euler
			A Exprimer l'équation	B Exprimer la dérivée	C Evaluation à la main de la relation de récurrence	
Nombre d'étudiants ayant répondus correctement	Correct 59 Faux 1	Correct 54 Partiellement correct 1 Faux 5	Correct 55 Faux 5	Correct 48 Faux 7 Absent 5	Correct 36 Partiellement Correct 6 Faux 12 Absent 6	Correct 11 Faux 3 Absente 46

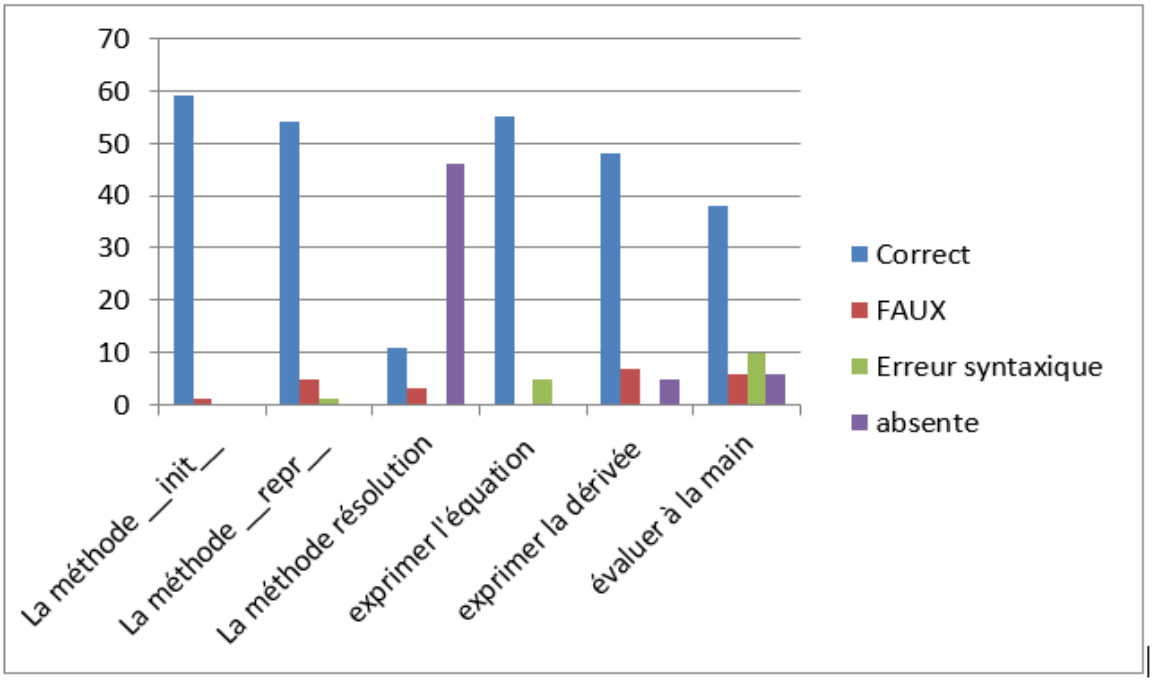


Figure 3. Histogramme des réponses.

Interprétations

- 10 % des étudiants ont répondu correctement sur la totalité, une maîtrise du niveau sémantique et syntaxique, l'étudiant parvient donc à trouver la solution numérique afin de la tracer graphiquement but de l'exercice. L'étudiant a compris qu'il ne s'agit pas d'une solution formelle mais d'une solution numérique/graphique, l'appel de ces deux registres (symbolique et graphique) au sens de Duval semble être utilisé avec une distinction claire.

L'implémentation de la méthode de résolution dans une approche objet, relève d'une maîtrise du concept d'encapsulation lié à la programmation objet,

- 18,33 % des étudiants ont tenté d'implémenter la méthode de résolution sans parvenir à le faire correctement, on y retrouve également des erreurs d'interprétation de la relation de récurrence, pour ce type d'étudiants, le but est clair mais ils ne parviennent pas à traduire en terme de code la solution mathématique, il y a passage de la relation de récurrence physique d'un point de vue syntaxique à sa représentation mathématique comme le montre la figure 4 avec incapacité du passage à la programmation. Le passage d'un modèle de représentation à un autre pose problème.

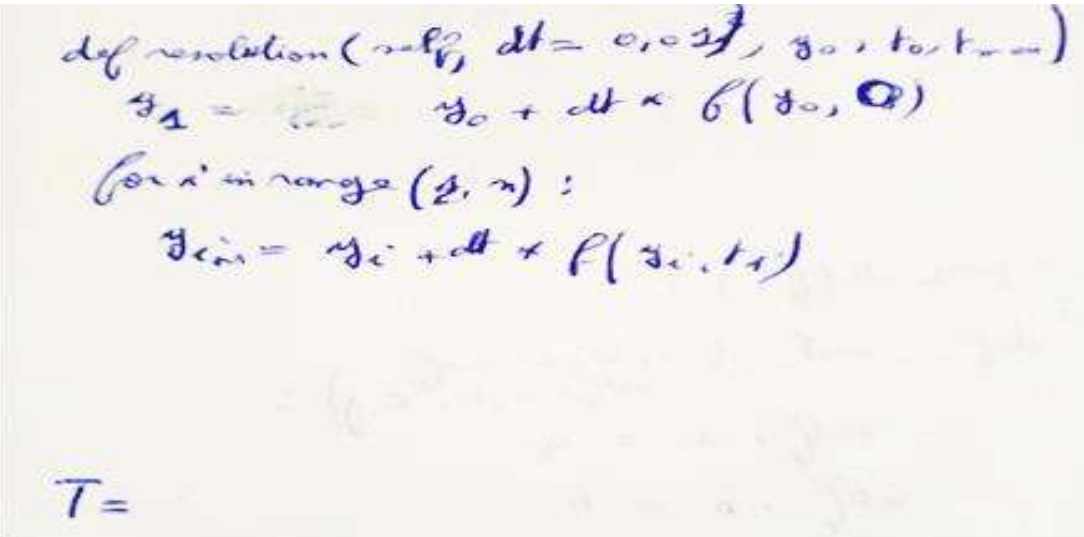


Figure 4

Pour cette catégorie l'absence de la programmation objet dénote d'une difficulté de la part des étudiants à repenser la solution dans un nouveau paradigme de programmation, ils optent alors à faire appel à leurs connaissances vues en première année, un schème existant, pour faire une simple reproduction de code dans une approche procédurale.

Le schème algorithme est présent, un déjà là vu en première année, l'étudiant connait comment il faut faire mais il a du mal à repenser ce schème dans un nouveau contexte de programmation, le changement de style pose problème, l'étudiant agit alors par évitement d'obstacle didactique auquel il est confronté et fait appel à des schèmes et agit en acte, il cherche il tente de résoudre mais ceci reste insuffisant pour atteindre une solution correcte, dans un sens didactique il procède par essai/erreur.

On pourrait conclure qu'appréhender une méthode de résolution dans un nouveau style de programmation n'est pas une tâche facile pour l'étudiant, car elle fait appel à plusieurs connaissances au même temps.

Il s'agit d'une double difficulté de représentation syntaxique des écritures algébriques en mathématiques définies par la relation de récurrence, une difficulté pour exprimer l'itération syntaxiquement et de l'exprimer encore plus dans une syntaxe objet

Nous pourrons conclure alors que pour ce groupe d'étudiants il y a assimilation de la relation de récurrence en écrivant correctement les évaluations numériques à la main, mais que la programmation objet pose problème d'un point de vue sémantique et syntaxique. Nous rajoutons aussi que les problèmes liés à la notion de variable dans un processus dynamique est un problème persistant.

- 58,33% Un phénomène frappant, les copies présentent une majeure partie correcte à savoir la déclaration de la classe equa_diff, son constructeur __init__ (question 1), la méthode spéciale __repr__ (question 2), un blocage sur la méthode de résolution mais une trace à la main effectuée de façon correcte, ceci montre que le registre mathématique est très présent pour ces étudiants,

l'outil mathématique est facile à appréhender pour ces filières scientifiques, ce qui dénote d'une compréhension de la méthode de résolution d'un point de vue sémantiques, mais d'une incapacité à passer à son implémentation (Kouki, 2018)(voir figure 5.1 et 5.2)

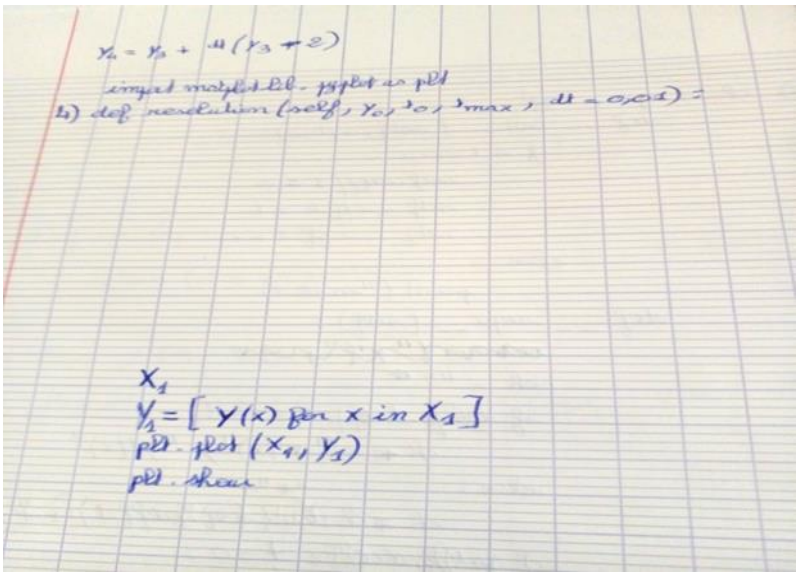


Figure 5.1.

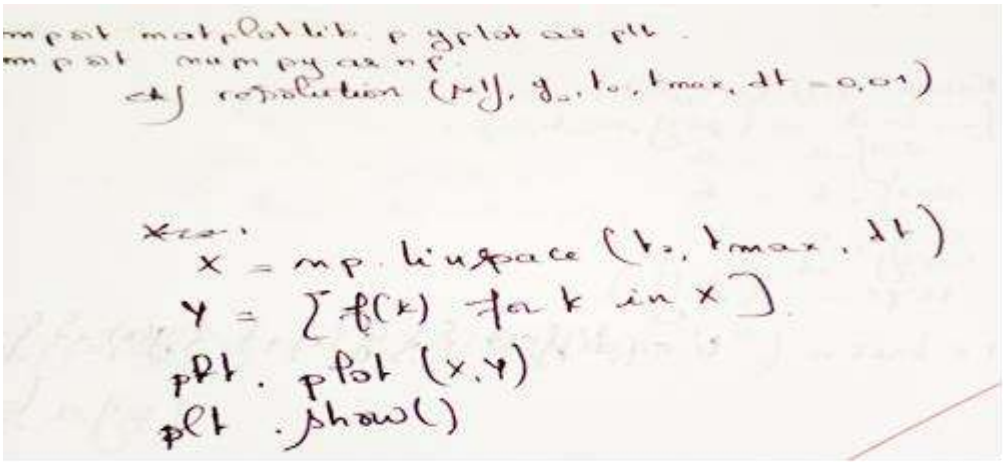


Figure 5.2.

Nous pouvons dégager les constatations suivantes :

1. Un blocage par rapport à la programmation objet, l'étudiant a du mal à implémenter la méthode de résolution, c'est cet aspect générique de la POO qui lui pose problème, la fonction résolution fait appel également à la notion de dérivée une fonction locale à déterminer qu'il n'arrive pas à voir de façon claire.
2. L'étudiant fait appel à ses connaissance antérieures et dans ce cadre de l'évaluation l'énoncé est différent de celui posé en premier année, ceci indique encore une fois que les étudiants de façon générale essayent d'employer les mêmes schèmes pour un problème visiblement le même mais syntaxiquement posé différemment, il est à la tache de l'étudiant de restructurer les schèmes existants, les restructurer dans un nouveau cadre pour répondre à un nouveau besoin au moindre changement ils sont incapable de réadapter ou réajuster leurs schèmes en fonction de nouveaux besoins.
3. Blocage par rapport à la relation de récurrence même, telle que présentée, les étudiants perçoivent cette relation de récurrence du domaine du continu, ils ont eu du mal à discrétiser, malgré qu'en mathématique ils ont aucun problème avec (expliqué par un certain nombre d'étudiants), les étudiants associent du sens à chaque représentation de la relation de récurrence.
4. Une Insuffisance de temps (expliquée par certains étudiants) donc probablement une mauvaise estimation du temps alloué à l'évaluation de notre part.

D'un point de vue sémantique, nous pouvons dire qu'il y a :

1. Une mauvaise assimilation de la notion d'algorithme numérique, car ce sont des algorithmes ou l'on est basé sur la discrétisation d'un intervalle, et que la notion de continu n'existe ni dans la notion d'algorithme numérique, ni même dans la représentation des réels sur ordinateurs, un concept déjà enseigné en première année. Le dt du schéma numérique est perçu comme une continuité dans le temps, influencée par la syntaxe du domaine de la physique sur l'interprétation des relations chez les étudiants, toutefois certains élèves ont su passer du premier schéma au deuxième après réflexion.

2. D'une interférence de connaissance tant-dis que le premier schéma numérique fait appel à un déjà là vu en sciences physiques le deuxième schéma numérique rappelle la notion de série numérique vue en maths.
3. Le dt rappelle les notations du domaine de la physique c'est la raison pour laquelle ils interprètent ceci comme du domaine du continu, nous parlons ici du passage de registres mathématique à celui de la physique.
4. Les résultats ont démontré que le paradigme objet constitue un obstacle pour les étudiants, le passage d'un paradigme procédural à un paradigme objet impose des contraintes aussi bien dans la conception du programme (comment appréhender la résolution) que dans la syntaxe du langage (nouvel syntaxe associée à l'objet).

- 16,66 % : c'est la catégorie des étudiants qui ont eu du mal à interpréter la relation de récurrence, ils n'ont pas saisi la méthode de résolution, même avec le rappel de la méthode, ils ont été incapable de reproduire à la main les premières itérations, ceci révèle d'une mauvaise assimilation la relation de récurrence donc ils ne s'y accommodent pas et donc un problème de source mathématique principalement, on retrouve encore une fois un problème entre représentation informatique et représentation mathématique. (figure 6)

Les étudiants ont utilisé des variables indéterminées malgré qu'une substitution numérique est demandée, le registre mathématique est très présent encore une fois l'interférence entre les disciplines pose un problème aussi bien d'un point de vue syntaxique que sémantique.

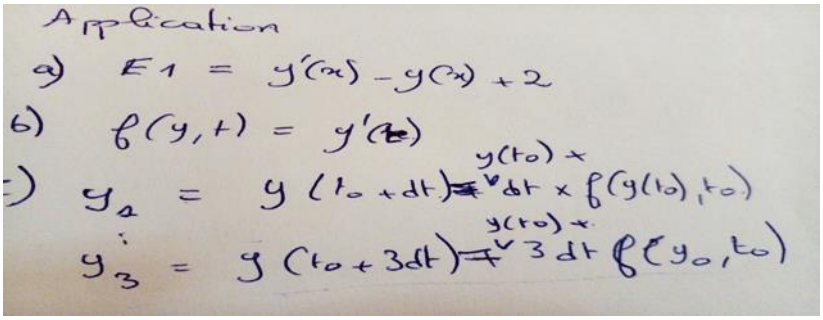


Figure 6.

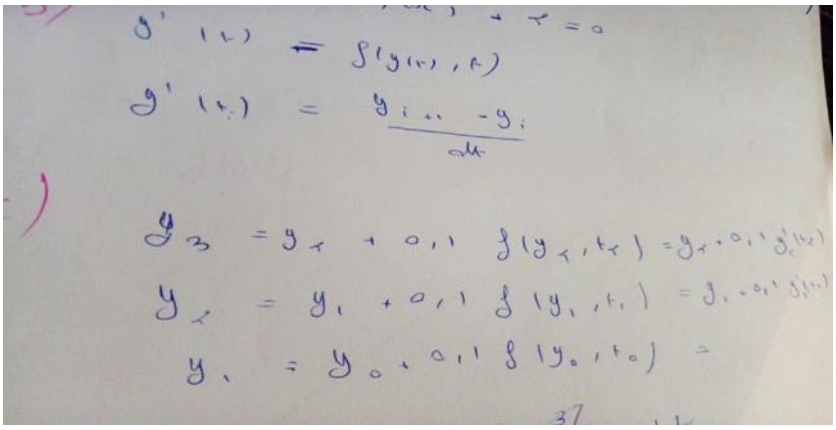


Figure 7.

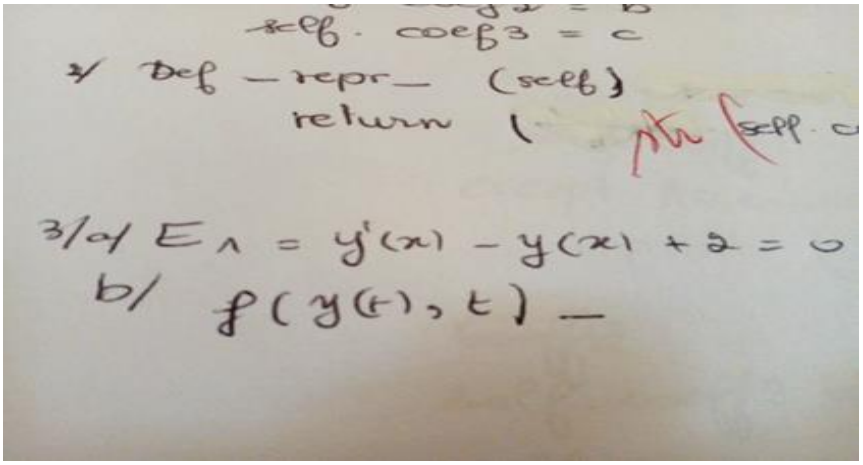


Figure 8.

Une application numérique demandée est absente (figure 7 & 8), l'étudiant ne parvient pas à présenter les premières étapes de la récurrence, il n'a probablement pas saisi la solution mathématique.

CONCLUSION

Ce travail de recherche a conduit à une panoplie d'obstacles didactiques dégagés dans cette étude que nous pourrons désormais classer dans différents points de vue.

D'abord nous avons dégagé des obstacles de type sémiotique, lié à l'interprétation de la relation de récurrence présentée avec une syntaxe proposée dans le domaine

de la physique. Pour la relation de récurrence proposée avec le « dt » ceci a dénoté d'un problème d'interprétation particulier, l'étudiant voit ce « dt » comme un problème continu et oublie qu'une résolution numérique est avant tout un problème de discrétisation, il est donc incapable de passer à la programmation suite à une représentation sémantique dont il a fait d'une représentation syntaxique.

Toujours d'un point de vue sémiotique le changement du paradigme de programmation pose en lui-même un problème crucial sur deux plans: d'une part d'un point de vue syntaxique, la syntaxe orienté objet semble difficile à appréhender pour les novices, d'autre part, d'un point de vue sémantique, l'aspect conceptuel de cette programmation ne semble pas être évident pour les étudiants, l'idée que l'équation devient un objet d'étude ne semble pas être facile à assimiler, le passage à une conceptualisation dans un aspect d'abstraction donc métacognitif est difficile pour les étudiants novices en matière de programmation orientée objet POO.

Sur le plan institutionnel nous pouvons dégager des obstacles organisationnel et institutionnel « La situation contemporaine des universités atteste qu'elles ont perdus toute vocation d'universalité. Elles ont cessé d'être des communautés culturelles où se noue l'alliance entre divers horizons de la connaissance, elle sont devenues plutôt des prisons centrales de la culture …avec le seul souci de faire valoir son petit domaine à l'abri …les universités ce sont fragmentés en facultés, puis en départements, en instituts dont le superbe isolement exclut l'esprit de dialogue » (Darbelley, 2011) personne ne parle à personne et d'ailleurs personne ne comprend plus personne, un émiettement de la connaissance s'est bien installé ! la spécialisation des disciplines représente donc un obstacle majeur pour les essais d'élaboration de connaissance interdisciplinaires, le professeur de mathématiques a du mal avec la notion d'algorithmique, le professeur d'informatique lui a du mal avec les phénomènes physiques, les professeurs ne font plus le travail cognitif nécessaire afin de toucher à d'autre disciplines à part celles qu'ils maîtrisent, l'étudiant lui il est au carrefour de ces enseignants en rupture de collaboration interdisciplinaire. De plus, l'émiettement du savoir défini ci-haut dans les obstacles organisationnels a construit à l'origine de l'histoire de la fragmentation des sciences, obstacle épistémologique,

institutionnellement des rapports hiérarchiques et cloisonnés entre les disciplines académiques se sont installés, ils sont donc d'origine épistémique où des champs scientifiques ont été découpé en disciplines et sous-disciplines.

Enfin, la démarche d'une formation spécialisée des professeurs engendre une hétérogénéité de profils spécialisés renforçant le morcellement institutionnel des connaissances sur l'homme et le social, dans ce cadre l'organisation institutionnelle soit pénalise le développement des réflexions interdisciplinaires sous prétexte de l'excellence disciplinaire comme les mathématiques par exemples dans les filières scientifiques, qui pour les spécialistes de ce domaine reste une excellence disciplinaire négligeant ainsi tout apport supplémentaire de tout autre discipline telle que l'informatique à cette dernière, l'étudiant hérite ainsi ce même comportement psychologique de son professeur de mathématique ou de physique ce qui fait surgir un autre type d'obstacle d'ordre psychologique.

RÉFÉRENCES

Daniel, M. (2021, Fevier). *Algorithme numérique.* Récupéré sur Ecole Polytech Marseille: https://marc-daniel.pedaweb.univ-amu.fr/Algonum/Algonum2diap.pdf

Darbelley, F. (2011, Octobre). Vers une théorie de l'interdisciplinarité ? entre unité et diversité. *Nouvelles perspectives en sciences sociales, 7*(1), pp. 65-87.

Duval. (1993). Registres et représnettaions sémiotiques et fonctionnement cognitif de la pensée. *Annales de didactique et de sciences cognitifs*, 37-65.

Julien, A. (2019-2020). *TP : Résolution d'équations différentielles : méthode d'Euler*. Récupéré sur http://desaintar.free.fr/python/tp/tp_euler.pdf: http://desaintar.free.fr/python/tp/tp_euler.pdf

Kirkerud.B. (1996). "Use- Read- Change" : a new introductioary course in informatics . *Norsk Informatik Konferense*, (pp. 173-180).

Kouki, R., & Griffiths, B. (2021). Semiotic Aspects of Differential Equations: Analytical and Graphical Competency in the USA and Tunisia. *African Journal of Research in Mathematics, Science and Technology Education*, 174-184.

Kouki (2018) L'articultaion des dimensions syntaxiques et sémantiques en algèbre du secondaire. Dans *recherche en didactiques des mathématiques* (pp. 43-78). la Pensée sauvage.

Reinfelds.J. (1995). Logic in first courses for computer science majors. *World Conference on Computers in Education VI: liberating the learner*, (pp. 467-477).

Rogalski, & Lagrange. (2017). *Savoirs, concepts et situations dans les premiers apprentissages en programmation et en algorithmique.* Récupéré sur https://hal.archives-ouvertes.fr: https://hal.archives-ouvertes.fr/hal-01740442/document

Samurçay, & Rogalski. (1991). Acquisition of programming knowledge and skills. Dans *Psychology of programming.* Elsevier.

T.Phit-Huan, Siew-Woei, L., & Choo-Yee, T. (2009). Learning difficulties in programming courses: Undergraduates perspective and perception. *International Conference on Computer Technology and Development*, (pp. vol.2, pp. 42–46).

Tarski. (1923-1944). *Logique, sémantique, mathématique.* Arman Colin, vol1, 1972.

Vergnaud. (1989). La théorie des champs conceptuels. *Publications de l'Institut de recherche mathématiques de Rennes*, (pp. 47-50). Rennes.

Vergnaud. (1990). La théorie des champs conceptuels,. *Recherches en Didactique des Mathématiques,, La Pensée Sauvage*, pp. vol.10 n°2-3, pp.133-170, éd.

Zamzouri, & Rosminah. (2012). Difficulties in learning programming : Views of students. *1st International Conference oo Current Issues in Education ICCIE.* Yogyakarta, Indonesia.