

Aider à la compréhension des fonctions avec un logiciel : les fonctions définies par un algorithme

Roselyne Halbert,
Jean-baptiste Lagrange,
Christine Le Bihan,
Bernard Le Feuvre,
Marie-Catherine Manens
Groupe Casyopée IREM de Rennes

Dès 2009, les objectifs pour le lycée précisait « l'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes posés doivent être en relation avec les autres parties du programme... » Ceci est rappelé par une circulaire de 2017 concernant des aménagements au programme de mathématiques de seconde générale et technologique (MEN 2017a). Cet article s'intéresse particulièrement à la place que l'algorithmique peut avoir dans le domaine des fonctions et vise à apporter des réponses à la question suivante :

Quel travail sur des algorithmes par les élèves pour contribuer à donner du sens aux notions liées aux fonctions et aider à leur compréhension ?

Les réponses que nous pensons apporter s'inscrivent dans l'activité d'un groupe de l'IREM de Rennes depuis une quinzaine d'années. Nous avons conçu, mis en œuvre et évalué des situations visant à donner du sens aux fonctions pour les élèves. Alors que l'algorithmique n'était pas dans les programmes du lycée, notre approche a été du type TICE : utilisation du calcul formel et de la géométrie dynamique ; nous avons conçu et expérimenté pour cela un logiciel permettant aux élèves de faire interagir ces deux outils (Halbert, Lagrange, Le Bihan et al. 2013). Nous allons en premier lieu revenir sur les enjeux d'apprentissage dans ce champ à partir de notre expérience. Ceci nous conduira à une hypothèse concernant l'apport de l'algorithmique dans ce champ. Nous nous interrogerons ensuite sur les choix possibles dans le domaine de l'algorithmique au lycée, et préciserons les nôtres. Ensuite nous présenterons un exemple de situation en classe à partir de ces choix avant de conclure.

Les fonctions au lycée

Le programme de Seconde de 2009 a introduit un champ « fonctions » et le document ressource associé (MEN 2009) précise d'abord qu'un objectif essentiel du travail amorcé au collège consiste à faire émerger progressivement « un processus faisant correspondre à un nombre un autre nombre ». Le document souligne aussi que pour beaucoup d'élèves, la notion de fonction ne fait pas encore sens en début de Seconde et qu'il importe que les élèves soient « confrontés à des situations dans lesquelles il y ait besoin d'identifier deux quantités qui varient tout en étant liées ».

Le document ressource rappelle aussi les différentes représentations « du lien entre quantités (tableau, courbe, formule...) » et la nécessité « d'identifier les avantages et les inconvénients de telle ou telle selon la question initialement posée ».

Cette présentation des fonctions comme lien entre quantités et les représentations proposées montrent que ce sont les fonctions de \mathcal{R} (ou d'un intervalle) vers \mathcal{R} qui constituent le champ à

étudier. Ceci est confirmé par le fait que l'étude des transformations géométriques, abordée au collège n'est pas prolongée au lycée par une approche fonctionnelle. De même les suites ne sont pas présentées comme des fonctions de \mathbb{N} vers \mathcal{R} . La lecture des programmes avec l'introduction de fonctions de références puis de méthodes d'étude calculatoires dans la partie analyse en Première et Terminale, est une autre confirmation. Mise à part l'observation qu'à une valeur de la première quantité correspond une seule valeur de la seconde et l'acquisition du vocabulaire associé (« image », « antécédent »), une compréhension plus générale comme correspondance entre ensembles n'est pas visée ; ainsi la composition des fonctions n'est pas abordée explicitement, seul le formalisme $f(u(x))$ est utilisé. Par ailleurs, bien que le programme précise qu'une fonction peut être définie à partir d'une table ou d'un graphe, la définition à partir d'une formule reste dominante, les autres représentations étant obtenues le plus souvent à partir de cette définition.

Dans (Halbert, Lagrange, Le Bihan et al. 2013, p.15) nous avons identifié deux aspects des fonctions, cohérents avec les enjeux d'apprentissage soulignés par le document : l'aspect « correspondance » renvoie à une relation isolée entre une valeur d'une première quantité et une valeur d'une seconde quantité ; l'aspect « dépendance » suppose deux quantités variables liées dans un système observable : une action sur la première quantité modifie la seconde. Nous avons aussi montré comment les représentations des fonctions se comprennent sous ces deux aspects. Nous avons identifié comme un objectif important pour les élèves de comprendre ce qu'est une dépendance fonctionnelle, comment elle s'exprime dans différentes représentations et les traitements que ces représentations permettent.

Nous avons étudié cet objectif dans le cadre de modélisations géométriques. Prenons l'exemple de la distance d'un point fixe A à une parabole dans un repère dont l'axe des ordonnées est l'axe de symétrie, le point A n'étant pas sur cet axe (Halbert, Lagrange, Le Bihan et al. 2013, p. 13). Le premier enjeu est la compréhension d'une correspondance entre la position d'un point M mobile sur la parabole et la distance AM de ce point au point fixe. Un logiciel de géométrie dynamique permet d'observer des valeurs de cette distance lorsqu'on bouge le point M sur la parabole. La correspondance n'est pas complète, car le logiciel donne le résultat, mais la donnée en entrée reste une entité non numérique (la position d'un point). La position d'un point se définissant par deux coordonnées l'abscisse et l'ordonnée, le choix de l'abscisse est guidé par la conception « ensembliste » : à une valeur de l'ordonnée correspondent deux positions et donc deux valeurs de la distance. Ce choix étant fait, une correspondance a été établie : une abscisse étant donnée, il est possible de positionner un point sur la parabole à cette abscisse et d'observer la valeur « correspondante » de la distance. Rechercher la distance minimum implique de passer à une dépendance « dynamique », ce que permet le mouvement « en continu » du point M dans la géométrie dynamique. Ainsi, une fonction est définie et le problème de la plus courte distance peut être résolu, de façon approximative, par observation de la dépendance. L'expression de cette fonction à l'aide d'une courbe permet une observation plus précise. Son expression par une formule permet un traitement algébrique « exact » et établit le lien avec les fonctions « habituelles » données par une formule.

Ce rappel nous permet de situer notre objectif. Nous nous situons dans le cadre des fonctions de \mathcal{R} (ou d'un intervalle) vers \mathcal{R} qui sont principalement étudiées au lycée. Nous souhaitons que ces fonctions soient perçues par les élèves comme des entités non réductibles aux représentations, (formule, courbe, table...) qui dominent dans l'enseignement, et que ces

représentations soient perçues comme des moyens de les étudier plutôt que comme une fin en soi. Nous nous intéressons donc aux modes de génération des fonctions indépendants d'une représentation, les relations de dépendance entre quantités comme celle que nous venons d'évoquer en étant un exemple. Nous nous intéressons ici à un mode de génération permis par l'introduction de l'algorithmique au lycée : les fonctions définies par un algorithme.

Algorithmes et fonctions

Nous venons de dire que l'algorithmique est pour nous une opportunité d'élargir les possibilités de génération des fonctions et donc une compréhension des fonctions par les élèves qui ne se réduise pas aux représentations et à leur utilisation. Ceci converge avec des préoccupations exprimées dans un récent document ressource « algorithmique et programmation » (MEN 2017b).

« Les notions mathématique et informatique de fonction relèvent du même concept universel. En informatique, une fonction prend un ou plusieurs arguments et renvoie une valeur issue d'un calcul. »

Définir des fonctions par un algorithme permet donc aussi une compréhension d'un concept informatique.

« Il est important de distinguer très clairement la conception des algorithmes et leur utilisation. Les élèves écriront des fonctions qui pourront être enregistrées dans des fichiers, appelés scripts ou modules (...) L'exécution d'un module comportant la définition d'une fonction f ne produit aucun affichage particulier (...) on peut (...) dans la console, faire des appels du type $f(2)$ et la console affiche la valeur renvoyée par la fonction. »

Nous retenons qu'un algorithme produit un objet fonction, cet objet s'ajoutant aux objets déjà existants dans l'environnement de travail. L'objet « fonction » contient en lui-même les idées de donnée, de résultat et éventuellement de paramètres dans le cas d'une famille de fonctions, et donc les instructions d'entrée-sortie (lecture d'une donnée, affichage d'un résultat) ne sont pas pertinentes.

Ensuite, la rédaction du document oriente vers un environnement spécifique à la programmation (Python). Nous divergeons sur ce point car nous souhaitons que les fonctions définies par un algorithme existent pour les élèves dans un environnement logiciel qu'ils utilisent de façon habituelle pour ce domaine. Puis, parmi les exemples proposés dans le document, seule la définition par morceaux à l'aide d'une structure alternative (si... alors...sinon) définit une fonction de \mathcal{R} (ou d'un intervalle) vers \mathcal{R} . Les autres exemples, qui mettent en jeu l'itération, un concept central en algorithmique, concernent des situations hors du champ des fonctions et déjà pointées par les programmes comme pouvant faire l'objet de travaux en algorithmique. Les principes énoncés dans le document se traduisent par le choix d'utiliser ces situations pour promouvoir une programmation privilégiant l'usage de fonctions informatiques, ce que nous appelons dans la suite un style de programmation « fonctionnel ». Ce choix impliquerait pour les élèves, un élargissement très important du champ des fonctions, tant en ce qui concerne les objets en jeu qui peuvent être toute entité représentable informatiquement, qu'en ce qui concerne l'usage notamment de la composition et de la récursivité¹.

¹ Par exemple, implémenter l'algorithme d'Euclide de façon fonctionnelle conduit à définir une fonction PGCD de \mathbb{Z}^2 vers \mathbb{N} . Cette définition prend son sens dans le cas où la fonction PGCD est appelée par une autre

Notre choix est différent : il est important pour nous que les élèves reconnaissent les fonctions définies par un algorithme comme similaires aux fonctions « habituelles » de \mathcal{R} (ou d'un intervalle) vers \mathcal{R} . C'est pourquoi les situations auxquelles nous nous intéressons concernent ce type de fonctions. C'est aussi pourquoi nous souhaitons que ces fonctions définies par un algorithme existent pour les élèves dans un environnement logiciel qu'ils utilisent aussi de façon habituelle pour des travaux sur des fonctions. Nous faisons l'hypothèse qu'il est possible d'enrichir la compréhension des fonctions par les élèves, en cohérence avec les points de convergence repérés ci-dessus dans le document ressource, tout en se limitant à un environnement de programmation directement accessible aux élèves et à un type de fonctions proche des fonctions usuelles au lycée, sans engager les élèves vers un style de programmation « fonctionnel ».

Nous pensons aussi que, alors que algorithmique et programmation n'ont que très récemment fait leur entrée au collège, des difficultés spécifiques sont à prévoir pour beaucoup d'élèves (Lagrange & Rogalski 2017 proposent une analyse de ces difficultés). Nous souhaitons par conséquent limiter le travail en algorithmique à une structure algorithmique simple.

Un exemple

Nous le prenons dans le baccalauréat 2018, sujet section S de Pondichery, exercice 1. Cet exercice va nous servir à montrer, à partir d'éléments de situation estimés intéressants par l'institution lycée, ce que les choix qui viennent d'être énoncés nous conduisent à proposer.

Le sujet propose deux modèles de décroissance de la température dans un four. En partie A, le sujet introduit la suite T_n , température en degré Celsius du four au bout de n heures écoulées et la définit par l'algorithme suivant :

```
T ← 1000
Pour i allant de 1 à n
    T ← 0,82 × T + 3,6
Fin Pour
```

L'algorithme est présenté de façon cohérente avec les principes ci-dessus : il définit un objet mathématique, ici une suite, sans mettre en œuvre d'instructions de lecture ou d'écriture. La structure algorithmique est l'itération, dont la compréhension est un objectif important au lycée.

En partie B, le temps « continu » est considéré comme la variable, et donc le modèle de décroissance n'est plus une suite, mais une fonction définie sur les réels positifs. A la question

fonction (par exemple une fonction PPMC rendant le quotient du produit de deux relatifs non nuls par leur PGCD) ou par elle-même dans une définition récursive. Certes le style de programmation « fonctionnel ». n'est pas de la « programmation fonctionnelle » au sens strict et Python n'est pas un « langage fonctionnel » pur, mais l'utilisation proposée dans le document ressource impliquerait de nombreuses ruptures par rapport à la programmation « impérative » dominante jusqu'ici au lycée : appel de fonctions en mode « console », notion d'argument....

1, on admet une relation entre la fonction et sa dérivée $f'(t) + \frac{1}{5}f(t) = 4$ et à la question 2 on

donne la formule : $f(t) = 980e^{-\frac{t}{5}} + 20$. Puis les élèves sont invités à différents calculs (variations, moyenne, limites) relevant de l'analyse « calculatoire » du lycée qui met en jeu principalement la représentation « formule ».

Malgré l'intérêt que présentent les éléments de situation proposés par le sujet, notre analyse est que les tâches demandées aux élèves ne mettent pas réellement à contribution l'algorithmique pour une compréhension des fonctions. En effet, il n'y a pas de lien compréhensible par les élèves entre la suite que définit l'algorithme de la partie A et la fonction de décroissance de la partie B.

Quelles seraient les conditions qui permettraient de faire un lien plus direct entre un algorithme et la fonction de décroissance la partie B ? Pour nous ces conditions seraient que l'algorithme définisse une fonction. Supposons qu'une observation empirique ait établi une relation entre les températures aux heures entières telle qu'exprimée dans l'algorithme. La fonction affine par morceaux g prenant les valeurs données par l'algorithme pour les valeurs entières de la variable est une première approximation de la fonction de décroissance. Pour les valeurs entières n , elle vérifie la relation $g'_d(x) + 0,18 \cdot g(x) = 3,6$. Pour une meilleure approximation, définissons une fonction affine par morceaux h prenant pour les valeurs entières ou $\frac{1}{2}$, les valeurs données par l'algorithme suivant, construit sur un modèle de décroissance géométrique et cohérent avec g pour les valeurs entières :

T ← 1000

Pour i allant de 1 à $2n$

$$T \leftarrow \sqrt{0,82} \times T + 20(1 - \sqrt{0,82})$$

Fin Pour

Pour les valeurs de x entières ou $1/2$, cette fonction vérifie la relation

$$h'_d(x) + h(x) \frac{1 - \sqrt{0,82}}{2} = 10 \cdot (1 - \sqrt{0,82}) \quad \text{ou de façon approchée } h'_d(x) + 0,19 \cdot h(x) = 3,8.$$

Ensuite, de façon formelle² ou intuitive, il est possible d'introduire la fonction f dérivable vérifiant la relation $f'(x) + 0,2 \cdot f(x) = 4$.

Notre hypothèse est qu'une approche de ce type, liant algorithme et fonction, permet aux élèves d'enrichir leur compréhension des fonctions, en considérant des fonctions différentes des fonctions « habituelles » définies par une formule, et en mettant en relation ces deux types

² Pour un pas $1/p$, le corps de boucle de l'algorithme est

$$T \leftarrow \sqrt[p]{0,82} \cdot T + 200(1 - \sqrt[p]{0,82}). \quad \text{L'algorithme définit une fonction } k \text{ vérifiant la relation } k'_d(x) + k(x) \cdot p(1 - \sqrt[p]{0,82}) = 20 \cdot p(1 - \sqrt[p]{0,82}).$$

En passant à la limite on obtient la relation $f'(x) - f(x) \cdot \ln(0,82) = -20 \cdot \ln(0,82)$ peu différente de la relation donnée sans démonstration par l'énoncé.

de fonctions. Nous allons donner plus loin un exemple de mise en œuvre, après avoir précisé la nature ces fonctions définies par un algorithme ainsi que des choix possibles quant au logiciel permettant de les traiter.

Les fonctions définies par un algorithme

L'exemple qui précède nous permet d'introduire un type d'objet qui nous semble répondre à nos choix énoncés ci-dessus, tout en restant dans le panel de situations étudiées au lycée. Il s'agit de fonctions affines par morceaux définies sur un intervalle de \mathcal{R} . Un processus itératif permettant de calculer des couples (x_i, y_i) , la suite des x_i étant croissante, on définit une fonction qui prend la valeur y_i pour chaque valeur de x_i et est affine sur chaque intervalle $[x_i, x_{i+1}]$. La représentation graphique est une ligne brisée (figure 1 pour les fonctions g et h de l'exemple précédent).

1: $x \leftarrow 0$ 2: $y \leftarrow 1000$ 3: Tant Que $x \leq 100$ 4: $x \leftarrow x+1$ 5: $y \leftarrow 0.82 \cdot y + 3.6$ 6: Fin Tant Que	1: $x \leftarrow 0$ 2: $y \leftarrow 1000$ 3: Tant Que $x \leq 100$ 4: $x \leftarrow x + \frac{1}{2}$ 5: $y \leftarrow \sqrt{0.82} \cdot y + 20 \cdot (1 - \sqrt{0.82})$ 6: Fin Tant Que	
L'algorithme définissant la fonction g	L'algorithme définissant la fonction h	Représentation graphique de g (noir) et h (rouge) sur l'intervalle $[23 ; 24]$

Figure 1.

Ces fonctions existent en mathématiques. Il peut s'agir de fonctions d'interpolation comme dans l'exemple précédent. Il peut s'agir aussi de fonctions issues de méthodes de résolution numérique des équations différentielles comme la méthode d'Euler. L'exemple précédent montre qu'elles peuvent exister aussi dans les mathématiques du lycée et l'intérêt qu'elles présentent.

Le choix d'un logiciel

Il s'agit ici d'évaluer la possibilité qu'offrent des logiciels pour définir une fonction à partir d'un algorithme. En cohérence avec les principes présentés en introduction, nous nous limitons à une sélection de logiciels habituellement mis en œuvre en classe pour des problèmes sur les fonctions, avant de préciser notre choix.

Nous reprenons l'exemple de la fonction h ci-dessus, et nous le traitons successivement avec un tableur, avec AlgoBox et avec GeoGebra. Ces trois logiciels permettent de calculer les

couples $(x, h(x))$, x parcourant les valeurs entières et $1/2$ de l'intervalle $[0 ; 100]$ et d'afficher les points ayant ces coordonnées dans un repère.

1) Avec un tableur :

On crée deux colonnes de nombres (en recopiant deux formules un nombre de fois suffisant) puis on crée une ligne brisée.

	A	B		A	B
1	x	h(x)		x	h(x)
2					
3	0	1000		0	1000
4	0,5	923,5154		=A3+0,5	=SQRT(0,85)*B3+20*(1-SQRT(0,85))
5	1	853		=A4+0,5	=SQRT(0,85)*B4+20*(1-SQRT(0,85))
6	1,5	787,9881		=A5+0,5	=SQRT(0,85)*B5+20*(1-SQRT(0,85))
7	2	728,05		=A6+0,5	=SQRT(0,85)*B6+20*(1-SQRT(0,85))
8	2,5	672,7898		=A7+0,5	=SQRT(0,85)*B7+20*(1-SQRT(0,85))
9	3	621,8425		=A8+0,5	=SQRT(0,85)*B8+20*(1-SQRT(0,85))
10	3,5	574,8714		=A9+0,5	=SQRT(0,85)*B9+20*(1-SQRT(0,85))

2) Avec AlgoBox :

On crée par algorithme les valeurs successives des termes des deux suites. On utilise des instructions « graphiques » pour afficher à chaque itération un point, ce qui revient à une instruction de sortie. On peut aussi tracer une ligne brisée, mais le travail devient plus technique.

```

1 VARIABLES
2 x EST_DU_TYPE NOMBRE
3 y EST_DU_TYPE NOMBRE
4 h EST_DU_TYPE NOMBRE
5 DEBUT_ALGORITHME
6 LIRE h
7 x PREND_LA_VALEUR 0
8 y PREND_LA_VALEUR 1000
9 10 TANT_QUE (x<=100) FAIRE
10 DEBUT_TANT_QUE
11 y PREND_LA_VALEUR y*pow(0.85,1/2)+20*(1- pow(0.85,1/2))
12 x PREND_LA_VALEUR x+0,5
13 TRACERPOINT (x,y)
14 FIN_TANT_QUE
15 FIN_ALGORITHME

```

3) Avec GeoGebra

La commande $L=IterationList((x(C)+0.5, SQRT(0,85)*y(C)+20*(1-SQRT(0,85)),C,\{(0,1000)\},20)$ produit les couples $(x, h(x))$ et les points correspondants sur l'écran de géométrie. Ce type de programmation est très éloigné des pratiques en algorithmique des élèves de lycée.

4) Notre choix

Chacun des trois logiciels permet une certaine représentation graphique de la fonction h , et éventuellement une comparaison graphique avec la fonction f modélisant la décroissance, mais cette représentation n'est pas reconnue par le logiciel comme représentative d'une fonction ; seuls existent des points ou des segments dans un repère. Pour nous, c'est une limitation compte-tenu de notre objectif d'élargir le champ des fonctions et la compréhension des élèves. Nous souhaitons pour cela que les fonctions « définies par un algorithme » existent dans le logiciel au même titre que les autres fonctions « définies par une formule ». Dans l'exemple ci-dessus, on doit pouvoir faire calculer au logiciel une valeur de $h(x)$ pour toute valeur de x dans l'intervalle et pas seulement pour les valeurs entières. On doit pouvoir aussi définir et étudier la fonction $f-h$.

De plus chacun des trois logiciels a des limitations spécifiques. Dans de nombreuses situations, comme nous le verrons plus loin, la fonction dépend d'un paramètre. Par exemple ce paramètre peut être le pas avec lequel évolue la variable x . Dans le cas du tableur, le nombre d'itérations dépendant du paramètre, il faut donc modifier le nombre de lignes pour chaque valeur du pas. Dans Algobox il faut passer par une instruction d'entrée que, comme les instructions de sortie, nous souhaitons éviter. Avec GeoGebra, on peut utiliser un curseur mais, nous l'avons vu, ce type de programmation n'est pas habituel pour les élèves.

Devant ces limitations, notre choix a été de faire évoluer le logiciel Casyopée. Nous avons en effet développé ce logiciel pour mettre en place et évaluer des situations d'apprentissage sur les fonctions, par exemple la situation de la distance à une parabole présentée plus haut. Il était donc naturel, à partir de nos objectifs, de créer un module de programmation, et une nouvelle entrée « créer fonction par un algorithme » à côté de l'entrée existante « créer fonction par une formule ».

Pour créer une telle fonction, un algorithme est écrit par l'utilisateur dans ce nouveau module de Casyopée de façon à faire évoluer un couple de variables (x, y) . A partir des valeurs successives de ce couple, Casyopée construit la fonction affine par morceaux et la traite ensuite comme les autres fonctions « définies par une formule ». Les paramètres de Casyopée (équivalents aux curseurs de GeoGebra) ainsi que les fonctions ou expressions déjà définies, peuvent être utilisés dans l'algorithme pour définir une famille de fonctions. Dans le module, nous avons choisi un type de programmation conforme aux pratiques au lycée avec un environnement de programmation comportant une aide à l'écriture comme dans Algobox et donc facile à maîtriser.⁴

Dans l'exemple ci-dessus, les fonctions g et h sont définies à l'aide des algorithmes de la figure et l'on obtient des représentations graphiques de fonctions affines par morceaux

⁴ Dans ce texte, nous ne nous attardons pas sur la différence de nature entre les fonctions définies par un algorithme conçues de façon mathématique, c'est-à-dire avec des valeurs exactes, et leur implémentation dans un logiciel qui va dépendre de la représentation des nombres (pseudo) réels interne à la machine sous-jacente. Nous privilégions en effet la compréhension de la relation entre algorithme itératif et fonction, l'implémentation étant un moyen de l'objectiver. Nous ne méconnaissons pas néanmoins les difficultés que cette différence peut entraîner, tant du point de vue pratique car l'itération peut cumuler les erreurs d'approximation, que du point de vue cognitif, les élèves étant peu familiers de fonctions définies « à quelque chose près ». Souhaitons que les pratiques et recherches au lycée donnent plus de place à la question de l'implémentation d'entités mathématiques dans les environnements informatiques et plus généralement aux problématiques d'approximation.

comme dans la même figure. De plus, il est possible de comparer les fonctions g et h pour des valeurs entières et non entières (figure 2 à gauche) et de les comparer graphiquement à la fonction dérivable f , en étudiant la différence (figure 3 à droite).

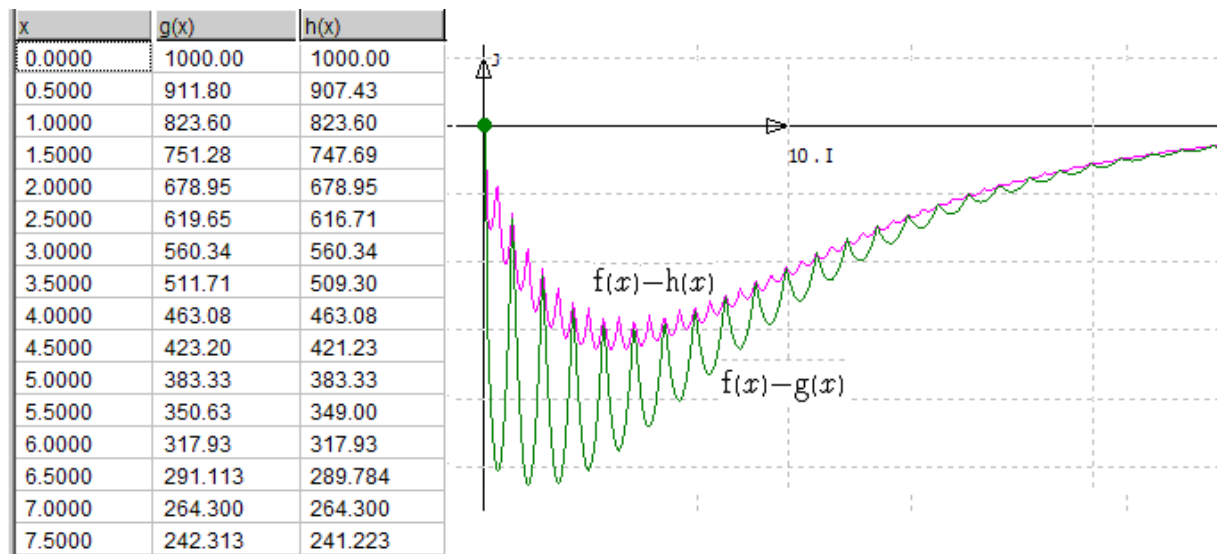


Figure 2

Un exemple de mise en œuvre en classe de Première S

La situation : construire une fonction dont on connaît la dérivée

Objectifs

La notion de nombre dérivé puis de fonction dérivée est centrale dans la partie analyse du programme de Première. Le nombre dérivé prend sens comme limite du taux d'accroissement et dans le tracé de la tangente en un point. L'objectif du travail proposé ici est la relation fonction, nombre dérivé et tangente. Il s'agit de la construction approchée d'une fonction inconnue, mais dont la fonction dérivée et la valeur en un point sont données. Le travail est d'abord géométrique : un nombre de pas et un point initial étant choisis, on construit la droite passant par le point et de pente la valeur de la dérivée en ce point et on réitère ensuite la construction en prenant un point sur la droite dont l'abscisse est celle du point précédent, augmentée de la longueur du pas. Il est ensuite algorithmique et fonctionnel : un algorithme permet de systématiser la construction pour un nombre de pas arbitraire et d'obtenir une fonction approchant la fonction inconnue d'aussi près que souhaité. Il est enfin d'ordre symbolique : il s'agit d'obtenir une expression de la fonction inconnue à partir de connaissance sur les dérivées des fonctions usuelles. Il est possible de vérifier cette expression par comparaison du graphe de la fonction ainsi obtenue avec la ligne brisée, représentation de la courbe définie par l'algorithme.

Le choix de la fonction inconnue

Nous choisissons d'étudier une fonction f dérivable sur $[1; 5]$ telle que $\forall x \in [1; 5], f'(x) = \sqrt{x}$ et $f(1) = \frac{2}{3}$. Une expression de f est $f(x) = \frac{2}{3}x\sqrt{x}$.

Cette fonction est choisie de façon à ce que la réponse ne soit pas directement déduite à partir d'une liste des dérivées de fonctions usuelles⁵. Le choix de la valeur en 1 permet d'éviter l'ajout d'une constante, question qui est abordée en Terminale dans le chapitre sur l'intégration.

Travail géométrique

Choisissons un pas de longueur 1. Construisons le point A de coordonnées $(1; \frac{2}{3})$ et la droite de pente $\sqrt{1}=1$. Cette droite est la tangente à la courbe représentant la fonction inconnue f au point A. Construisons le point B d'abscisse 2 sur cette droite. Il peut être construit comme intersection de la droite avec une parallèle à l'axe des ordonnées. Le point B n'est pas sur la courbe de la fonction inconnue f, mais le segment [AB] est tangent à la courbe. Nous traçons la droite de pente $\sqrt{2}$ passant par B puis le point C d'abscisse 3 sur cette droite. Le segment [BC] n'est pas tangent à la courbe, mais il est parallèle à la tangente au point d'abscisse 2 et en est proche. Nous itérons en construisant la droite de pente $\sqrt{3}$ passant par C, puis de même pour les points D et E respectivement d'abscisse 4 et 5. La ligne brisée ABCDE est considérée comme approchant la courbe de la fonction inconnue f (figure 3).

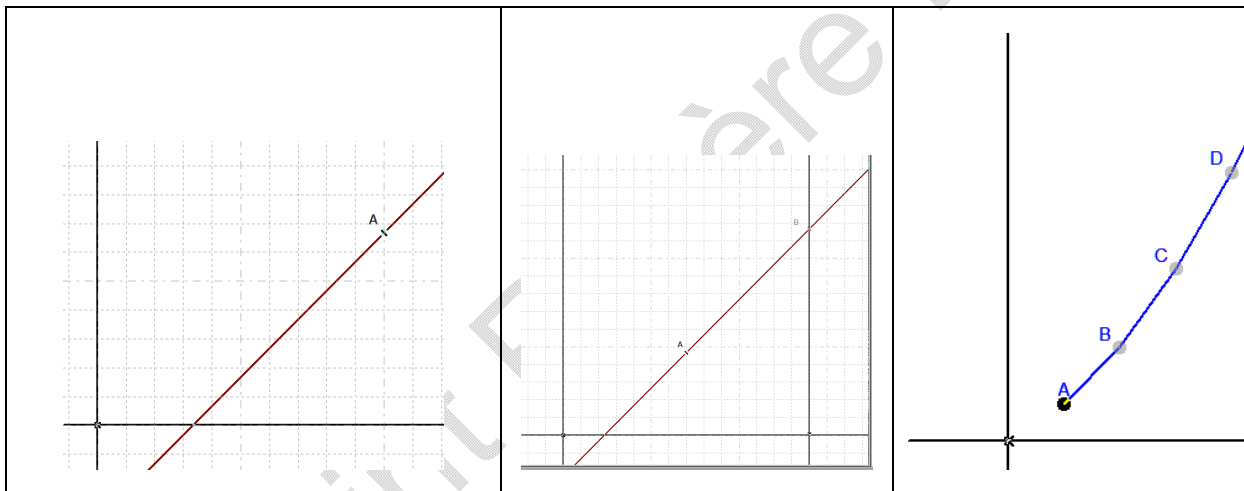


Figure 3

Travail algorithmique

Il s'agit ici de systématiser la construction précédente à l'aide d'un algorithme fonctionnant pour un nombre de pas arbitraire,⁶ et de faire apparaître la ligne brisée comme la représentation graphique d'une fonction g approchant la fonction inconnue. Nous allons montrer comment ce travail peut être effectué avec le module de programmation de Casyopée.

⁵ La situation est inspirée de http://lycee.lagrave.free.fr/IMG/pdf/TS_derivation_methode_Euler.pdf.

Néanmoins, les objectifs sont différents. Il ne s'agit pas d'introduire la méthode d'Euler, mais de renforcer la compréhension du lien entre fonction, nombre dérivé et tangente.

⁶ En tenant compte de la remarque de la note 4, il faut distinguer la définition mathématique où le nombre de pas est arbitraire, de l'implémentation dans un environnement informatique donné où les valeurs de ce nombre sont limitées. Dans ce dernier cas, on dira plutôt que le nombre de pas est « variable ».

Ce choix permet, comme nous l'avons annoncé plus haut, de définir une fonction approchant la fonction inconnue, plutôt qu'une suite de valeurs ou de points, et de faire dépendre cette fonction d'un paramètre.

Nous supposons définie dans le volet d'algèbre une fonction f' telle que $\forall x \in [1;5], f'(x) = \sqrt{x}$. Nous créons aussi un paramètre n positif (curseur) représentant le nombre de pas. L'algorithme (figure 3) permet de définir une fonction affine par morceaux g dépendante d'un paramètre n telle que $g(x)=y$ pour chaque couple de valeurs de (x, y) obtenus dans l'itération.

<pre> 1: $x \leftarrow 1$ 2: $y \leftarrow \frac{2}{3}$ 3: $h \leftarrow \frac{4}{n}$ 4: $I \leftarrow 1$ 5: Pour I allant de 1 à n 6: $y \leftarrow y + f'(x) \cdot h$ 7: $x \leftarrow x + h$ 8: Fin Pour </pre>	<p>Les lignes 1 et 2 initialisent les coordonnées du point courant pour l'extrémité gauche de la courbe représentant la fonction inconnue f (point A ci-dessus). La ligne 3 définit la longueur du pas (n pas dans un intervalle d'amplitude 4). La ligne 4 initialise une variable compteur.</p> <p>Les lignes 6 et 7 constituent le corps d'une boucle POUR, qui sera répété n fois.</p> <p>La valeur de l'ordonnée du point suivant est calculée en ligne 6. Elle se déduit de l'équation de la droite passant par le point courant et de pente $f'(x)$: $y=y_i + f'(x_i) (x-x_i)$, x_i et y_i étant les coordonnées du point courant. On obtient $y_{i+1} = y_i + f'(x_i) (x_{i+1} - x_i) = y_i + f'(x_i) h$.</p> <p>La ligne 7 calcule la valeur suivante de l'abscisse du point courant.</p>
--	---

Figure 3

Travail symbolique

Il s'agit d'obtenir une expression de la fonction inconnue à partir de connaissances sur les dérivées des fonctions usuelles. Cela utilise la propriété de la dérivée des fonctions $x \rightarrow k x^a$, a étant un paramètre non nul. Cette dérivée étant la fonction $x \rightarrow ka x^{a-1}$, il s'agit de l'identifier à la fonction $x \rightarrow \sqrt{x}$. Exprimant \sqrt{x} comme $x^{1/2}$, il vient $a-1=1/2$ et $ka=1$. Après calcul de a et k , et interprétation de l'expression $x^{3/2}$, on obtient la fonction définie par $f(x) = \frac{2}{3} x\sqrt{x}$. On vérifie qu'elle prend la valeur $2/3$ en 1. La figure 4 montre la représentation graphique de cette fonction, avec celle de la fonction g pour une valeur donnée de n , et avec la construction géométrique.

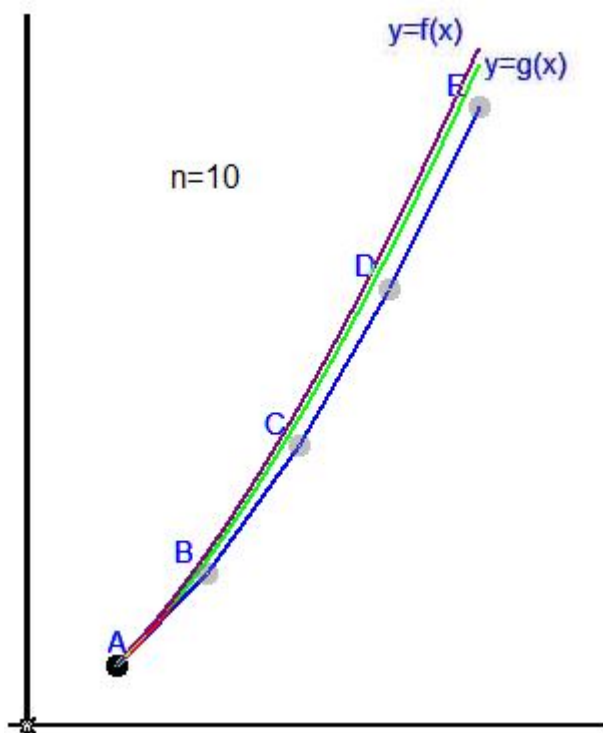


Figure 4

Mise en œuvre en classe

Organisation

La séance s'adresse à une classe de Première S de 33 élèves. Elle a lieu en salle ordinaire. Les élèves ont à leur disposition des ordinateurs portables avec Casyopée et un fichier où la fonction f et sa fonction dérivée f' ont été créées, mais où seule la fonction dérivée est visible. La fonction f est donc *cachée*. La séance se déroule en deux phases de travail de groupe d'une demi-heure chacune, suivies d'une synthèse collective.

Phase 1

Dans la première phase, les élèves sont répartis en six groupes. Deux groupes font une tâche étiquetée A. Deux groupes font une tâche étiquetée B. Deux groupes font une tâche étiquetée C. Dans les tâches A et B, il s'agit de construire la tangente au point d'abscisse 1, puis des tangentes "approchées" à la courbe représentative de f comme indiqué ci-dessus. La tâche A se fait dans le volet de géométrie dynamique de Casyopée et la tâche B se fait en papier crayon. Les propriétés à utiliser par les élèves sont :

- un point sur la tangente est "proche" de la courbe,
- la pente de la tangente en un point de la courbe est le nombre dérivé en ce point, c'est à dire la valeur de la fonction dérivée pour l'abscisse de ce point.

Pour la tâche A, les élèves peuvent créer le point d'abscisse 2 (resp. 3, 4 et 5) par intersection de la droite déjà créée avec une parallèle à l'axe des ordonnées. On peut s'attendre cependant à ce que des élèves placent ce point au jugé sur cette droite, de façon à ce que son abscisse soit 2 (resp. 3, 4 et 5), ce qui n'est pas gênant ici. Ensuite ils sont guidés pour la construction de la tangente "approchée" en ce point.

Pour la tâche B, en papier crayon, les points sont portés en s'aidant du repère et la construction d'une droite de pente donnée passant par un point donné est classique : "je vais à droite de 1, je monte de la valeur de la pente". Il est possible que certains élèves calculent l'équation des droites avant de les tracer, ce qui nécessite un calcul de l'ordonnée du point porté précédemment. Dans la tâche C les élèves doivent proposer une valeur approchée de $f(2)$, $f(1,5)$, $f(1,1)$ et $f(1,01)$. Il s'agit de réactiver la définition du nombre dérivé. On peut penser que les élèves vont directement assimiler le taux d'accroissement à ce nombre et conclure par exemple $f(1,01) = f(1) + 0,01 f'(1)$.

Phase 2

Dans cette phase, les élèves travaillent par groupes de trois. Il leur est demandé de concevoir un algorithme permettant d'obtenir une fonction approchant la fonction cachée f . Chaque groupe compte un élève ayant fait la tâche A, un élève ayant fait la tâche B et un élève ayant fait la tâche C⁷. Les tâches A et B ayant permis de voir le tracé comme un processus à plusieurs étapes: une initialisation, puis la répétition de quatre constructions, les élèves peuvent donc reconnaître un processus itératif. Le passage à un algorithme implique de penser le processus de façon fonctionnelle : à une valeur x de l'abscisse correspond une valeur y de l'ordonnée. Ensuite, il faut inclure dans le corps de boucle le calcul de l'ordonnée du point courant. Celui-ci est préparé par la tâche C. Le pas 1 est cohérent avec le calcul de $f(2)$ dans la tâche C et les abscisses des points dans les tâches A et B. Une itération pour ce pas sera sans doute la plus facile à mettre en place. La tâche C suggère que des pas plus petits peuvent être utilisés, ce qui nécessite une adaptation de l'algorithme. Lors de la synthèse le professeur montrera l'utilisation d'un pas arbitraire comme ci-dessus.

Avec Casyopée, les algorithmes tels que ceux qui viennent d'être présentés définissent tels quels une fonction affine par morceaux dont les élèves pourront facilement faire tracer la courbe par le logiciel et comparer à leur construction de la tâche A, ou à la courbe de la fonction f cachée (annexe ci-dessous).

Analyse du travail d'un groupe d'élèves en phase 2

Nous analysons le travail d'un groupe de trois élèves au cours de la deuxième phase à partir d'un enregistrement vidéo et du document « papier/crayon » réalisé par les élèves. Comme les élèves utilisent pour la première fois le module de programmation de Casyopée, l'observateur les guide pour l'utilisation de ce module et d'autres fonctionnalités. Il s'agit d'un groupe d'élèves relativement à l'aise avec les tâches demandées et ils réussissent ces tâches dans le temps imparti. Nous récapitulons les points qui ont pu constituer des difficultés. Ces difficultés ont été rencontrées et parfois non surmontées par les autres groupes.

Dans le volet de géométrie dynamique, le tracé des droites et la fonctionnalité correspondante du logiciel, ainsi que la technique pour porter un point sur la tangente sont bien compris. En revanche, les élèves restent indécis quant au statut d'un point ainsi porté. Comment un point d'une courbe peut-il être sur la tangente en un autre point ? Les élèves font difficilement le lien avec la tâche C de la première phase et plus généralement avec la problématique d'approximation.

⁷ Pour une présentation de ce dispositif, couramment pratiqué dans nos classes, voir Halbert & Manens (2017).

Pour l'élaboration de l'algorithme, les élèves sont influencés par un travail sur la calculatrice : l'affichage de points de la courbe d'une fonction avec un pas donné. C'est pourquoi ils proposent une boucle POUR sur la variable x avec un pas de 0,1. Mais l'adaptation n'est pas aisée. D'une part, seul un pas de 1 est prévu dans l'environnement de programmation de Casyopée avec la version utilisée, et donc l'observateur propose d'adopter un pas de 1. D'autre part, dans l'affichage de points d'une courbe, y est calculé dans la boucle sans référence à sa valeur antérieure, alors dans l'algorithme en jeu, le calcul prend en compte la valeur antérieure. L'algorithme tel que proposé par les élèves à partir de la construction géométrique comporte un corps de boucle correct $y \leftarrow y + \sqrt{x}$, mais calcule l'ordonnée pour une abscisse $x+1$ plutôt que x .

L'introduction d'une variable compteur permet de faire évoluer explicitement x dans le corps de boucle. Le point délicat est celui de l'ordre des instructions dans le corps de boucle. Le choix des élèves de faire évoluer x avant y paraît naturel, mais ici ce n'est pas adapté car, comme dit plus haut $y \leftarrow y + \sqrt{x}$ calcule l'ordonnée pour une abscisse $x+1$. La visualisation invalide ce choix et les élèves intervertissent les instructions du corps de boucle. Ensuite, assez facilement, ils adaptent l'algorithme pour un pas de 1/10. La figure 5 donne l'algorithme tel qu'écrit par les élèves : il s'agit de l'algorithme pour le pas 1 « raturé » pour un pas 1/10.

Algorithme sur l'ordi

1: $x \leftarrow 1$
 2: $y \leftarrow 2/3$
 3: $I \leftarrow 1$
 4: Pour I allant de 1 à 50
 5: $y \leftarrow y + \sqrt{x}/10$
 6: $x \leftarrow x + 0,1$
 7: Fin pour.

Figure 5

Les élèves s'étonnent de l'absence d'une instruction de sortie « *il faut faire afficher les points ?* ». L'observateur précise que l'algorithme définit une fonction dont la représentation graphique pourra être affichée par le logiciel.

Les élèves sont ensuite incités à trouver une expression pour la fonction inconnue f , de façon à comparer avec la fonction définie par l'algorithme. Ils ont bien intégré les règles de dérivation et l'interprétation de la racine carrée comme la puissance d'exposant $1/2$ et trouvent relativement facilement une expression. Affichant sur le même graphique la courbe de cette

fonction et celle de la fonction définie par l'algorithme les élèves constatent avec satisfaction leur quasi-coïncidence.

Conclusion

A partir de l'analyse du travail des élèves dans l'exemple de situation qui vient d'être donnée, nous relevons que les points portés dans la construction géométrique n'ont pas eu d'emblée une signification claire, mais que, à partir du travail sur l'algorithme, les élèves ont bien compris que celui-ci ne définit pas des points isolés, mais une fonction d'un nouveau type approchant la fonction inconnue. Nous relevons aussi que, pour ces élèves qui n'ont pas fait d'algorithmique au collège, une structure itérative, même simple, ne va pas de soi.

Nous avons fait les mêmes observations dans d'autres situations. Par exemple, en Terminale pour la modélisation d'un câble principal d'un pont suspendu (Le Feuvre à paraître), les élèves ont eu à coordonner une approche géométrique, en calculant les coordonnées des points de suspension à partir d'une étude des tensions, et des approches algorithmique et symbolique de fonctions modélisant le câble. Ici aussi nous avons observé un travail en algorithmique non trivial pour les élèves et la prise de conscience, à partir de ce travail, de la relation entre une fonction affine par morceaux définie par l'algorithme et une fonction définie par une formule qui modélise le câble en supposant un nombre infini de points de suspension.

En Seconde, les élèves ont beaucoup de difficultés à considérer la trajectoire d'un point mobile dans un repère du plan comme la représentation graphique d'une fonction $x \rightarrow y$ et à faire intervenir le temps, même dans le cas d'un mouvement uniforme (Vandebrouk et Cazes, 2014). Nous avons expérimenté une situation autour d'algorithmes de parcours « pas à pas » de trajectoires (figure 6). Avec des fonctions par algorithme ainsi définies, les élèves ont pu résoudre de façon fonctionnelle un problème de rencontre de mobiles (Coyote va-t-il rattraper BipBip ?) qu'ils n'auraient pas pu résoudre directement par l'algèbre.

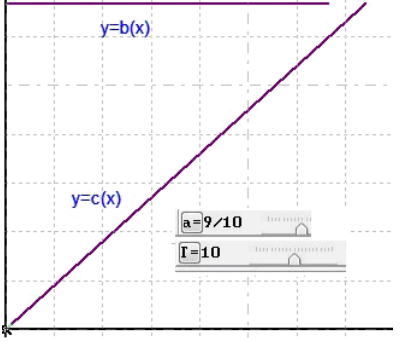
<pre> 1: x ← 0 2: y ← 20 3: t ← 0 4: répéter 5: t ← t+1 6: x ← x+2 7: jusqu'à t > T </pre>	<pre> 0: x ← 0 1: y ← 0 2: t ← 0 3: répéter 4: t ← t+1 5: x ← x+3/√(1+a^2) 6: y ← y+3*a/√(1+a^2) 7: jusqu'à t > T </pre>	
<p>Algorithme de BipBip</p>	<p>Algorithme de Coyotte</p>	<p>Les représentations graphiques : Coyote passe devant BipBip sans l'atteindre</p>

Figure 6

Ces observations vont dans le sens de notre hypothèse : il est possible de construire des situations permettant aux élèves à la fois de développer des compétences en algorithmique et d'élargir la compréhension des fonctions. L'idée de « fonction définie par un algorithme » nous semble un point-clé pour cela. Dans les situations proposées, elle rompt avec une

approche purement graphique où la fonction n'apparaît pas. Elle introduit un nouveau type de fonction pour lequel la représentation « formule » n'est pas pertinente, mais dont les autres représentations peuvent être traitées comme celles des fonctions « habituelles ». Les élèves prennent conscience de ce que la représentation « formule » n'est pas indispensable pour définir une fonction et travailler avec, alors que, habituellement, elle occupe une place dominante. Notre démarche conduit aussi, certes modestement, à faire travailler les élèves sur des objets mathématiques ayant un mode de génération particulier à l'informatique, ce que certains auteurs comme Knuth (1985) rattachent à une « pensée algorithmique » spécifique.

Un autre point important est le style de programmation : les instructions d'entrée/sortie sont absentes, ce qui amène une compréhension de l'algorithme comme définissant un objet, plutôt que réalisant une action ; l'écriture demandée reste proche des pratiques au lycée et tient compte de ce que les élèves de lycée ont encore dans leur majorité une expérience limitée de la programmation.

Nous avons exposé les choix méthodologiques qui nous ont permis de tester notre hypothèse, notamment notre choix d'utiliser le logiciel Casyopée après l'avoir fait évoluer. Nous avons expliqué ce choix par les limitations ressenties avec d'autres logiciels et l'histoire de notre groupe. Ce choix est aussi cohérent avec les capacités encore limitées en algorithmique et programmation de beaucoup d'élèves de lycée. Nous avons vérifié dans l'exemple de situation présenté plus haut, que la programmation d'un traitement itératif sur deux variables ne va pas de soi pour les élèves observés.

Il est possible qu'avec l'introduction de l'algorithmique et de la programmation au collège, les élèves de lycée soient dans l'avenir mieux préparés à l'utilisation d'un style de programmation fonctionnel tel que proposé dans le document ressource (MEN 2017b) plutôt que d'un environnement TICE comme Casyopée. D'autres choix seraient alors possibles pour les situations que nous avons évoquées dans l'article⁸. Nous espérons que l'analyse que nous proposons sera une source d'inspiration.

Bibliographie

Cazes, C. et Vandebrouck, F. (2014) Vil Coyote rattrapera-t-il Bip-Bip ? *Repères-IREM*, N° 95, 5-22.

Halbert R. et Manens M-C. (2017) Un "Jigsaw-teaching" en classe de TS pour l'étude de la fonction ln. *Bulletin de l'APMEP*. Num. 522. 10-20.

Halbert R., Lagrange J-B., Le Bihan C., Le Feuvre B., Manens M-C., Meyrier X. (2013). *Les fonctions : comprendre la notion et résoudre des problèmes, de la 3ème à la terminale*. Edition IREM de Rennes, Université Rennes 1.

Halbert R., Lagrange J-B., Le Bihan C., Le Feuvre B., Manens M-C., Meyrier X. (2016). Définir des fonctions par un algorithme : motivation, réalisation dans Casyopée et exemples. *Mathematice* N°53 <http://revue.sesamath.net/spip.php?article914>

⁸ Par exemple, pour la fonction de décroissance h du baccalauréat 2018 pris plus haut comme exemple, il faudrait d'abord définir récursivement une fonction donnant la température pour les valeurs entières ou $\frac{1}{2}$, puis la fonction définie pour les réels positifs par interpolation entre les valeurs entières ou $\frac{1}{2}$, les plus proches.

- Knuth, D.E. (1985). Algorithmic thinking and mathematical thinking. *The American Mathematical Monthly*, 92(1), 170-181.
- Lagrange, J.B., Rogalski, J. (2017) Savoirs, concepts et situations dans les premiers apprentissages en programmation et en algorithmique. *Annales de didactique et de sciences cognitives*. Vol 22, 119-158.
- Le Feuvre, B. (à paraître) Espaces de travail mathématiques connectés au lycée : un pont entre les mathématiques et les sciences physiques. *Actes de EMF 2018*. Genevilliers, octobre 2018.
- MEN (2017a) Seconde générale et technologique, Aménagements des programmes d'enseignement de mathématiques et de physique-chimie. http://www.education.gouv.fr/pid285/bulletin_officiel.html?cid_bo=115984
- MEN (2017b) Ressources pour le lycée, Mathématiques, Algorithmique et programmation. http://cache.media.eduscol.education.fr/file/Mathematiques/73/3/Algorithmique_et_programmation_787733.pdf
- MEN (2009) Ressources pour la classe de seconde - Fonctions - http://cache.media.eduscol.education.fr/file/Programmes/18/1/Doc_ressource_fonctions_109181.pdf

Annexe : l'environnement Casyopée

L'environnement Casyopée et le projet qui le sous-tend sont présentés sur le site <http://casyopee.eu>. Halbert, Lagrange, Le Bihan et al. (2016) présentent en détail l'implémentation des fonctions définies par un algorithme qui font l'objet de l'article.

Voici quelques données visant à faciliter la lecture de l'article.

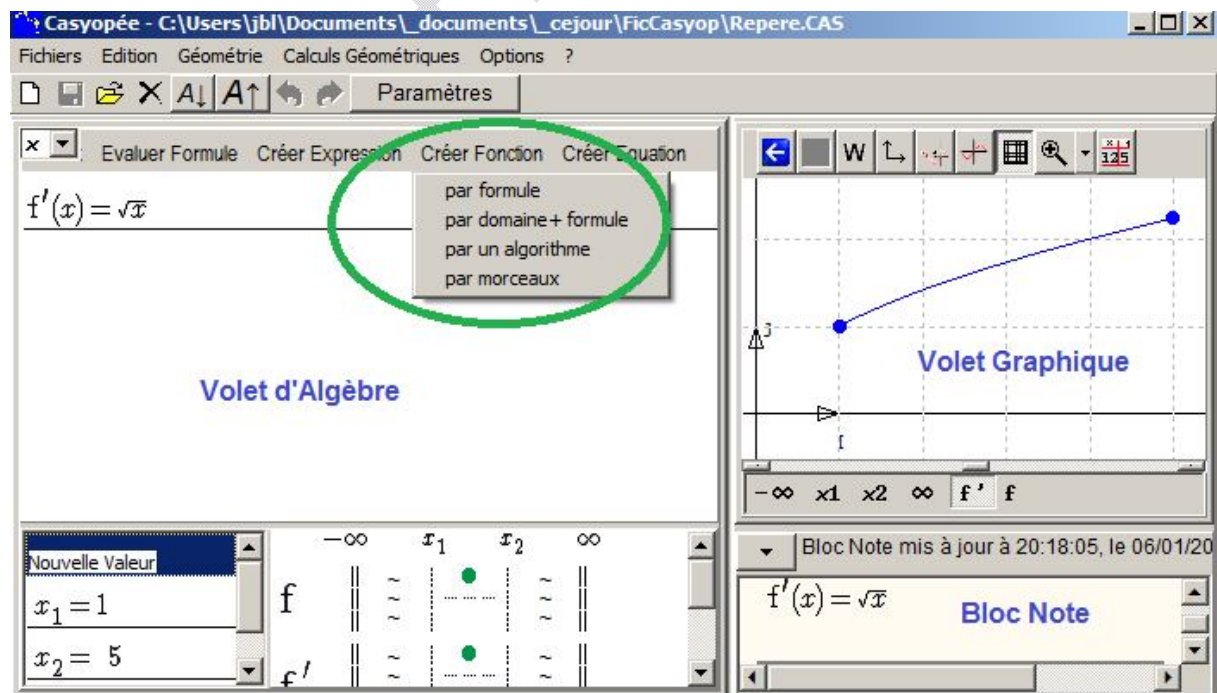


Figure 7

L'interface de Casyopée est constituée de 4 volets qui peuvent être affichés deux par deux. Les volets de Géométrie dynamique et de Calculs géométriques ont été construits pour des situations de modélisation géométrique, telles que l'exemple de la distance d'un point fixe à une parabole ci-dessus. Les fonctions définies par un algorithme font intervenir principalement les deux autres volets : volet d'algèbre et volet graphique (figure 7). L'entrée « créer fonction » du menu du volet d'algèbre propose quatre sous-entrées. Trois entrées proposent différents mode de définition faisant intervenir une formule, chacune activant une boîte d'entrée spécifique. Dans l'exemple de situation mise en œuvre, une fonction f dérivable sur $[1; 5]$ telle que $\forall x \in [1;5], f'(x) = \sqrt{x}$ a été ainsi été définie (figure 7). Sa dérivée apparaît dans la liste des expressions du volet d'algèbre et peut être graphée dans le volet graphique. Les fonctions apparaissent aussi dans un mini tableau, en bas du volet d'algèbre.

Nous nous intéressons ici à l'entrée « par un algorithme ». Elle active une boîte d'entrée qui propose un embryon de programme, séparant déclaration-initialisation et itération (figure 8). Les variables x et y sont prédéclarées.

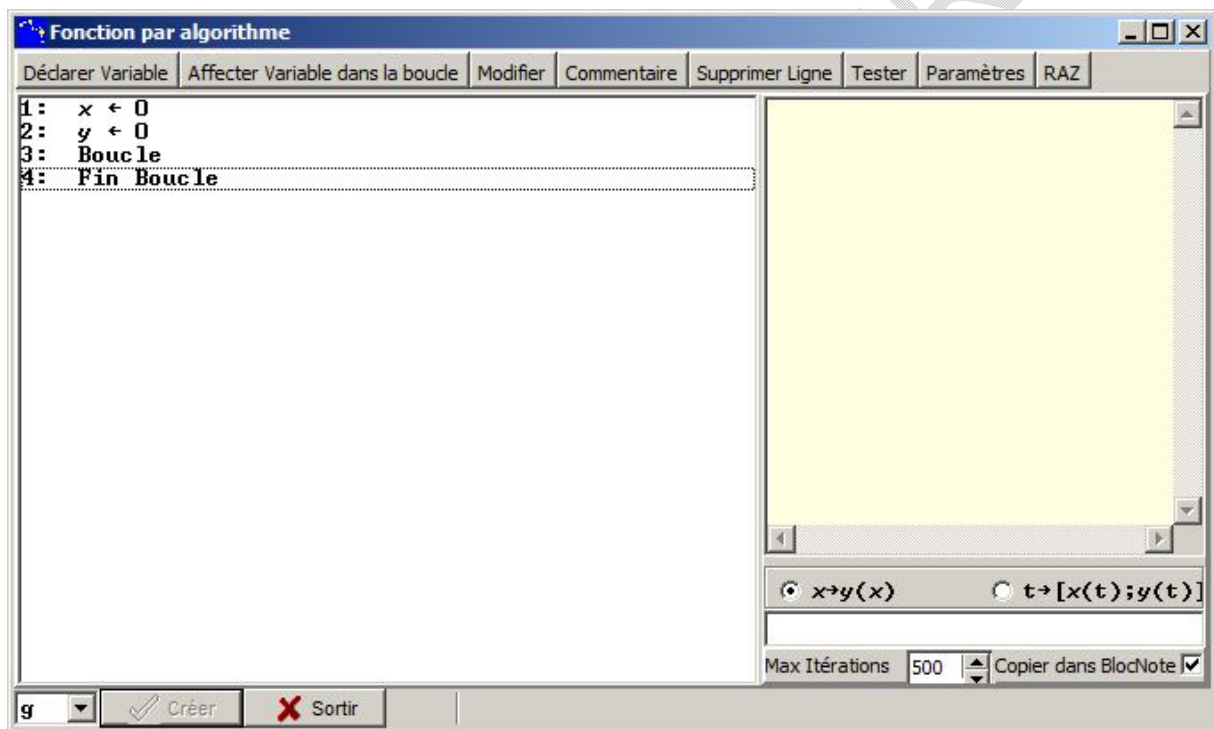


Figure 8

Chacune des parties peut être modifiée à l'aide des entrées de menu dans la partie Editeur (figure 9). Des aides à l'écriture sont fournies, notamment en proposant des identificateurs de variable adéquats lors de la composition de chaque ligne.

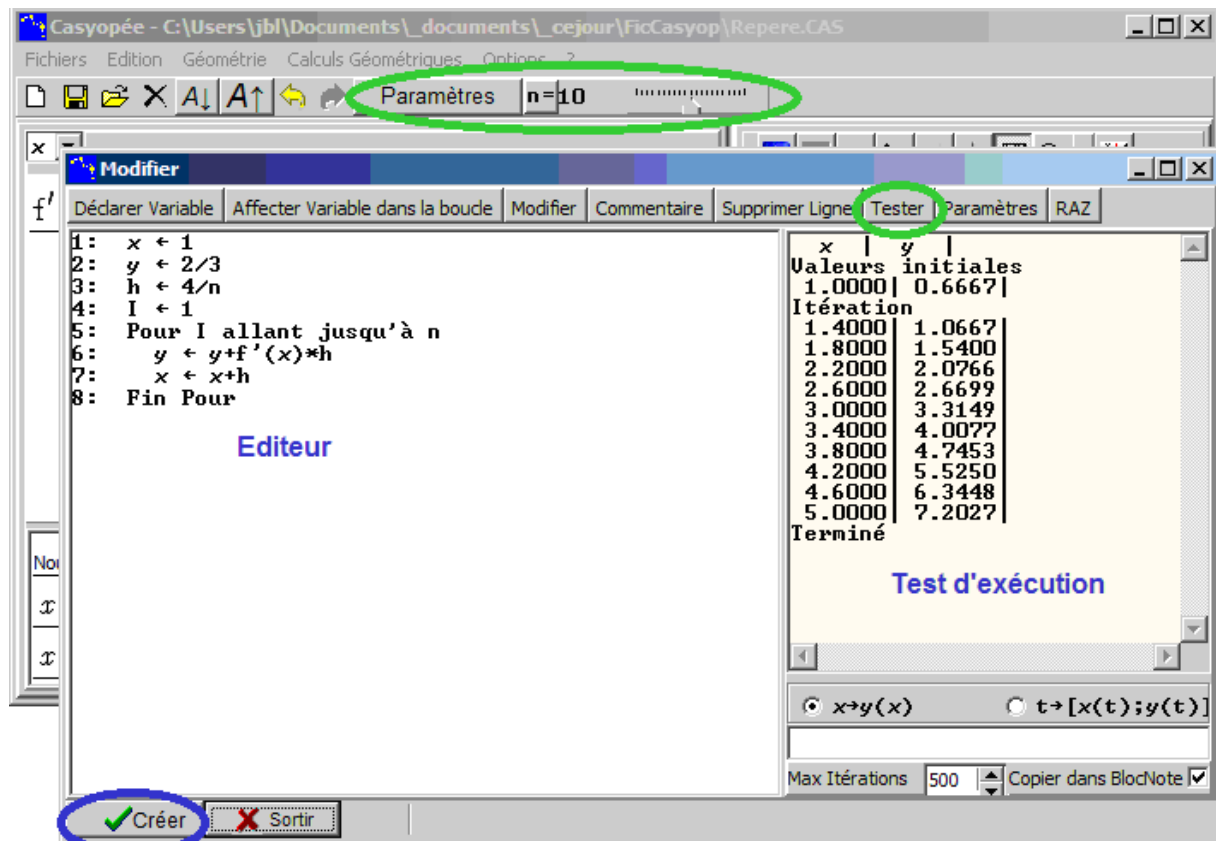


Figure 9

L'entrée de menu Tester permet d'évaluer la syntaxe du programme et d'étudier les valeurs prises par les variables dans la partie Test d'exécution (figure 9). Le paramètre n créé précédemment ou au moment du test a le statut de variable globale, sa valeur ne dépendant pas de l'exécution du programme et étant fixée par le biais d'un curseur.

Si l'exécution réussit, un bouton Créer permet la création d'une fonction affine par morceaux qui prend la valeur de la variable y pour chaque valeur de la variable x et est affine sur chacun des intervalles délimités par deux valeurs successives de la variable x. La fonction apparaît dans la table en bas du volet d'Algèbre et peut être graphée dans le volet graphique. Différents calculs peuvent être opérés sur cette fonction dont le résultat apparaît dans le Bloc Note (figure 10). Il est possible aussi d'afficher la courbe de la fonction dans le volet de Géométrie dynamique, de façon à comparer avec une construction segment par segment (figure 4 ci-dessus). Les affichages sont mis à jour dynamiquement au changement de valeur du paramètre n.

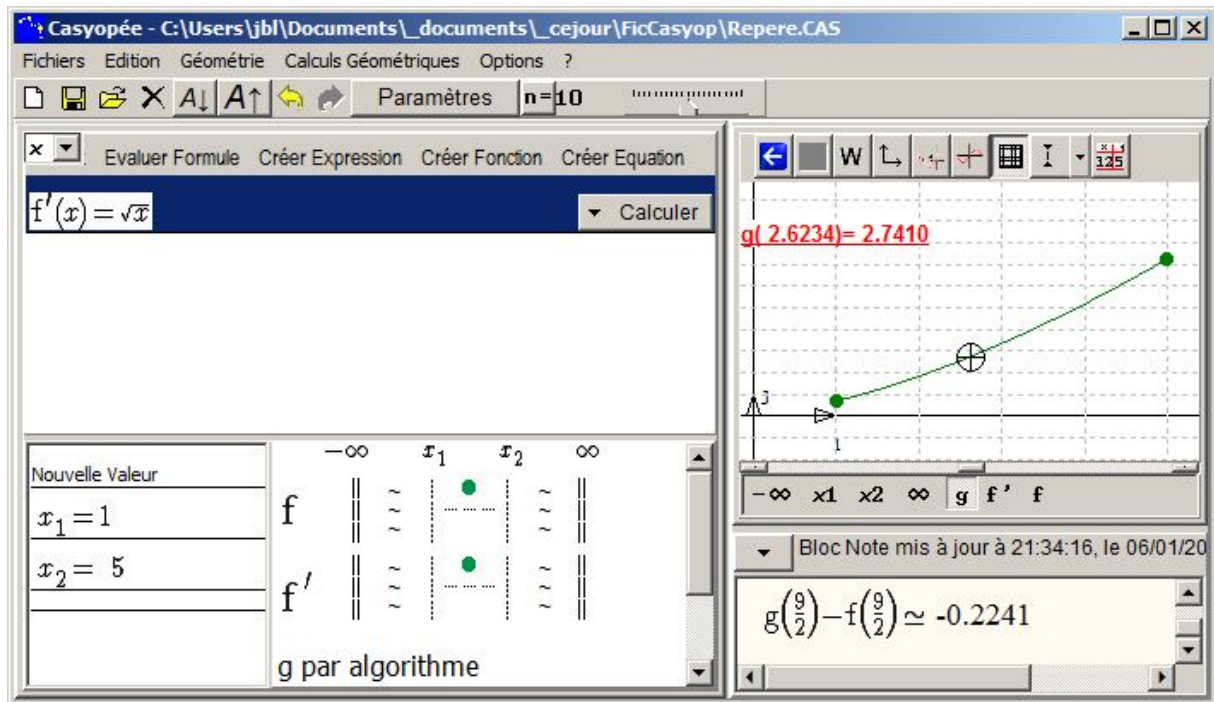


Figure 10