Algorithmique au Lycée

Commission de Réflexion sur l'Enseignement des Mathématiques

(suite du numéro 445)

4. Algèbre et géométrie

4.1. Constructions

Dans les problèmes de construction on demande de déterminer un algorithme qui, à partir des données, fournit le résultat escompté en utilisant seulement des fonctions précisées.

Les fonctions utilisées sont généralement décrites par l'expression « à la règle et au compas ». On peut imaginer d'autres fonctions permises, par exemple à travers l'usage d'autres instruments. L'usage du « double-décimètre » est généralement proscrit.

Les entrées et les sorties ont nécessairement une forme particulière qui exclut souvent les nombres.

Donnons quelques exemples.

Procédure *Conjugué harmonique*
Entrées : A, B, M points du plan distincts deux à deux et alignés sur une droite D.
Sorties : N point de la droite D tel que $\frac{NA}{NB} = \frac{MA}{MB}$.
Choisir un point w en dehors de D.
Tracer la droite Δ passant par M et w.
Mener la parallèle δ_1 à Δ par A.
Mener la parallèle δ_2 à Δ par B.
Tracer la droite D_1 par A et w.
Tracer la droite D_2 par B et w.
Marquer le point d'intersection AA des droites δ_1 et D_2.
Marquer le point d'intersection BB des droites δ_2 et D_1.
Tracer la droite DD par AA et BB
Marquer le point d'intersection N des droites D et DD.

On a utilisé ici les procédures

Choisir un point dans le plan,
Tracer la droite par deux points,
Mener par un point la parallèle à une droite,
Marquer le point d'intersection de deux droites

que l'on suppose déjà écrites ou données. On remarque que la première de ces procédures n'est pas déterministe.

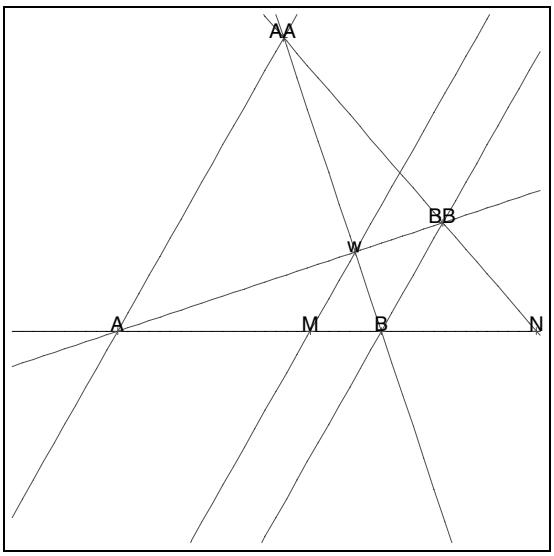

FIG. 6 – Conjugué harmonique: construction

On démontre ensuite (Thalès) que le résultat convient et que l'algorithme proposé s'arrête au bout d'un nombre fini de pas pour toute donnée convenable. Reste à vérifier que le point N trouvé est le seul point de la droite D, différent de M, qui convient.

Si on permet l'utilisation d'un instrument de mesure des longueurs et les multiplications et divisions de réels, il existe un autre procédé : mesurer MA et MB, calculer $\frac{MA}{MB}$, et en déduire, par exemple, une mesure de NA. Cette démarche est très proche de la preuve de l'unicité de la solution.

1. Les procédures élémentaires évoquées au cours de l'exemple précédent sont toutes intéressantes à programmer, de même que les constructions de la médiatrice d'un segment, de la projection orthogonale d'un point sur une droite, du cercle circonscrit à un triangle, …

   Les fonctions primitives sont celles de la règle et du compas c'est-à-dire :

   Choisir un point dans le plan,
   Tracer la droite par deux points,
   Tracer le cercle de centre donné passant par un point donné,
   Marquer le point d'intersection de deux droites, d'une droite et d'un cercle.

2. On note que les objets manipulés sont des points, des droites, des cercles du plan, qui peuvent être soit des primitives du langage, soit recouvrir une représentation adaptée : expression à l'aide de coordonnées, tracé à l'écran, etc. Les algorithmes en dépendent : penser par exemple à l'intersection de deux droites.

4.2. Réduction d'un système linéaire, pivot de Gauss

La résolution d'un système linéaire est abordée au Lycée de manière expérimentale, sur des exemples. Cette méthode de calcul conduit en fait à un algorithme, qui a de nombreuses variantes. L'étude de la complexité de ces algorithmes est intéressante, étant donnée l'importance des calculs linéaires dans les calculs effectifs.

On part d'un système de n équations linéaires homogènes E_1, …, E_n à p inconnues x_1, …, x_p. Pour $1 \leq i \leq n$, l'équation d'indice i a pour coefficients $a_{i,1}$, …, $a_{i,p}$.Ce sont des éléments du corps **k** (qui peut être celui des réels, des complexes, mais aussi celui des rationnels, celui des entiers modulo un nombre premier p ou pire encore : on veut juste pointer ici que les calculs effectués ne sortiront jamais du corps de départ). On agence ces coefficients en un tableau $n \times p$, *i.e.* une matrice, dont la ligne d'indice i a pour éléments les coefficients de la i-ème équation. Dans la suite on parlera indifféremment de la ligne du tableau ou de l'équation correspondante.

On dispose des deux procédures :

P1 : **Multiplier (tous les éléments d')une ligne d'un tableau par un élément inversible du corps.**

P2 : **Soustraire d'une ligne du tableau un multiple d'une autre ligne.**

Ces deux procédures ont une propriété fondamentale : leur résultat (*i.e.* le système obtenu) a les mêmes solutions que le système de départ.

En utilisant ces deux procédures (appelées parfois élémentaires), on écrit la procédure suivante :

Procédure *Pivot de Gauss (essai)*
Entrées : T, **Tableau** $n \times p$ d'éléments du corps **k**.
Sorties : S, **Tableau** $n \times p$ d'éléments du corps **k**.
Pour j **de** 1 **à** p **faire**
Pour i **de** 1 **à** n **faire**
  si $a_{i,j} \neq 0$ **alors faire** *Pivot* := $[i,j]$

$$E_i \leftarrow \frac{1}{a_{i,j}} E_i$$

  pour ℓ **de** $i+1$ **à** n **faire**
    $E_\ell \leftarrow E_\ell - a_{\ell,j} E_i$

On remarque que cette procédure, composée des procédures élémentaires P1 et P2, ne change pas non plus l'ensemble des solutions. Le pivot existe si l'un au moins des coefficients du tableau est non nul. Sauf dans le cas où les tableaux d'entrée et de sortie sont nuls, le résultat de la procédure *Pivot de Gauss (essai)* est un système S *résolu en* x_j, ce qui signifie :

– il ne dépend pas des inconnues x_1, …, x_{j-1},
– la seule équation qui contient x_j est la i-ème,
– le tableau obtenu en conservant les colonnes d'indice $k > j$ et les lignes d'indice $\ell \neq i$ est associé à un système linéaire S′ où ne figurent plus les inconnues x_1, …, x_j. Toute solution du système S est obtenue à partir d'une solution de S′ et

de la valeur de x_j donnée par la i-ème équation.

Pour résoudre le système, il suffit de répéter la procédure *Pivot de Gauss (essai)* avec le système S′. Pour le faire convenablement, on modifie légèrement cette procédure pour pouvoir la composer avec elle-même :

Procédure *Pivot de Gauss*
Entrées : T, **Tableau** $n \times p$ d'éléments du corps **k**,
L, **liste** de couples d'entiers.
Sorties : S, **Tableau** $n \times p$ d'éléments du corps **k**,
Liste des pivots, **liste** de couples d'entiers.
Initialisation : E := T,
s := numéro de la ligne du dernier élément de L,
t := numéro de la colonne du dernier élément de L.
Pour j **de** $t+1$ **à** p **faire**
Pour i **de** $s+1$ **à** n **faire**
si $a_{i,j} \neq 0$ **alors faire** *Pivot* := $[i,j]$ **et** K := L, $[i,j]$.

$$\mathrm{E}_i \leftarrow \frac{1}{a_{i,j}}\mathrm{E}_i.$$

pour $\ell \leq n$ **et** $\ell \neq i$ **faire**
$\mathrm{E}_\ell \leftarrow \mathrm{E}_\ell - a_{\ell,j}\mathrm{E}_i$
retourner T, K **et sortir**
retourner *Pivot* := *fin*.

On obtient donc :

Procédure *Résolution*
Entrées : T, **Tableau** $n \times p$ d'éléments du corps **k**,
Sorties : R, **Tableau** $n \times p$ d'éléments du corps **k**,
Liste des pivots, **liste** de couples d'entiers.
Initialisation : U := (T, *listevide*), P := [0,0].
Tant que *Pivot* ≠ *fin* **faire** U := *Pivot de Gauss*(U)

Variantes. Il existe de nombreuses variantes de cet algorithme. Celle que nous venons d'expliciter tient pour connues les procédures de calcul dans le corps **k** des coefficients du système de départ. Cependant, on sait que ces procédures de calcul ne sont pas aussi simples qu'il y paraît. Par exemple : pour des calculs exacts sur **Q**, les divisions peuvent faire croître les dénominateurs successifs ; par ailleurs, pour le cas de calculs approchés, la différence entre deux éléments très proches peut avoir comme résultat un nombre dont l'ordre de grandeur est très éloigné de celui des opérandes et rendre très sensible le recours aux arrondis. Les façons de remédier à ces difficultés sont diverses.

On peut effectuer une meilleure recherche du pivot, pour laquelle le coefficient pivot sera par exemple le plus grand possible (et pas le premier venu non nul). Les questions effleurées ici conduisent à la considération du *conditionnement* du système, qui dépasse nos objectifs.

On peut également ne jamais diviser (donc rester dans l'anneau engendré par les

coefficients du système de départ) et considérer la variante suivante de la procédure *Pivot de Gauss (essai)*, écrite pour des coefficients entiers :

Procédure *Pivot de Gauss (essai), coefficients entiers*
Entrées : T, **Tableau** $n \times p$ d'éléments d'éléments de **Z**.
Sorties : S, **Tableau** $n \times p$ d'éléments d'éléments de **Z**.
Pour j **de** 1 **à** p **faire**
Pour i **de** 1 **à** n **faire**
  si $a_{i,j} \neq 0$ **alors faire** $Pivot := [i,j]$
  pour ℓ **de** $i+1$ **à** n **faire**
    $\delta := \text{pgcd}(a_{i,j}, a_{\ell,m})$, $b := a_{i,j}/\delta$, $c := a_{\ell,m}/\delta$
    $E_\ell \leftarrow bE_\ell - cE_i$
  sortir

Rang. Si *Liste des pivots* $= [i_1, j_1], \ldots, [i_r, j_r]$, *fin* alors $j_1, \ldots, j_r$ est la liste des indices des variables dites principales dans le processus de résolution choisi. Pour toute valeur donnée aux variables x_j, $j \notin \{j_1, \ldots, j_r\}$, l'équation E_{i_k} $(1 \leq k \leq r)$ donne une unique valeur de x_{j_k}. C'est en cela que le système obtenu est dit *résolu*.

L'entier r est le rang du système. Jusque là il n'est pas clair que c'est un invariant du système étudié. Cela ne découle pas simplement de l'algorithme de résolution. Le montrer revient à étudier la notion de dimension. En revanche on a construit explicitement une présentation de l'espace des solutions muni d'une projection sur $\mathbf{k}^{p-r}$ qui est aussi un isomorphisme linéaire.

Indépendance linéaire, relations. Il est facile, à partir de la procédure *Pivot de Gauss*, d'écrire un algorithme qui, partant d'un système de vecteurs $v_1, \ldots, v_p$ et d'un vecteur v, tous dans $\mathbf{k}^n$, décide si v est combinaison linéaire des $v_1, \ldots, v_p$ et, si oui, donne les coefficients d'une telle combinaison.

De même, il est facile d'écrire, toujours à partir de la procédure *Pivot de Gauss*, un algorithme d'inversion d'une matrice carrée $n \times n$ ou de calcul de son déterminant.

On montre que la complexité de ces algorithmes est en $O(n^3)$, la mesure de *coût* étant le nombre des additions et des multiplications effectuées dans le corps de base. Les considérations précédentes montrent qu'on doit parfois utiliser des mesures plus fines, notamment si l'on doit faire des calculs exacts, avec des entiers, des rationnels, ou des polynômes(10), par exemple.

Signalons enfin que Gauss, qui est à l'origine de nombreuses méthodes de calcul matriciel effectif, était motivé par des considérations très pratiques de mécanique céleste (l'orbite de Pallas) et de géodésie (la triangulation de l'état du Hanovre) : voir les notes historiques du livre *Histoire d'algorithmes*, déjà cité. De nos jours, ces méthodes interviennent, via la discrétisation et les méthodes d'éléments finis dans la plupart des résolutions numériques (sur ordinateur, bien sûr) d'équations différentielles : en météorologie, simulation d'écoulements fluides, analyse numérique des systèmes mécaniques, etc.

(10) Ces questions de calcul effectif sont excellemment traitées dans les ouvrages suivants :
C. Gomez, B. Salvy, P. Zimmermann, *Calcul formel : Mode d'emploi – Exemples en Maple*, Masson, 1995.
P. Dumas, X. Gourdon, *Maple : Son bon usage en mathématiques*, Springer, 1997.

5. Analyse : intégration numérique

La discussion de modèles dynamiques simples et motivés pose assez vite des questions mathématiques pour lesquelles l'expérimentation numérique peut être source de compréhension. L'implantation de ces méthodes permet d'aborder d'une manière simple et illustrée les concepts fondamentaux de la programmation : boucles, tests, gestion de fichiers, graphes. Mais elle permet aussi d'explorer des questions d'analyse mathématique sortant du programme, dans l'esprit de travaux d'approfondissement.

5.1. Équations différentielles : méthode d'Euler

Soit f une fonction de **R** vers **R**. On considère l'équation différentielle

$$y'(t) = f(y(t)),\ x \geq 0,\ y(0) = y_0. \qquad (1)$$

La méthode d'Euler consiste à approcher la solution de (1) par la récurrence

$$y_{k+1} = y_k + hf(y_k),\ k = 1, \ldots$$

Ici $h > 0$ est le pas de temps (destiné à tendre vers 0), et y_k représente une approximation de $y(kh)$. On a pris la même notation y pour la solution de l'équation différentielle et son approximation, sans risque de confusion.

On peut associer au pas h la fonction $y^h(t)$ définie en interpolant linéairement entre les points kh et $(k + 1)h$:

$$\begin{cases} y^h(kh) = y_k, \quad k = 1, \ldots, \\ y^h(t) \text{ affine sur } [kh, (k+1)h]. \end{cases}$$

On peut vérifier que si f est localement lipschitzienne, on a une estimation du type

$$|y^h(t) - y(t)| \leq C_T h,\ t \in [0,T]$$

au moins si T est assez petit. Nous admettons ce résultat.

Remarque. On peut tout de même analyser ce qui se passe pour $t \in [0, h]$. Puisque $y(t)$ est primitive de sa dérivée, on a :

$$y(h) = y(0) + \int_0^h f(y(t))\,\mathrm{d}t \qquad (2)$$

et aussi $y_1 = y(0) + \int_0^h f(y(0))\,\mathrm{d}t$, donc l'erreur $e_1 := y(h) - t_1$ vérifie

$$|e_1| = \left|\int_0^h f(y(t)) - f(y_0)\,\mathrm{d}t\right| \leq \int_0^h |f(y(t)) - f(y_0)|\,\mathrm{d}t.$$

Soient $L > 0$ et $\varepsilon > 0$ telles que f est lipschitzien de constante L sur la boule $B(y_0,\varepsilon)$. Si h est assez petit, alors pour tout $t \in [0, h]$, on a $y(t) \in B(y_0,\varepsilon)$, et donc

$$|e_1| \leq \int_0^h |f(y(t)) - f(y_0)|\,\mathrm{d}t \leq L\int_0^h |y(t) - y_0|\,\mathrm{d}t. \qquad (3)$$

De plus, (2) implique $|y(t) - y_0| \leq Mt$, où $M := \sup\{|f(y)| ; y \in B(y_0,\varepsilon)\}$ (noter que

$M \leq |f(y_0)| + \varepsilon L$ est bien finie) donc avec (3),

$$|e_1| \leq LM\int_0^h t\,dt = \frac{1}{2}LMh^2.$$

L'erreur au bout d'un pas de temps est donc de l'ordre de h^2. Par une variante de l'argument précédent, on peut montrer que l'erreur augmente à chaque pas de temps d'au plus $O(h^2)$. Pour t fixé, il faut $O(1/h)$ pas de temps, d'où finalement une erreur en $O(h)$.

Note historique. Euler a utilisé ce procédé pour l'étude de problèmes de calcul de courbes optimales, du type

$$\inf \int_0^1 L(y(t), y'(t))\,dt \; ; \; y(0) = y_0 \; ; \; y(1) = y_1,$$

avec $L : \mathbf{R} \times \mathbf{R} \to \mathbf{R}$. On a ensuite appelé cette classe de problèmes le calcul des variations, en référence à une approche de Lagrange, plus simple car elle évite tout argument de discrétisation. Cependant, la méthode d'Euler reste à la base des procédés d'approximation numérique de la solution de (1), dont on ne connaît pas de solution explicite dans la plupart des applications.

5.1.1. Fonction exponentielle

On peut être un peu plus précis dans le cas de l'équation différentielle $y' = y$, pour laquelle le schéma d'Euler s'écrit

$$y_{k+1} = y_k + hy_k = (1 + h)y_k.$$

1. Montrer que la convergence de l'approximation (que nous admettons) implique la formule

$$e^T = \lim_{n \to +\infty} \left(1 + \frac{T}{n}\right)^n.$$

   N.B. On peut éventuellement justifier la formule en prenant le logarithme de chaque membre, puis en effectuant un développement limité au premier ordre.

2. Soit y^h la fonction obtenue par interpolation linéaire entre les points (kh, y_k), $k = 1, \ldots$ Supposons par exemple $y_0 > 0$. Montrer (par récurrence) que $y^h(x)$ a une dérivée toujours inférieure à e^x, et donc est toujours inférieure à e^x.

3. Étude expérimentale de l'ordre d'erreur. Prenons $y(0) = 1$; vérifier que $(y^h(1) - e)$ est d'ordre 1 en h, soit $y^h(1) = e + Ch + o(h)$. Pour cette étude, on recommande de tracer $\ln(y^h(1) - e)$ en fonction de $\ln h$ (pourquoi ?). Comment obtenir une estimation de C avec ce graphique ? Combien de points faut-il pour avoir une estimation de cette constante ? Que se passe-t-il si h est « trop petit » ou « trop grand » ?

   (Pour ce dernier point : si h est trop petit, erreurs numériques excessives ; si h est trop grand, effets du second ordre).

Sur cet exemple élémentaire on peut aussi étudier la méthode du point milieu :

$$y_{k+1} = y_k + h\, f\left(\frac{1}{2}(y_k + y_{k+1})\right), \quad k = 1,\ldots$$

1. Montrer que si $f(x) = x$, ce schéma se ramène encore à une récurrence linéaire sur y_k, de la forme

$$y_{k+1} = \frac{1+\frac{1}{2}h}{1-\frac{1}{2}h}\, y_k.$$

2. Par une étude expérimentale, montrer que l'erreur est cette fois proportionnelle à h^2.
3. Avec la formule

$$\frac{1}{1-\frac{1}{2}h} = 1+\frac{1}{2}h+\left(\frac{1}{2}h\right)^2+\ldots$$

montrer que

$$y_{k+1} = \left(1+h+\frac{1}{2}h^2\right)y_k + \mathrm{O}(h^3).$$

4. Comparer la méthode d'Euler à la méthode du point milieu : simplicité de programmation, précision ; quelle est la plus efficace au total ?

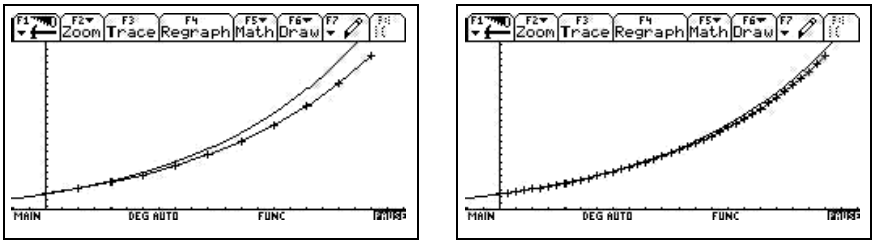

FIG. 7 – Solutions exactes et approchées avec des pas de 0,2 et 0,05

5.1.2. Oscillateur harmonique

Soit l'équation différentielle

$$z' = -y, \quad y' = z$$

dont les courbes intégrales sont les cercles centrés en 0 (parcourus dans le sens trigonométrique). On peut éliminer y en écrivant

$$z'' = -z \tag{4}$$

1. Interprétation mécanique : ressort de masse unité, une extrémité fixée en 0, longueur au repos négligeable, force de rappel proportionnelle et opposée à l'allongement.
2. Montrer la conservation de l'énergie $\frac{1}{2}(z^2+\dot{z}^2)$ (somme d'une énergie de déformation et de l'énergie cinétique).
3. Montrer que la méthode d'Euler, étendue au cas d'équations différentielles du second ordre, conduit au schéma :

$$z_{k+1} = z_k - hy_k, \quad y_{k+1} = y_k + hz_k.$$

Montrer que l'énergie du système discret augmente (par $\|\cdot\|$ on note la norme euclidienne) :

$$\|(z_{k+1}, y_{k+1})\|^2 = (1+h^2)\|(z_k, y_k)\|^2 = (1+h^2)^{k+1}\|(z_0, y_0)\|^2 .$$

Interprétation géométrique (les tangentes au cercle sont extérieures à ce cercle).

4. Montrer qu'on a la relation de récurrence suivante, portant seulement sur z_k :

$$\frac{z_{k+1} - 2z_k + z_{k-1}}{h^2} = -z_{k-1}$$

qu'on interprétera comme une discrétisation de (4).

5. Lien avec les suite récurrentes ; comparer les solutions analytiques des systèmes continus et discrets.

5.2. Un problème de commande

Dans les exemples précédents, on a examiné le comportement de la discrétisation par la méthode d'Euler, dans des cas où la solution analytique est connue ; ceci permet une estimation numérique des erreurs. De plus la solution était régulière (infiniment différentiable). Examinons maintenant des modèles simples (du premier ordre), mais où apparaissent des termes non linéaires, et une variable dite de commande qui peut varier dans un certain domaine.

On considère le système dynamique :

$$y'(t) = ay(t) + e(t) - c(t)$$

dans lequel $y(t)$ représente la population de poissons d'une lagune, $a > 0$ est le taux de reproduction, $e(t)$ un terme de perturbation (apports de l'océan) supposé connu, et $c(t)$ le prélèvement de pêche (la commande).

5.2.1. Modèle linéaire avec saturation

On suppose la commande $c(t)$ proportionnelle à la population de poissons, mais avec une saturation (limitation des moyens de pêche) :

$$c(t) = \min(by(t), c_{\mathrm{M}})$$

où $b > 0$. On suppose $e(t)$ constant.

1. Dans le cas où $e(t)$ est nul, sait-on calculer la solution de l'équation différentielle ? On distinguera suivant les valeurs de a, b et $y(0)$.
2. Même question, quand $e(t)$ est constant, strictement positif (ce qu'on supposera dans la suite).
3. On suppose $e(t)$ constant, strictement positif, et $a > b$. Discuter l'application de la méthode d'Euler, et donner une estimation d'erreur.
4. On suppose dans cette question $c(t)$ proportionnel au carré de la population de poissons :

   $$c(t) = b_2(y(t))^2$$

   où $b_2 > 0$. On notera la différence concernant le régime asymptotique, et on étudiera si le modèle discret a le même régime asymptotique.

5.2.2. Réaction tout on rien

Dans ce modèle, le pêcheur ne sort le bateau que s'il y a une quantité suffisante de poissons. On suppose $e(t) = 0$, et on prélève le plus possible si la population dépasse une quantité donnée :

$$c(t) = \begin{cases} 0 & \text{si } y(t) \leq y_d, \\ c_{\mathrm{M}} & \text{sinon,} \end{cases}$$

où $y_d > 0$, et $ay_d < c_{\mathrm{M}}$ pour assurer le régime asymptotique.

Le second membre étant discontinu, l'équation différentielle n'a pas de solution à dérivée continue en général. Nous n'essaierons pas de donner un sens mathématique précis à l'équation différentielle (on peut par exemple invoquer la théorie des opérateurs maximaux monotones[11]). Signalons cependant la possibilité de définir une solution comme limite de l'approximation obtenue avec le schéma d'Euler, si cette limite existe ; on peut en faire une étude numérique.

1. Construire une solution explicite, par raccordement des intervalles sur lesquels $y(t) < y_d$ et $y(t) = y_d$. On admet pour l'instant (et on le retrouvera dans les expériences numériques) que, si $y(t_0) = y_d$, alors $y(t) = y_d$ pour tout $t > t_0$.
2. Montrer que le système dynamique est *bien posé*, au sens de Hadamard : $y(t)$ dépend de façon continue de $y(0)$.
3. On considère l'approximation obtenue avec le schéma d'Euler. Montrer que le régime asymptotique est une oscillation autour de $x = y_d$, avec une amplitude en $O(h)$.
4. Enfin on pourra aborder la méthode d'Euler *implicite* définie par :

   $$y_{k+1} = y_k + h(ay_{k+1} - c(y_{k+1})), \qquad (5)$$

qui s'écrit ici

$$y_{k+1} = y_k + h(ay_{k+1} - c(y_{k+1})), \quad k = 1,\ldots,$$

pour laquelle, si la perturbation reste petite et si y_0 est proche de y_d, la suite y_k reste égale à y_d. La résolution de l'équation (5) est en soi un objet d'étude. Le grand avantage de la méthode implicite est qu'elle évite les oscillations de la méthode d'Euler explicite.

5.3. Méthode de Newton

On part de la procédure suivante :

Procédure *Newton*
Entrées : f fonction, a réel, ε réel positif
Sorties : α réel.
Initialisation : $u := a$,

Tant que $\left|\dfrac{f(u)}{f'(u)}\right| > \varepsilon$ **faire** $u \leftarrow u - \dfrac{f(u)}{f'(u)}$.

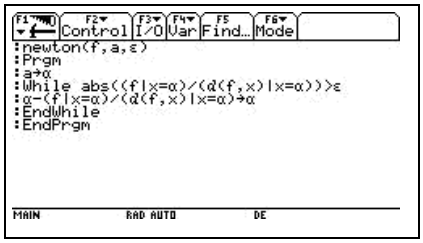

FIG. 8 – Implantation de la méthode de Newton sur une calculatrice du commerce

1. Cette méthode des tangentes (qui peut être illustrée graphiquement) conduit en principe à une approximation d'une racine de f.
2. Quelles sont les fonctions à connaître pour mettre en œuvre l'algorithme ? Évaluer f et sa dérivée en un point, faire le quotient, le comparer à ε et soustraire.

   Comme il s'agit à chaque fois de valeurs approchées (sauf cas particulier), il est crucial de les avoir avec une précision largement supérieure à ε.
3. Le test d'arrêt est-il convenable ? Il mérite d'être discuté !
4. Dans chaque cas particulier, par exemple si on dispose d'un intervalle où f est deux fois dérivable avec une dérivée seconde de signe constant, on peut améliorer la fiabilité du résultat en perfectionnant l'algorithme.
5. Il est cependant intéressant d'observer le comportement du système dynamique

$$u \leftarrow u - \frac{f(u)}{f'(u)}$$

défini par une fonction polynôme (par exemple à coefficients entiers) et d'un point a de départ.
Quelles racines trouve-t-on ? Quel est leur bassin d'attraction ?

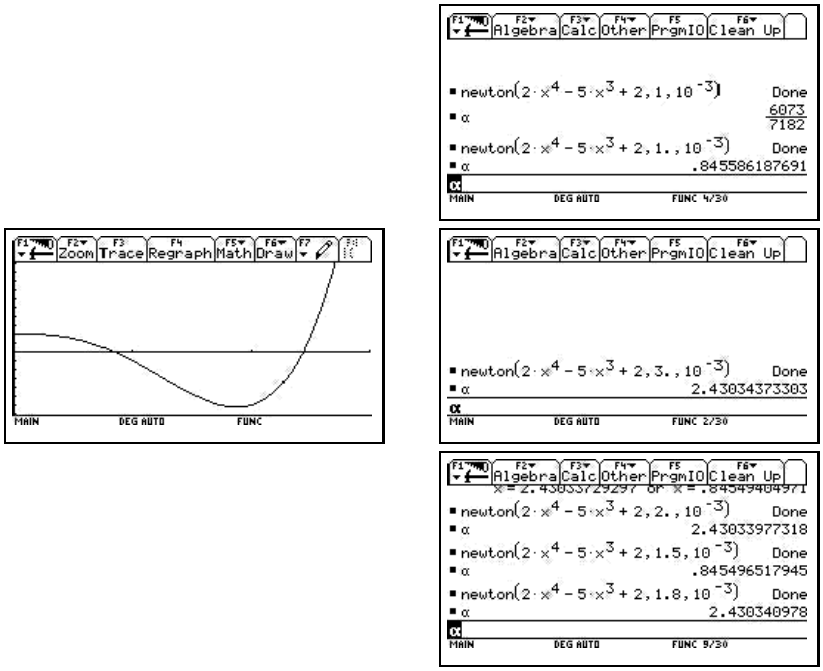

FIG. 9 – Un exemple : $f(x) = 2x^4 - 5x^3 + 2$

Remarque : Dans le cas où la méthode de Newton ne converge pas, on peut la modifier de la manière suivante.

Notons $d(x) := -f'(x)^{-1}f(x)$ le déplacement de Newton en un point x. On observe que, pour $\varepsilon > 0$, on a

$$\lim_{\varepsilon \to 0} \frac{f(x + \varepsilon d(x))}{\varepsilon} = f'(x)d(x) = -f(x)$$

est de signe opposé à $f(x)$; un petit déplacement dans la direction de Newton permet donc de trouver un point « meilleur ». On peut proposer une méthode de Newton avec réglage du pas :

$$u := u - \varepsilon \frac{f(u)}{f'(u)}.$$

L'idée la plus simple est de choisir comme ε le plus grand élément de la forme β^k, avec k entier naturel, tel que

$$f\left(u - \beta^k \frac{f(u)}{f'(u)}\right) < f(u).$$

En particulier, si $k = 0$, on retrouve la méthode de Newton.

5.4. Suites, séries

On peut expérimenter :

Procédure *Série harmonique*
Entrées : n entier.
Sorties : s réel.
Initialisation : $u := 0$

Pour i **de** 1 **à** n **faire** $u \leftarrow u + \dfrac{1}{i}$.

À partir de cet algorithme, peut-on conclure que la série harmonique diverge ? On pourra comparer avec la série suivante :

Procédure *Série divergente*
Entrées : n entier.
Sorties : s réel.
Initialisation : $u := 0$

Pour i **de** 1 **à** n **faire** $u \leftarrow u + \dfrac{i!}{100^i}$.

Systèmes dynamiques

1. Étudier de manière expérimentale les bassins d'attraction des points fixes du système dynamique $u \leftarrow f(u)$ (f est par exemple la fonction polynôme $x \mapsto x^3 - \dfrac{x^2}{2} - x + \dfrac{3}{2}$).
2. On donne la méthode de calcul suivante :

   Procédure *Syracuse* (ou *Collatz*)
   Entrées : $n > 0$ entier.
   Initialisation : $u := n$
   Tant que $u \neq 1$ **faire**
   si u pair **faire** $u \leftarrow u/2$,
   sinon faire $u \leftarrow \dfrac{3u+1}{2}$.

   La question est encore ouverte de savoir si, pour toute donnée de départ n, la procédure *Syracuse* s'arrête. On peut donc expérimenter librement… Par exemple, on peut écrire un programme qui a pour valeur le maximum des termes atteint à partir du point de départ n, ou le nombre d'itérations nécessaires pour revenir à 1, ou le nombre de changements de sens de variation (*i.e.* de parité), etc.

6. Statistique et probabilités : les aiguilles de Buffon

Le comte de Buffon (1707-1788), essentiellement connu comme naturaliste, fut également philosophe et mathématicien. Dans son *Essai d'arithmétique morale* (1777), on trouve le fameux problème de l'aiguille. Voici l'énoncé qu'il donne : *Je suppose que dans une chambre, dont le parquet est simplement divise par des joints parallèles, on jette en l'air une baguette, et que l'un des joueurs parie que la baguette ne croisera aucune des parallèles du parquet, et que l'autre au contraire parie que la baguette croisera quelques-unes de ces parallèles. On demande le sort de ces deux joueurs. On peut jouer ce jeu sur un damier avec une aiguille à coudre ou une épingle sans tête.*

En notant $2a$ la distance entre deux parallèles, $2b$ la longueur de l'aiguille, t l'angle de l'aiguille avec la direction des parallèles, et y la distance du milieu de l'aiguille à la parallèle la plus proche, il y aura intersection si et seulement si ($y + b\sin(t) \geq 2a$ ou $y - b\sin(t) \leq 0$).

La valeur aléatoire de y dans $[0 ; 2a]$ est distribuée de façon uniforme, ainsi que celle de t sur $[0 ; \pi/2]$. Les événements possibles sont donc paramétrés par le rectangle $[0 ; \pi/2] \times [0 ; 2a]$ muni de la loi de probabilité uniforme. L'événement « intersection avec l'une des parallèles » est paramétré par la région du rectangle $\{(t,y)$, $y \geq 2a - b\sin t$ ou $y \leq b\sin t\}$; cf. la figure 10.

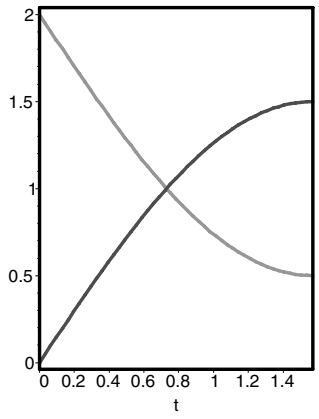

FIG. 10 – Buffon : Frontières des domaines du plan (t, y) pour $a = 1$, $b = 1.5$

On suppose $b \leq a$. Les deux événements $y + b\sin(t) \geq 2a$ et $y - b\sin(t) \leq 0$ sont disjoints (à un ensemble de probabilité nulle près). La probabilité pour qu'il y ait intersection est donc égale au rapport de l'aire

$$\int_0^{\frac{\pi}{2}} [2a - (2a - b\sin t)]\,\mathrm{d}t + \int_0^{\frac{\pi}{2}} b\sin t\,\mathrm{d}t = 2b.$$

à l'aire du rectangle $[0 ; \pi/2] \times [0 ; 2a]$.La probabilité cherchée est donc $p = \dfrac{2b}{\pi a}$.

Dans le cas particulier $a = b$ (la longueur de l'aiguille est égale à la largeur des lames

du parquet), on obtient $p = \frac{2}{\pi}$.

Si $b > a$, les deux événements $y + b\sin(t) \geq 2a$ et $y - b\sin(t) \leq 0$ ne sont plus disjoints, la probabilité de l'événement ($y + b\sin(t) \geq 2a$ ou $y - b\sin(t) \leq 0$) n'est plus la somme des deux probabilités.

Application : En simulant le lancé de n aiguilles de longueur égale à la largeur du parquet, la fréquence observée f des cas d'intersections est une approximation de $\frac{2}{\pi}$. En comparant les approximations décimales de π que nous connaissons avec le quotient $\frac{2}{f}$, on peut observer l'influence de n sur la précision obtenue.

Procédure *Calcul de* π
Entrées : n entier.
Sorties : p réel positif.
Initialisation : $i, j, t, y, p := 0, 0, 0, 0, 0$
Tant que $j < n$ **faire**

(nombre aléatoire compris entre 0 et $\frac{\pi}{2}$) $\to t$,

(nombre aléatoire compris entre 0 et 2) $\to y$
Si $y + \sin(t) \geq 2a$ **ou** $y - \sin(t) \leq 0$ **faire** $i + 1 \to i$

faire $p \to \frac{2n}{i}$.

Affichage des rationnels obtenus, et d'une approximation décimale.

(Une observation s'impose ici : les langages de programmations usuels offrent une primitive (souvent appelée « **random** ») de génération *pseudo*-aléatoire qui approxime efficacement un aléa véritable par une succession de transformations déterministes bien conçues initialisée par par un événement extérieur imprévisible, comme un temps d'horloge mesuré en microsecondes. Ces générateurs, pour peu qu'ils soient convenablement réalisés suffisent pour des simulations mettant, comme ici, seulement en jeu quelques millions ou quelques milliards d'appels. Au delà, on pourra puiser dans la très vaste littérature sur le sujet, notamment [Knuth, Chapitre 3].)

Cet exemple jouet pose deux types de questions. L'une est de nature statistique : *Quel sens peut-on donner à un résultat de simulation* ? L'autre est de nature logique et algorithmique : l'algorithme de simulation calcule le nombre π par simulation, mais il possède le défaut d'utiliser une fonction trigonométrique et la valeur de π elle-même : *Peut-on donner un algorithme de calcul de* π *par simulation qui n'utilise que les opérations arithmétiques usuelles* ?

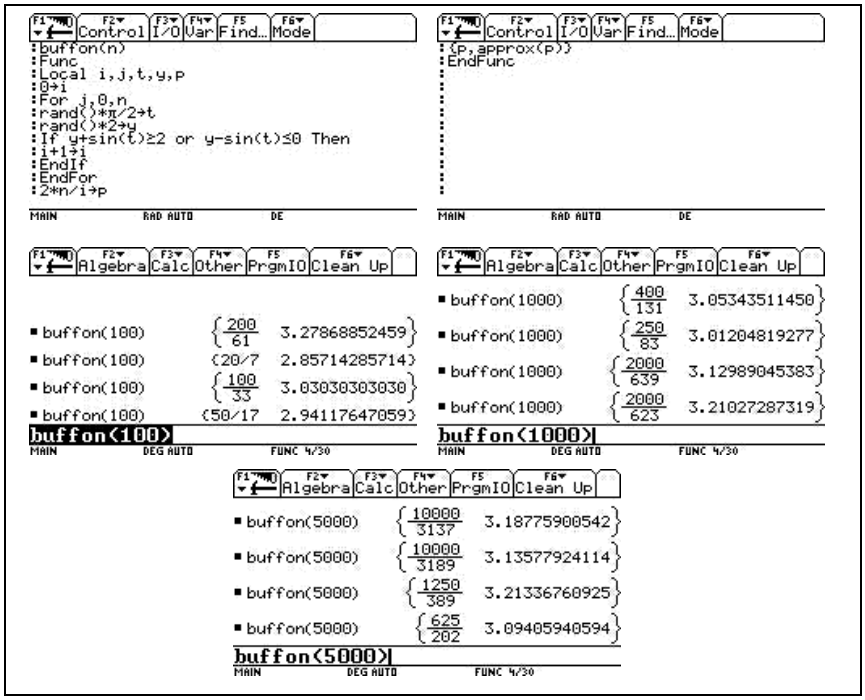

FIG. 11 – Un exemple d'implantation sur une calculatrice du commerce et quelques essais avec 100, 1000 et 5000 lancers d'aiguilles

6.1. Statistiques et fluctuations

L'observation des résultats montre quelques phénomènes statistiques simples. Clairement, les résultats de simulation fluctuent. Clairement aussi, augmenter le nombre de simulations *tend* à augmenter la précision du résultat. Ces deux phénomènes sont illustrés par la figure 12 qui montre l'évolution de l'estimation de π au cours du temps lors de 5 simulations avec n allant jusqu'à 5000.

Il importe d'abord de noter que la progression de précision si elle est attendue « en général » n'a rien d'obligatoire. Ainsi, l'une des simulation de la Figure 11 avec $n = 100$ donne-t-elle la valeur approchée $\pi^0 = 3.0303$ qui est meilleure que l'une des simulations obtenue avec $n = 1000$ pour laquelle $\pi^0 = 3.0120$. L'observation de l'ensemble des quatre simulations pour $n = 100, 1000, 5000$ montre néanmoins une tendance à l'amélioration résumée par le tableau suivant :

n	min	max	moyenne	(erreur)
100	2.8571	3.2786	3.0268	–3.7%
1000	3.0120	3.2102	3.1014	–1.3%
5000	3.0940	3.2133	3.1577	+0.5%

Qualitativement, l'erreur apparaît ici comme décroissant d'un facteur 3 environ quand on passe de $n = 100$ à $n = 1000$, puis d'un autre facteur d'environ 2 en passant

de $n = 1000$ à $n = 5000$. Si le signe de l'erreur est imprévisible, la décroissance de l'erreur attendue obéit, elle, à des principes bien connus depuis plus de deux siècles, grâce à De Moivre, Gauss, et Laplace.

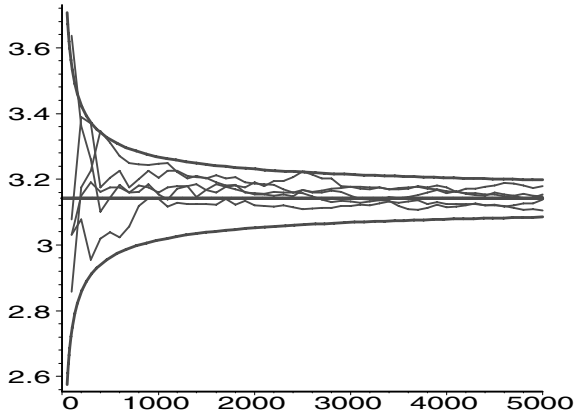

FIG. 12 – L'évolution au cours du temps de cinq simulations ($n = 5000$)

L'estimation de $\frac{2}{\pi}$ est obtenue en comptant la proportion de « succès », c'est-à-dire le nombre de fois où l'aiguille intersecte une parallèle lors de n essais, rapporté à n. On constate facilement que ce nombre de fois est une variable aléatoire binomiale équivalente au lancer n fois d'une pièce biaisée, la probabilité étant $\frac{2}{\pi}$ de tirer face (par exemple). Soit X_n cette variable aléatoire : on a tout d'abord le fait que la variance de X_n est proportionnelle à n. De manière équivalente, l'écart type de X_n est de l'ordre de $\sqrt{n}$ et celui de $\frac{X_n}{n}$ est de l'ordre de $\frac{1}{\sqrt{n}}$. C'est cet écart type qui mesure la dispersion des résultats : *la précision de l'estimation de* $\frac{2}{\pi}$ *lors de n essais est « très probablement » de l'ordre de* $\frac{1}{\sqrt{n}}$. Ceci explique bien les données numériques de la figure 11 ainsi que le diagramme de la figure 12 où les deux courbes continues représentant les fonctions

$$f_-(x) = \pi - \frac{4}{\sqrt{x}}, \quad f_+(x) = \pi + \frac{4}{\sqrt{x}},$$

résument la tendance générale des estimées à converger au cours d'une simulation. Plus finement, la distribution de probabilités de X_n est connue,

$$\Pr(X_n = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad p = \frac{2}{\pi}, \quad \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

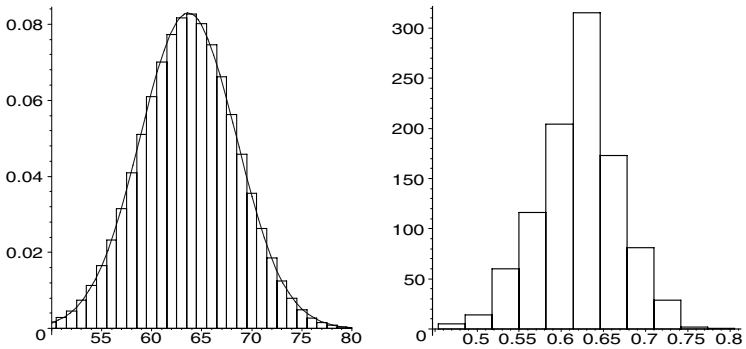

FIG. 13 – La loi binomiale pour $n = 100$ et son approximation normale (à gauche) ; les fréquences observées des estimations pour 1000 simulations correspondant à $n = 100$ (à droite)

Cette loi de X_n, lorsque n devient grand approche la loi de Gauss. Ainsi un tirage de X_n fluctue-t-il, aléatoirement certes, mais selon une règle bien définie. La figure 13 montre l'histogramme de la loi de X_n comparé à la loi de Gauss, ainsi que ce que l'on observe en effectuant un lot de 1000 simulations correspondant à $n = 100$. Noter que $\frac{2}{\pi} = 0.63661\ldots$ et que le passage de $\frac{X_n}{n}$ (qui estime $\frac{2}{\pi}$) à $\frac{2n}{X_n}$ (qui estime π et est aussi asymptotiquement normal) déborde sensiblement du cadre du programme.

EXERCICE : (À réaliser par simulation et à discuter analytiquement.) On considère une élection entre deux candidats A et B où l'un des candidats bénéficie de 48% d'intentions de vote, l'autre de 52%, mais on ignore ces faits et l'on ne sait pas qui a l'avantage. Quelle taille d'échantillon un institut de sondage doit-il prendre pour avoir environ 9 chances sur 10 de prédire correctement le gagnant ?

6.2. Algorithmique de la simulation

Les méthodes de simulation sont importantes dans diverses activités des sciences de l'ingénieur : simulation de phénomènes physiques, de trafic routier, de répartition de tâches dans les systèmes informatiques, de calculs d'intégrales en dimension élevée, de problèmes d'optimisation combinatoire difficiles, etc. Ce domaine donne lieu à une algorithmique riche. On examinera simplement ici un problème suscité par l'aiguille de Buffon : comment tirer $y = \sin(t)$ lorsque t est uniforme dans $(0 ; \pi/2)$, ce sans faire appel à la valeur de π ni aux fonction trigonométriques. Le résultat est un programme de « pur » calcul de π par simulation à partir des opérations arithmétiques de base complétées par la racine carrée.

Il est clair que tirer $\sin(t)$ pour t uniforme sur $(0 ; \pi/2)$ revient à choisir uniformément un point sur le cercle unité et en extraire la seconde coordonnée. Choisir un tel point peut se faire en choisissant un point $P = (x, y)$ uniformément et au hasard à l'intérieur du cercle unité, puis à prendre l'intersection du rayon vecteur $\overrightarrow{OP}$ avec le cercle unité. Tirer P lui-même s'effectue en tirant au hasard une paire (x, y) où x, y sont uniformes et indépendants dans le carré $[0 ; 1] \times [0 ; 1]$, et en rejetant les points qui ne conviennent pas. Le programme est alors :

Procédure *Tirage de* $\sin(t)$ *pour t uniforme dans* $[0 ; \pi/2]$
répéter
  tirer x, y uniformément dans $[0 ; 1]$
jusqu'à $(x^2 + y^2) \leq 1$ **et** $x^2 + y^2 \neq 0$;

retourner $\dfrac{y}{\sqrt{x^2 + y^2}}$.

Exercice : Écrire une procédure complète d'estimation de π par simulation de l'aiguille de Buffon qui n'utilise que les opérations $+$, $-$, $\times$, $\div$, $\sqrt{\ }$.

Ces méthodes de rejet s'avèrent utiles pour le calcul d'intégrales ou de volumes de corps dans des espaces à un grand nombre de dimensions. Elles sont alors connues sous le nom de méthodes de Monte-Carlo (en l'honneur de son casino).

Exercice : Évaluer l'aire sous le quart de cercle en adaptant la procédure de tirage de $\sin(t)$. Pour 1000 appels au générateur uniforme, quelle méthode donne la meilleure estimation de π, l'aiguille de Buffon ou l'estimation de l'aire sous le quart de cercle ? Qu'est-ce qui paraît le plus rapide à précision donnée, en temps d'éxécution sur calculatrice ou ordinateur ? Pourquoi ?

7. Graphes : recherche de la distance entre deux sommets.

Des exemples d'utilisation des graphes sont vus en classe terminale.
On considère un graphe avec n sommets, certains de ces sommets étant reliés par des arêtes de longueur donnée. On fixe deux de ces sommets, et l'on cherche la longueur d'un chemin de distance minimale qui les joigne.
C'est un problème pratique très courant, que chacun a rencontré : toute carte routière est un graphe de ce type. Dès que le graphe a plus d'une dizaine de sommets, il n'est pas évident de trouver le plus court chemin.

L'algorithme le plus immédiat est de considérer tous les chemins joignant les deux sommets fixés, et de chercher le plus court. Cet algorithme est très inefficace : on peut bien sûr se limiter aux chemins sans cycles, donc de longueur bornée par le nombre de sommets, mais même comme cela, le nombre de chemins possibles reste très grand, en général exponentiel en le nombre de sommets.

Une méthode plus efficace consiste à procéder de proche en proche : si l'on trouve un plus court chemin entre les deux sommets qui passe par S_1, S_2, …, S_k, alors son début est aussi un plus court chemin du premier sommet à S_1, puis S_2, etc. On va

donc commencer par chercher un plus court chemin du premier sommet à un sommet quelconque, puis ajouter un nouveau sommet à chaque étape de l'algorithme.

Si l'on connaît déjà un ensemble S de sommets pour lesquels on a trouvé un plus court chemin, l'idée de l'algorithme qui suit est de regarder tous les sommets liés par une arête à un sommet de S, et de prendre parmi ces sommets le plus proche du sommet de départ. On résout ainsi l'exercice de proche en proche.

Pour écrire un programme, il faut choisir une structure de données. On supposera que les sommets sont numérotés de 1 à n, et que le graphe est représenté par une matrice A, de taille $n \times n$, où $A(i, j)$ est un réel positif qui donne la longueur de l'arête reliant i à j. S'il n'y a pas d'arête reliant les deux sommets, on posera par convention $A(i, j) = +\infty$. On va chercher le plus court chemin reliant 1 à n. On fait l'hypothèse que le graphe est connexe ; sans cela, si 1 et n ne peuvent pas être joints, la procédure ne termine pas. Il serait facile d'ajouter un test supplémentaire pour détecter ce cas.

On peut décrire informellement l'algorithme ainsi :

On initialise l'algorithme, en prenant 1 comme sommet courant, en initialisant un vecteur des distances D par les valeurs provisoires $D(1) = 0$, et $D(i) = +\infty$ pour tous les autres. Comme la valeur D(1) est évidemment minimale, on peut la marquer définitivement (à l'encre).

On répète la suite d'opérations suivante, jusqu'à ce qu'on ait marqué définitivement le sommet n :

- On note SC (*Sommet Courant*) le dernier sommet marqué à l'encre.
- Pour tout sommet i non encore marqué à l'encre, et relié à SC par une arête, on calcule la somme de la distance de SC et de la longueur de l'arête qui relie SC à i. On remplace l'ancienne distance $D(i)$ de 1 à i par la nouvelle distance si celle-ci est plus petite, sinon on laisse l'ancienne distance.
- Parmi tous les sommets non encore marqués à l'encre, on en choisit un de distance minimale, et on le marque à l'encre.

On continue ainsi jusqu'à avoir marqué n à l'encre. La distance calculée de 1 à n est alors la distance minimale cherchée.

De façon plus formelle, en utilisant une primitive **marquer** qu'il est facile d'implanter, on obtient le programme qui suit :

Procédure *Plus court chemin : algorithme de Dijkstra*
Entrées : n entier, A matrice $n \times n$, à valeurs réelles strictement positives ou $+\infty$.
Sorties : L réel.
Initialisation : SC := 1, D(1) := 0
Pour i **de** 2 **à** n **faire** $D(i) = +\infty$
Marquer(1)
Répéter
  calcul des nouvelles distances
  Pour i **de** 1 **à** n **faire**
    Si i **non marqué** $D(i) := \inf(D(i), D(SC) + A(SC, i))$
    recherche du plus proche parmi les points non encore marqués

L:= +∞
SC:= 0
Pour *i* **de** 2 **à** *n* **faire**
si (*i* **non marqué et** D(*i*) < L) **faire** L := D(*i*), SC := *i*
Marquer (SC)
Jusqu'à ce que *n* soit marqué
Quand n est marqué, L *est la longueur d'un plus court chemin joignant* 0 *à n.*

Cet algorithme est-il efficace?

Cet algorithme peut sembler bien complexe pour le problème que l'on cherche à résoudre. Il est utile d'estimer le nombre d'étapes qu'il comporte, pour pouvoir comparer avec d'autres algorithmes.

Il est clair qu'à chaque grande étape, on marque définitivement un sommet : il y a donc au plus n grandes étapes, si n est le nombre de sommets. Combien y a-t-il d'opérations élémentaires dans une étape ? On se contente de recalculer D pour les sommets adjacents au sommet courant, donc au pire n sommets, puis de comparer les sommets non encore marqués pour trouver le plus petit, encore au maximum n opérations. On voit que l'algorithme se termine en au plus $2n^2$ opérations (on peut raffiner l'estimation à $n(n-1)$, et même mieux si le degré maximum d'un sommet est majoré par $d < n$).

On a donc un nombre d'étapes qui est de l'ordre du carré de n ; si l'on compare avec l'algorithme naïf qui consiste à comparer tous les chemins, on voit que celui-ci est en général exponentiel en n, donc beaucoup plus long ; déjà pour des graphes de 50 sommets, ce deuxième algorithme est complètement inutilisable, alors que l'algorithme de Dijkstra est quasi-instantané pour un ordinateur.

Le problème des plus courts chemins appartient au domaine de l'*optimisation combinatoire*, appelé aussi « recherche opérationnelle ». Les algorithmes de graphe y jouent un rôle important. Des exemples se rencontrent dans les problèmes de transport (d'électricité, de marchandises), de réseaux de distribution, d'allocation optimale de ressources (problèmes d'emploi du temps, penser à l'affectation des équipages dans une compagnie aérienne), etc. Sur ces sujets, voir par exemple le livre *Graphes et algorithmes*, par M. Gondran et M. Minou (Eyrolles, 1979).

8. Quelques algorithmes abordables au Lycée

On a dressé ici une première liste de quelques algorithmes qui peuvent être envisagés à partir du programme actuel du Lycée(12).

8.1. Arithmétique

1. Génération de la liste des nombres premiers inférieurs à un entier par le crible d'Ératosthène
2. Recherche du PGCD de deux entiers par l'algorithme d'Euclide

(12) On pourra consulter : **J. Chabert et al.**, *Histoire d'algorithmes – Du caillou à la puce*, (Belin, 1994), ainsi que: *Algorithmique et traduction pour calculatrice*, (IREM de Grenoble, 2000).

3. Recherche du PPCM de deux entiers
4. Résolution de l'identité de Bézout par l'algorithme d'Euclide
5. Test de primalité
6. Conversion du système décimal au système binaire
   - avec un test systématique des entiers inférieurs à $\sqrt{N}$
   - amélioration : test avec 2, puis les entiers impairs inférieurs à $\sqrt{N}$
   - amélioration : test avec 2, puis 3, puis les entiers de la forme $6p - 1$ ou $6p + 1$ inférieurs à $\sqrt{N}$
7. Factorisation d'un entier en produit de facteurs premiers
   - avec un test systématique des entiers inférieurs à $\sqrt{N}$
   - amélioration : test avec 2, puis les entiers impairs inférieurs à $\sqrt{N}$
   - par recherche de a et b entiers tels que $N = a^2 - b^2$, ou encore par recherche de a pour que $a^2 - N$ soit un carré parfait
   - amélioration : algorithme de Fermat
8. Recherche de nombres amicaux
9. Recherches de nombres parfaits, abondants, déficients
10. Recherche de triplets pythagoriciens
    - entiers quelconques
    - avec la contrainte : un des 3 entiers est fixé ou le plus petit des 3 entiers est fixé

8.2. Analyse

8.2.1. Suites numériques

1. Calcul des termes d'une suite récurrente
2. Vitesse de convergence d'une suite récurrente monotone convergente

8.2.2. Approximation de réels

1. Calcul d'une approximation d'une fonction par une fonction affine
2. Recherche approchée d'une solution d'une équation différentielle du type $y' = ay + b$ par la méthode d'Euler
3. Recherche d'une aire sous une courbe par la méthode des rectangles ou des trapèzes
4. Régression affine

8.3. Statistique

1. Calculs de paramètres statistiques (moyenne, écart-type, médiane, quartiles, déciles)

2. Algorithmes de simulation :
   - Lancements répétés d'une pièce ou d'un dé, comptage des occurrences
   - Lancements répétés d'une pièce ou d'un dé truqués, comptage des occurrences
   - Lancements répétés de 2 ou 3 dés, calcul de la somme des faces, comptage des occurrences
   - Lancements répétés d'un dé, comptage du nombre de lancers jusqu'à l'apparition du 6
   - Réalisation d'un sondage sur un échantillon aléatoire
   - Marche aléatoire dans le plan ou dans l'espace

8.4. Mathématiques discrètes, graphes

1. Coloration d'un graphe
2. Recherche de la plus courte chaîne entre deux sommets

8.5. Divers

1. Génération du triangle de Pascal
2. Rangement d'une liste par ordre croissant
3. Mélange d'une liste de façon aléatoire
4. Résolution d'un système linéaire par la méthode de Gauss