L'OUVERT

JOURNAL DE L'A.P.M.E.P. D'ALSACE ET DE L'I.R.E.M. DE STRASBOURG
n° 59 – JUIN 1990 I.S.S.N. 0290 - 0068

NOTRE COUVERTURE :

Evanouissement du buste de VOLTAIRE *(1941) par Salvador* DALI. Sur ce thème DALI a peint plusieurs tableaux. Citons ici "Marché d'esclaves avec buste invisible de VOLTAIRE", mais reviennent très souvent dans son œuvre des images à deux ou plusieurs sens. Un groupe de personnages qui forme un visage, des cygnes qui se transforment en éléphant... Cela renouvelle à bien des égards des cubes concaves ou convexes.

MATHÉMATIQUES ET CULTURE

Dans le volume 3 (année 1990) des "*Annales de Didactique et de Sciences Cognitives*" (*) dont je ne saurais trop conseiller la lecture, on trouve un article de Jean-Paul FISCHER intitulé : "Pourquoi les élèves asiatiques surclassent-ils les américains (en maths)?"

L'étude de l'auteur est minitieuse et il passe en revue les différents facteurs qui peuvent expliquer ces meilleures réussites aux tests (de niveau primaire) des élèves du Japon, de Taiwan et de Corée par rapport aux élèves des U.S.A. Travail de recension, mais aussi et bien sûr de réflexion, J.-P. FISCHER avance trois raisons importantes :

* Le temps de travail, tant en classe qu'à la maison, est bien supérieur en Asie et, à la maison, les parents estiment ne jamais assez aider leur enfant, à tel point que devant l'échec de celui-ci ils mettent en cause son travail et non pas ses aptitudes.

* L'attitude des maîtres et des élèves : en Asie, les maîtres donnent beaucoup d'explications verbales sur un petit nombre de sujets, aux U.S.A. les maîtres laissent leurs élèves travailler indépendamment sur de nombreux problèmes mathématiques; corrélativement, on note une plus grande écoute des élèves asiatiques.

* Le facteur linguistique avec une très grande régularité du système de numération orale au moins pour le chinois et le japonais (ce qui compte beaucoup à l'école primaire).

Une réflexion franco-française sur cet aspect éminemment culturel de l'enseignement des mathématiques me paraîtrait judicieuse si on veut l'améliorer ou y intéresser une autre catégorie de la population telle que celle des filles. Ce sont des aspects qu'on a moins l'habitude d'envisager (pensons au rôle des parents), mais qui méritent d'être pris en considération.

J. LEFORT.

(*) Publication IREM de Strasbourg (prix de vente : 70 F port compris).

SOMMAIRE

N° 59 – JUIN 1990

L'OUVERT

ISSN 0290 – 0068

◇ *Responsable de la publication* : Jean LEFORT

◇ *Correspondance à adresser à :*

Université Louis Pasteur
Bibliothèque de l'I.R.E.M.
10, rue du Général Zimmer
67084 STRASBOURG CEDEX
Tél. : 88-41-64-40

◇ *Abonnement (pour 4 numéros annuels)*

50 F (95 F/2 ans) pour les membres A.P.M. d'Alsace
90 F (170 F/2 ans) pour l'Alsace
120 F (220 F/2 ans) pour la France ou l'Étranger.
Chèque à l'ordre de Monsieur l'Agent Comptable de l'U.L.P. (IREM)

◇ *Prix du numéro* : 25.– F

QUOI DE NEUF CONCERNANT LES TRIANGLES RECTANGLES?

(2^e partie)

Alain ROBERT

3.— NOMBRES CONGRUENTS

FERMAT a démontré que l'aire d'un triangle rectangle à côtés entiers n'est jamais un carré parfait $S \neq \square$. Il l'a fait par "*descente infinie*" dans la fameuse marge de son exemplaire du DIOPHANTE ... (une démonstration élémentaire se trouve dans J. ITARD, (cf. Bibliographie).

Similairement,

$S \neq 2 \ \square$ (VIÈTE, BACHET),
$S \neq 3 \ \square$ (LUCAS).

Considérons maintenant le triangle rectangle obtenu par la méthode de PYTHAGORE avec

$$a = 9, \quad a^2 = 81 = 2 \times 40 + 1, \quad b = 40 \text{ et } c = 41.$$

Son aire est

$$S = 9 \times 20 = 5 \times 36 = 5 \times \square \ .$$

En divisant par 6 les dimensions linéaires de ce triangle, on aboutit au triangle rationnel

$$a = 3/2, \quad b = 20/3 \quad et \quad c = 41/6$$

d'aire entière $S = 5$. Ce triangle avait déjà été construit par FIBONACCI (Leonardo PISANO) (1220). Comme on l'a vu, l'aire d'un triangle rectangle entier est un nombre pair, donc $S = 5$ ne peut être l'aire d'un triangle rectangle entier. Le résultat de FERMAT montre que $S \neq 1$ pour tout triangle rectangle rationnel (donc aussi $S \neq 4$ pour tout triangle rectangle rationnel). Avec les résultats de BACHET, VIÈTE et LUCAS, on voit donc que

5 est le plus petit entier qui apparaît comme aire d'un triangle rectangle à côtés rationnels.

Quels sont les entiers qui apparaissent comme aires possibles de tels triangles?

Classiquement, ces entiers sont appelés **nombres congruents**. Donnons-en une caractérisation arithmétique. La relation

$$c^2 = a^2 + b^2$$

montre que

$$c^2 \pm 2ab = (a \pm b)^2.$$

En travaillant avec des nombres rationnels a, b et c, on peut diviser par 4 cette relation et obtenir

$$(c/2)^2 \pm S = [(a \pm b)/2]^2.$$

Les nombres congruents sont les entiers $S > 0$ tels qu'il existe trois nombres rationnels α, β et γ avec

$$\gamma^2 - S = \alpha^2 \quad \text{et} \quad \gamma^2 + S = \beta^2.$$

Les trois carrés rationnels α^2, γ^2 et β^2 sont à égale distance S l'un de l'autre. Léonardo PISANO écrivait par exemple

$$(3 + \frac{1}{4} + \frac{1}{6})^2 + 5 = (4 + \frac{1}{12})^2,$$
$$(3 + \frac{1}{4} + \frac{1}{6})^2 - 5 = (2 + \frac{1}{3} + \frac{1}{4})^2.$$

On sait aujourd'hui que les équations :

$$\gamma^2 - 5 = \alpha^2 \quad \text{et} \quad \gamma^2 + 5 = \beta^2$$

n'ont que quatre solutions en nombres rationnels ayant des numérateurs et dénominateurs à moins de 50 chiffres! D'autre part, ces équations ont une infinité de solutions rationnelles. Depuis plusieurs siècles, les mathématiciens ont essayé de trouver les plus petits nombres congruents et EULER semble avoir été le premier à donner un triangle rectangle rationnel d'aire 7 : il s'agit de

$$a = 24/5, \quad b = 35/12 \quad \text{et} \quad c = 337/60.$$

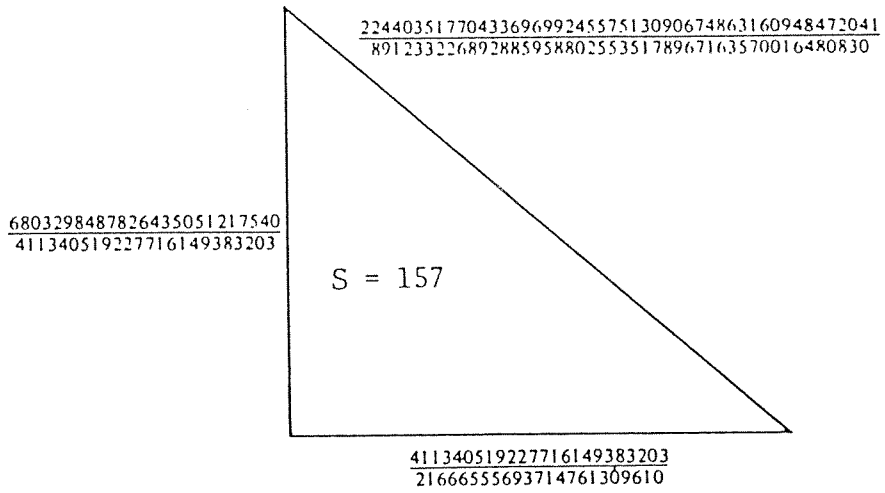
Les premiers nombres congruents sont ainsi

$$5, \quad 6, \quad 7, \ldots$$

Voici un triangle rectangle

$$a = 6 + \frac{3}{20}, \quad b = 13 + \frac{1}{3} \quad \text{et} \quad c = 14 + \frac{41}{60}$$

dont l'aire est $S = 41$. Donc 41 est un nombre congruent! On peut montrer que 157 est un nombre congruent : le triangle rectangle rationnel le plus simple ayant pour aire 157 a été déterminé par D. ZAGIER.



On conçoit bien maintenant la difficulté de déterminer les nombres congruents! On appréciera donc d'autant mieux le critère fourni par TUNNEL (Princeton) en 1983. Voici comment il s'énonce. Considérons les deux séries

$$\begin{aligned}\theta(x) &= \Sigma_{-\infty}^{\infty} x^{n^2} = 1 + 2\Sigma_{n>1} x^{n^2} = \\ &= 1 + 2x + 2x^4 + 2x^9 + \cdots \\ g(x) &= x\Pi_{n>1}(1 - x^{8n})(1 - x^{16n})\end{aligned}$$

puis les deux développements

$$\begin{aligned}g(x)\theta(x^2) &= x + 2x^3 + x^9 - 2x^{11} - \ldots = \Sigma_{n>1} a_n x^n, \\ g(x)\theta(x^4) &= x + 2x^5 - x^9 - 2x^{13} + \cdots \\ &= \Sigma_{n>1} b_n x^n.\end{aligned}$$

Alors TUNNEL démontre le résultat suivant.

THÉORÈME. *Soit n un entier positif sans facteur carré. Alors :*

$$\begin{aligned}a_n \neq 0 &\Longrightarrow n \text{ n'est pas congruent}, \\ b_n \neq 0 &\Longrightarrow 2n \text{ n'est pas congruent}.\end{aligned}$$

Ainsi, par exemple, $a_1 = 1$ montre que 1 n'est pas congruent (FERMAT). De même $b_1 = 1 \Longrightarrow 2$ n'est pas congruent, $a_3 = 2 \Longrightarrow 3$ n'est pas congruent. Comme $4 = \square$ ce n'est pas un nombre congruent non plus!

Pour démontrer ce théorème, TUNNEL se base sur des résultats récents obtenus dans la théorie des courbes cubiques (aussi appelées courbes elliptiques (cf. ci-dessous)) par différents mathématiciens (SHIMURA, WALDSPURGER, GROSS, ...). Une célèbre conjecture formulée par BIRCH et SWINNERTON-DYER permettrait de démontrer que les conditions données par TUNNEL caractérisent entièrement les nombres congruents, justifiant l'appellation de "*critère*" parfois donnée au théorème de TUNNEL. Précisément cette conjecture permettrait de montrer qu'inversement

$$\begin{aligned} &n \text{ impair sans facteur carré et } a_n = 0 \Longrightarrow n \text{ congruent,} \\ &n \text{ pair sans facteur carré et } b_{n/2} = 0 \Longrightarrow n \text{ congruent.} \end{aligned}$$

Remarquons encore que l'ensemble des triangles rectangles rationnels est dénombrable. On peut donc en dresser une liste exhaustive en établissant pour chacun d'entre eux la surface. On obtiendra ainsi une liste de tous les nombres congruents ... Mais si n est un entier donné (positif sans facteur carré, par exemple 157) on se sait pas a priori combien de temps attendre pour le voir apparaître ou pour décider qu'il n'apparaîtra plus! A cet égard, le théorème de TUNNEL donne une condition extrêmement simple pour démontrer qu'un entier n'est pas congruent. Pour un entier n grand, on peut même remarquer qu'il existe un logiciel "*MacSyma*" permettant d'effectuer des calculs littéraux et qui serait capable de donner rapidement les coefficients a_n et b_n.

4.— RELATION AVEC LES COURBES ELLIPTIQUES

Le problème de savoir si un entier S (positif, sans facteur carré) est un nombre congruent se ramène à un problème diophantien sur une cubique. Voici pourquoi. Dire que S est congruent revient à dire que le système d'équations

$$x = \square \ , \quad x + S = \square \quad et \quad x - S = \square$$

possède une solution rationnelle $x \in \mathbb{Q}$. Lorsque ceci est le cas, il est clair que

$$x(x+S)(x-S) = x(x^2 - S^2)$$

est un carré rationnel > 0 et donc on peut trouver un couple (x, y) formé de nombres rationnels satisfaisant

$$y^2 = x(x^2 - S^2) \quad et \quad y > 0.$$

Autrement dit, on peut trouver un point à coordonnées rationnelles sur la cubique d'équation

$$y^2 = x(x^2 - S^2),$$

point qui ne soit pas sur l'axe des x. Inversement, tout point (x, y) à coordonnées rationnelles et $y > 0$ sur cette cubique fournit un triangle rectangle rationnel d'aire S selon les formules

$$a = \frac{|S^2 - x^2|}{y}, \quad b = \frac{2|x|S}{y} \quad \text{et} \quad c = \frac{S^2 + x^2}{y}.$$

Résumons

THÉORÈME. *Soit $S > 0$ un entier sans facteur carré. Alors les propriétés sont équivalentes :*

i) Il existe un triangle rectangle à côtés rationnels et d'aire S.

ii) S est un nombre congruent, i.e. il existe trois nombres rationnels α, β et γ avec

$$\gamma^2 - S = \alpha^2 \quad \text{et} \quad \gamma^2 + S = \beta^2.$$

iii) Sur la cubique d'équation

$$y^2 = x(x^2 - S^2)$$

il y a un point $P = (x, y)$ à coordonnées x et y rationnelles avec $y \neq 0$.

Par exemple, le point de coordonnées

$$x = 41^2/7^2, \quad y = 2^4 \times 3^2 \times 5 \times 41/7^3$$

est sur la cubique

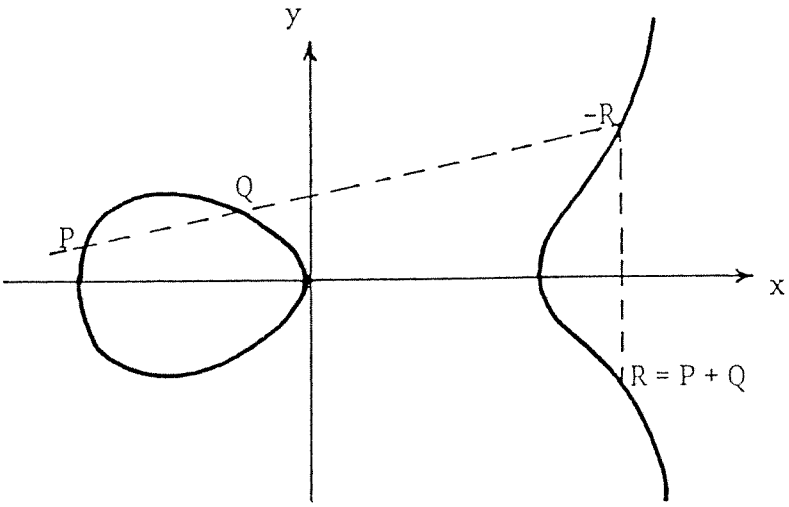
$$y^2 = x(x^2 - 31^2).$$

Donc 31 est un nombre congruent!

Revenons à l'étude d'une cubique de la forme

$$y^2 = x(x^2 - S^2)$$

où S est fixé. Il est facile de se faire une idée de l'ensemble des couples $P = (x, y)$ à coordonnées réelles sur cette cubique. Cette courbe est symétrique relativement à l'axe des x et on trouve deux points $(x, \pm y)$ au-dessus des valeurs de x rendant strictement positive $x(x^2 - S^2)$. On trouve donc des valeurs au-dessus des valeurs de $x \in [-S, 0]$ et $x \in [S, \infty]$.



On sait depuis le XIX^{e} siècle que si on ajoute un point à l'infini jouant le rôle d'élément neutre, l'ensemble des points (réels) sur la cubique est un groupe abélien pour la loi d'addition donnée par
a) $P_1 + P_2 + P_3 = e = (\infty) \Longleftrightarrow P_1, P_2$ et P_3 alignés,
b) $P = (x, y) \Longleftrightarrow -P = (x, -y)$.

En d'autres termes, la somme de deux points P et Q sur la cubique s'obtient en prenant le symétrique R du troisième point d'intersection avec la droite reliant P et Q (cf. dessin). Lorsque $P = Q$, la droite les reliant est naturellement remplacée par la tangente en P à la cubique : le point $2P = P + P$ est donc le symétrique du troisième point d'intersection de la tangente en P à la cubique.

Remarque. Lorsque S est un nombre rationnel, il est facile de voir que si P et Q ont des coordonnées rationnelles, il en est de même de $P + Q$ (même si $P = Q$) et l'ensemble des points à coordonnées rationnelles forme un sous-groupe de l'ensemble des points à coordonnées réelles sur la cubique. On a vu comment à tout point rationnel $P = (x, y)$ avec $y > 0$, on peut associer un triangle rectangle rationnel d'aire S. De plus, on a aussi montré comment à ce triangle rectangle rationnel, on peut associer un point rationnel sur la cubique. Les calculs montrent qu'on obtient en fait le point $2P$.

Résultats. *Supposons que S est un entier sans facteur carré. Alors le groupe des points rationnels sur la cubique $y^2 = x(x^2 - S^2)$ est un groupe de type fini isomorphe au produit du groupe d'ordre 4 : $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ par un groupe libre $\mathbb{Z}^r$ (r entier > 0).*

Le sous-groupe d'ordre 4 est constitué du point à l'infini (également neutre) et des trois points d'intersection avec l'axe des x. On trouvera donc un point rationnel sur la cubique avec $y > 0$ précisément lorsque $r > 0$, i.e. lorsque le groupe formé des points rationnels est infini. Les trois conditions du théorème ci-dessus sont

donc encore équivalentes à :

iv) Sur la cubique d'équation $y^2 = x(x^2 - S^2)$ *il y a une infinité de points à coordonnées* x *et* y *rationnelles.*

C'est la détermination de l'invariant r, rang du groupe des points à coordonnées rationnelles sur la cubique, qui est si difficile. On ne connaît pas d'algorithme effectif pour calculer r en fonction de S ... autre que celui donné par la conjecture de BIRCH et SWINNERTON-DYER (et qu'il serait même trop difficile d'expliquer ici ...).

Il est aussi très intéressant de considérer les points $P(x, y)$ à coordonnées complexes sur la cubique

$$y^2 = x(x^2 - S^2).$$

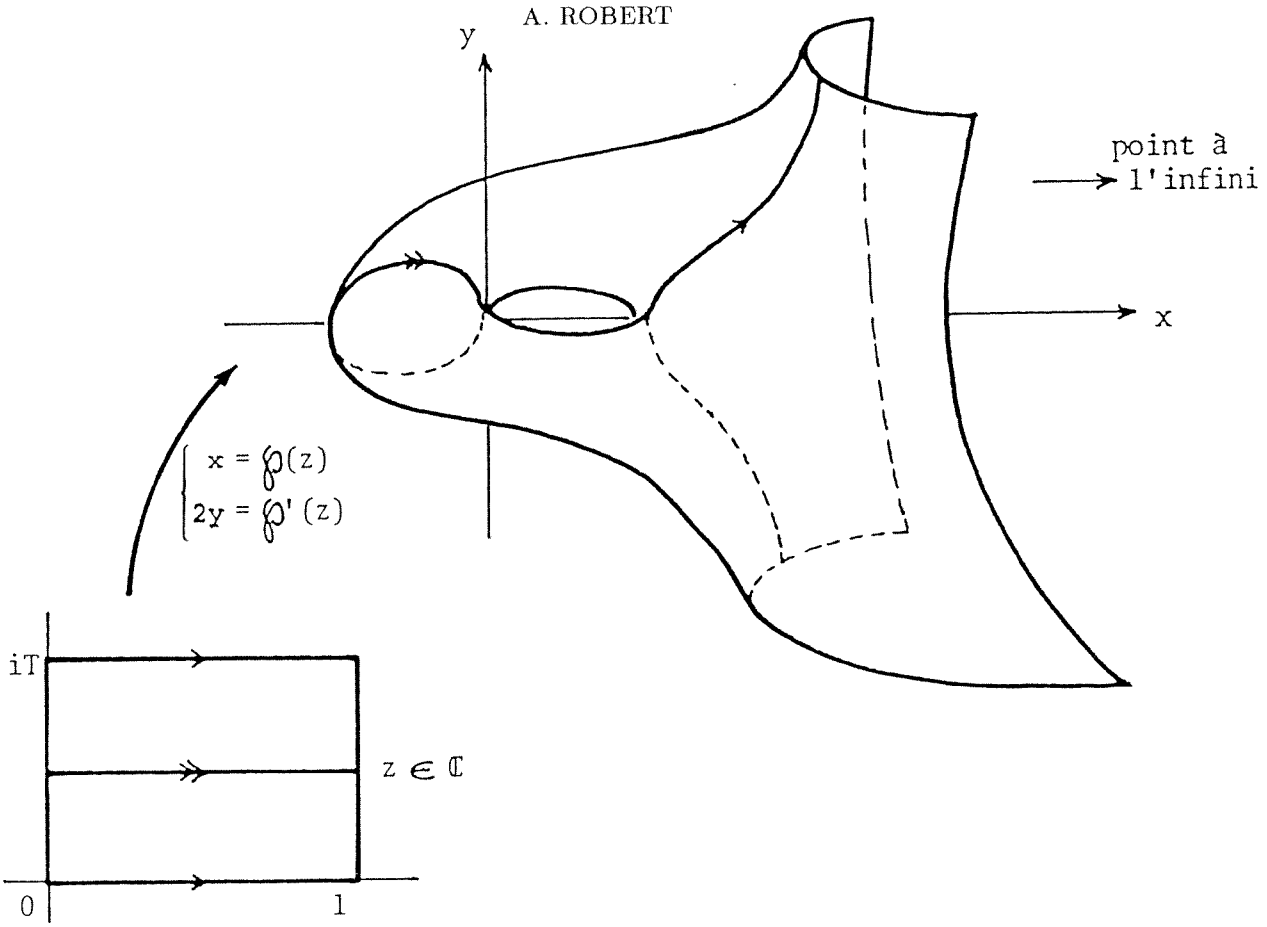
Ces points forment aussi un groupe dont la structure peut se déterminer par la paramétrisation suivante. Pour chaque $T > 0$, WEIERSTRASS a construit une fonction transcendante, classiquement dénotée par $\mathcal{P}$ qui est bi-périodique dans le sens suivant

$$\mathcal{P}(z + 1) = \mathcal{P}(z) = \mathcal{P}(z + iT)$$

et qui présente un pôle double en chaque point du réseau $\mathbb{Z} + iT\mathbb{Z} \subset \mathbb{C}$ (on devrait donc dénoter $\mathcal{P} = \mathcal{P}_T$). La dérivée de $\mathcal{P}$ a les mêmes propriétés de périodicité que $\mathcal{P}$ (mais présente des pôles triples où $\mathcal{P}$ avait des pôles doubles!). Il se trouve que

$$z \longmapsto (\mathcal{P}(z), \frac{1}{2}\mathcal{P}'(z)) = (x, y)$$

est une paramétrisation des points complexes sur une cubique $y^2 = x(x^2 - S^2)$ où S est une fonction de T. Dans cette paramétrisation, le point $z = 0$ (ou n'importe quel point z du réseau $\mathbb{Z} + iT\mathbb{Z}$) est envoyé sur le point à l'infini de la cubique et la somme dans $\mathcal{C}$ correspond à la somme sur la cubique par cette paramétrisation! Le groupe des points complexes sur la cubique est ainsi isomorphe au groupe additif $\mathcal{C}$ modulo le réseau $\mathbb{Z} + iT\mathbb{Z}$. C'est donc un tore (produit du cercle $\mathbb{R}/\mathbb{Z}$ par lui-même $i\mathbb{R}/iT\mathbb{Z}$). On peut même se représenter la situation intuitivement. L'ensemble des points complexes $(x, y) \in \mathbb{C}^2$ situés sur la cubique est une surface réelle dans $\mathbb{C}^2 \cong \mathbb{R}^4$. Par une projection convenable sur un sous-espace $\mathbb{R}^3$, on peut voir que cette surface est un tore topologique. On peut même prendre un sous-espace de dimension 3 contenant les deux axes réels $\mathbb{R} \times \{0\}$ et $\{0\} \times \mathbb{R}$ dans $\mathbb{C}^2$ et contenant donc le sous-espace $\mathbb{R}^2$ dans lequel on trouve les points à coordonnées réelles. Voici une illustration :



La situation est comparable à celle de la paramétrisation d'une équation $x^2+y^2 = 1$ (cercle) par les fonctions trigonométriques usuelles $x = \cos z$, $y = \sin z$. Ces dernières présentent aussi une périodicité importante!

WEISTRASS a d'abord rencontré les fonctions $\mathcal{P}$ en essayant de calculer la longueur d'un arc d'ellipse, d'où le nom de "*fonctions elliptiques*" donné à ces fonctions. Le terme "*elliptique*" a ensuite été appliqué aux courbes paramétrées par ces fonctions elliptiques : en particulier, toutes les courbes cubiques planes non singulières sont des courbes elliptiques.

C'est l'étude des courbes elliptiques qui a tant progressé depuis 1960 qui a permis à TUNNEL d'obtenir son critère remarquable!

RÉFÉRENCES

La théorie élémentaire des nombres, triples pythagoriciens entre autres, est remarquablement bien expliquée dans
G.-H. HARDY, E.-M. WRIGHT : *The Theory of Numbers.* Oxford at the Clarendon Press, nouveau tirage (1975).

QUOI DE NEUF CONCERNANT LES TRIANGLES RECTANGLES?

On trouve une démonstration du fait que l'aire d'un triangle rectangle entier n'est jamais un carré dans
J. ITARD : *Arithmétique et Théorie des Nombres.* Que sais-je? n°: 1093, P.U.F. (1967).

Voici la référence à l'article de base :
J.-B. TUNNEL : *A Classical Diophante Problem and Modular Forms of Weight 3/2.* Invent. Math 72 (1983) pp. 323–334.

Un survol de résultats préliminaires est donné dans
J. COATES : *The Work of Gross and Zagier on ...* Sém. Bourbaki, (Nov. 1984), exposé 635 (à paraître dans "*Astérisque*").

Voici un livre qui expose systématiquement toute la théorie utilisée par TUNNEL :
N. KOBLITZ : *Introduction to Elliptic Curves and Modular Forms*, Springer Verlag (1984), Graduate Text in Math. Nb. 97.

Un autre exposé introductif à la théorie des fonctions et courbes elliptiques est :
A. ROBERT : *Elliptic Curves*, Springer Verlag, Lect. Note in Math. N° 326, 2nd corrected ed. (1985).

MATHS SANS FRONTIÈRES

... ou une recette peu ordinaire pour un menu de compétition.

Recette :

Prenez 30 élèves du Collège Foch de Haguenau, de préférence; mettez-les dans une salle de classe.
Ajoutez-y un brin de compétition et une promesse de récompense. Remuez le tout, en l'assaisonnant d'un semblant de jeu.
Intégrez-y une pincée d'intérêt, un zeste de curiosité, sans oublier un soupçon de liqueur de pensée et une bonne dose de logique.
Vous obtenez alors une pâte enthousiaste, que vous ferez lèver grâce à un professeur de mathématiques stimulant.
Agitez régulièrement la préparation, 4 mois durant, en l'échauffant, le 4 décembre très exactement, à l'aide d'une séance d'essai. Découpez la, pour ce faire, en groupes autonomes. Poursuivez en même temps la cuisine mathématique habituelle.
Le 8 Mars, faites lui subir l'épreuve de la cuisson proprement dite : amenez, d'abord, lentement à l'échauffement, en regroupant bien tous vos ingrédients. Laissez mijoter, en attente, de longues minutes. Incorporez alors d'un coup, 12 exercices plus ou moins salés; l'un d'eux devra être accomodé obligatoirement à la crème anglaise. Laissez frémir puis bouillonner, en surveillant bien pendant deux heures.
Le tout reposera ensuite, durant plusieurs semaines. Le résultat sera servi, le 17 Mai, à la Salle des Douanes, accompagné éventuellement de la coupe de champagne du vainqueur.

(Voir l'article de R. JOST, page 19.)

Les belles histoires de Tonton Lulu

LES CARTES MARINES : HISTOIRE D'UN PROBLÈME POSÉ AUX MATHÉMATIQUES (*)

Jacques LUBCZANSKI

Si dresser une carte est un acte fondateur – la prise de possession symbolique d'un territoire – c'est aussi donner un outil au navigateur pour aller d'un point à un autre.

A ceux qui ne voient pas là un problème mathématique, je répondrai que jusqu'au XVIII^{e} siècle, tous les cartographes étaient mathématiciens; d'ailleurs l'histoire du problème de la cartographie maritine s'inscrit parfaitement dans l'histoire des maths, et donc au bout du compte dans celle des hommes.

Notre problème va commencer au cinquième siècle avant J.C. – en Grèce, bien sûr – et trouver une solution satisfaisante au XVI^{e} siècle en Europe. Cette solution est toujours utilisée de nos jours.

Et ce problème reste d'actualité, au moins sur le fond, puisqu'on doit lancer le satallite Hipparchos du côté de Neptune : c'est l'espace qu'il s'agit à présent de cartographier ...

OÙ EST LE PROBLÈME?

Comment les navigateurs se repèrent-ils?

Tout dépend de l'endroit où ils sont : tant qu'ils longent la côte, sans s'en éloigner au point de la perdre de vue, ils n'ont besoin que de points de repères terrestres : une bonne connaissance du rivage suffit au cabotage qui a été la première forme de navigation.

Mais dès qu'on ne voit plus la côte, et que l'horizon n'est plus fait que d'eau tout autour de soi – à propos, à quelle distance du rivage cela arrive-t-il? – il faut d'autres points de repère pour connaître sa position et sa direction.

La position d'un navire est donné par la latitude (hauteur angulaire au dessus de l'équateur) et par la longitude (déviation angulaire par rapport à un méridien donné). Mais si la latitude est facile à mesurer par rapport au soleil et aux étoiles, il n'en va pas de même pour la longitude : une mesure directe et précise à l'aide des étoiles est impossible à cause de la rotation de la terre sur elle-même. On mesure donc la longitude de façon indirecte, à partir de la vitesse et de la direction du

(*) Texte d'une conférence donnée à Nyon le 2 octobre 1989, dans un cours organisé par la Commission Romande de Mathématiques.

navire; et la mesure de la vitesse d'un navire a elle même posé problème pendant longtemps : le loch, qui mesure la distance parcourue date de 1577 et les premières horloges portables, à ressort, datent des années 1600.

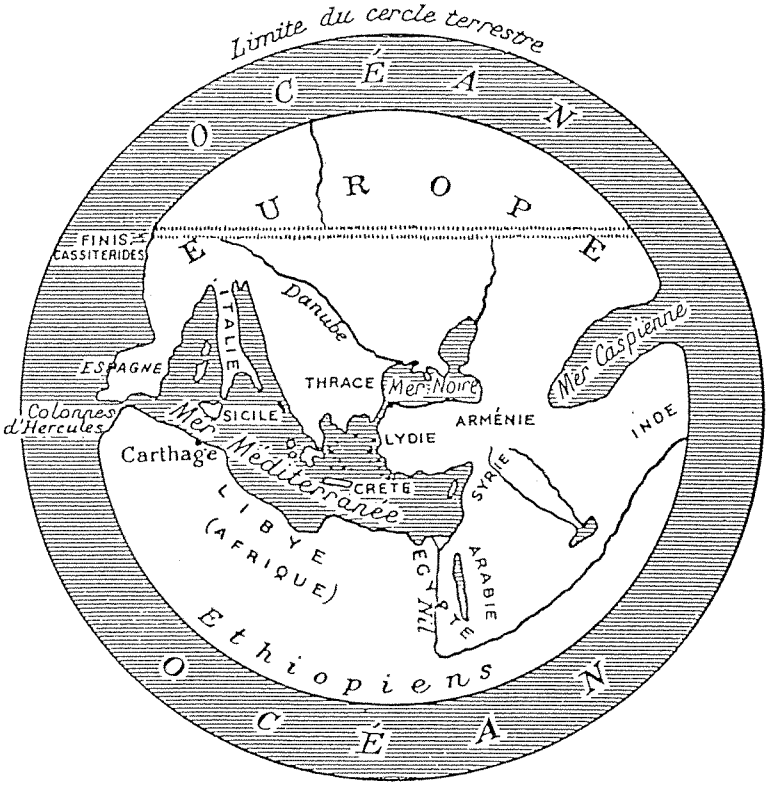
La direction d'un navire est donnée par l'angle du cap avec le nord, c'est-à-dire avec un méridien. Dans l'hémisphère nord, l'étoile polaire donne le nord (et dans l'hémisphère sud, c'est la croix du sud). La boussole apparaîtra vers 1200. Sur une carte, les lignes qui font un angle constant avec les méridiens s'appellent les lignes de Rhumb : en se fixant un "rhumb" précis, on suivra ces lignes.

Le problème de fond, d'un point de vue mathématique, est que la position et la direction d'un navire sont des notions "locales" alors qu'une carte maritime est une représentation "globale". Mettre la sphère à plat implique des déformations mais autorise des choix : que veut-on préserver sur la carte?

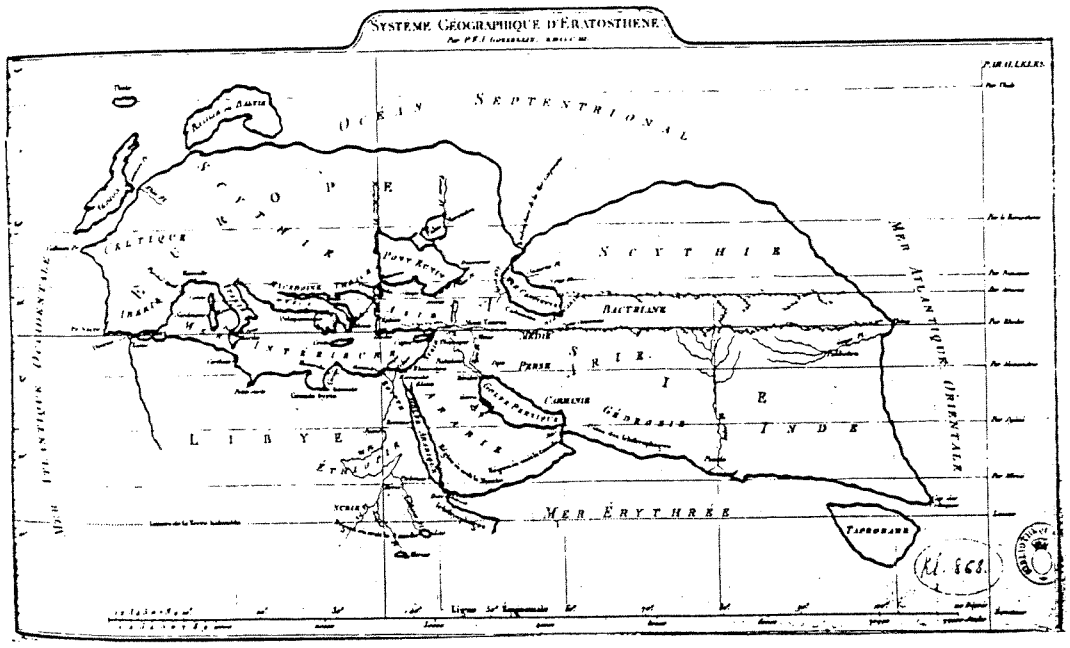
LES CARTES DE L'ANTIQUITÉ

Reprenons l'histoire à son commencement; pour les grecs de la fin de V^{e} siècle av. J.C., la terre est ronde. D'une part cela explique que l'horizon s'éloigne à mesure q'uon avance vers lui, et d'autre part cela satisfait les préoccupations esthétiques d'un PLATON, pour qui la sphère est la seule forme parfaite.

De façon naturelle, la terre est alors représentée par un cercle évoquant la rotondité du globe. Et, bien sûr, la carte est centrée sur la Grèce et ses voisins.



Carte du monde par HICATÉE *(de Milet)* 517 *av. J.C.*

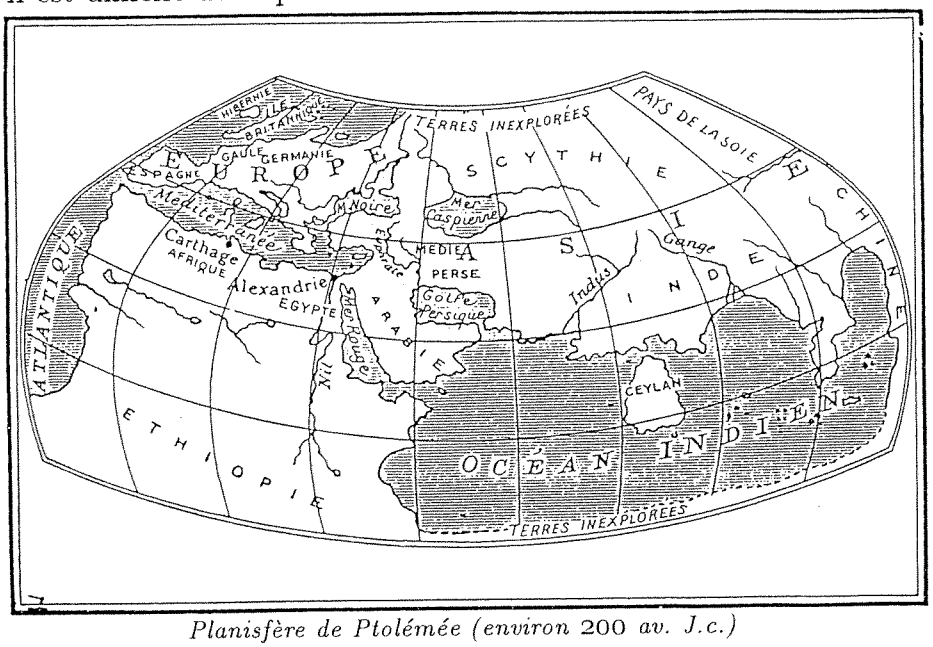


Carte d'Eratosthène reconstituée au XIX^e siècle

Il faut attendre plus de deux cents ans pour qu'on commence à mesurer, à calculer les distances et les positions des lieux connus. ERATOSTHÈNE imagine des lignes imaginaires, les parallèles, qui relient des points de même latitude.

Encore quatre siècles et PTOLÉMÉE vint : en utilisant les progrès de la trigonométrie, il trace une projection du globe, avec latitude et longitude. Et cette carte fera autorité pendant longtemps : c'est la première carte du monde imprimée au XV^e siècle.

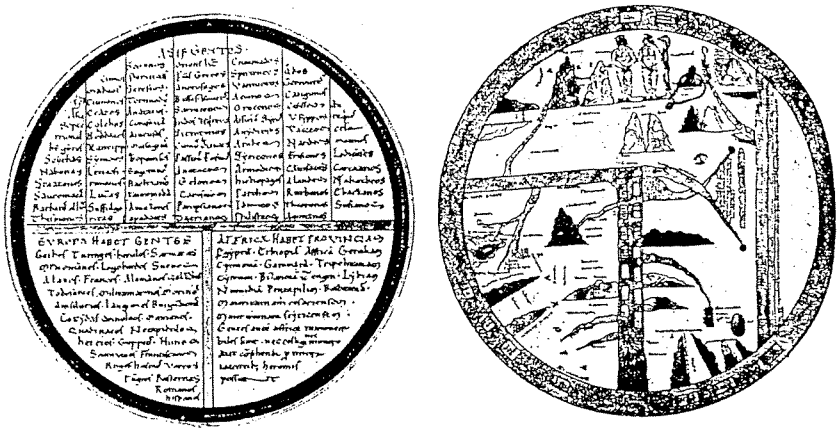
Mais la navigation dans le monde grec concerne avant tout la Méditerranée, sur laquelle il est difficile de se perdre ...



Planisfère de Ptolémée (environ 200 av. J.c.)

L'INTERMÈDE MÉDIÉVAL

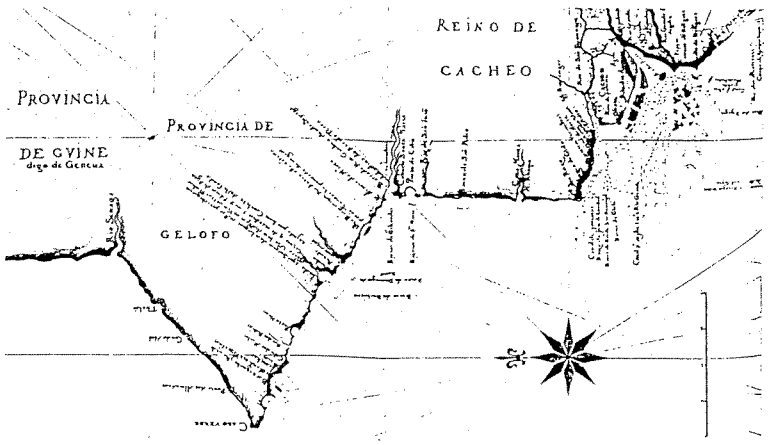
Pendant la période médiévale, ou l'Occident est sous la domination idéologique de l'Eglise, les cartes du monde sont d'abord des actes de foi; c'est l'interprétation des textes saints qui dicte la géographie et la cartographie. Par exemple, si la Bible dit que l'Océan recouvre 1/7 du globe, il doit en être de même sur la carte, même si la réalité est autre. Cela conduit aux cartes dites en "T sur O", en abrégé T/O, où le Nil, le Don et la Méditerranée forment en T sur le O qui délimite la Terre.



Mappemondes du XII^{e} siècle

Bien entendu les cartes en T/O n'étaient d'aucun secours pour les navigateurs. Cela n'affectait pas outre mesure les chrétiens qui naviguaient en Méditerranée, et qui employaient des pilotes arabes.

Par contre, quand les Portugais développaient une politique d'exploration et de colonisation, à partir du XIV^{e} siècle, leurs navigateurs créèrent leurs cartes : les Portulans sont des cartes basées sur des relations de voyages précédents : ce sont plutôt des itinéraires, des listes de lieux par lesquels il faut passer que des cartes sur lesquelles on peut repérer sa position et sa direction.



Atlas maritime portugais

MATHÉMATICIENS ET MARINS

Pour les mathématiciens, à partir du XVI^{e} siècle, une carte est une "projection de la surface du globe selon les lois de la perspective" (1). Ils en distinguent trois sortes :

— **carte stéréographique** : l'œil est au pôle, le plan de projection est le plan équatorial,
— **carte gnomonique** : l'œil est au centre, le plan de projection est un plan tangent à la sphère,
— **carte orthographique** : l'œil est à l'infini, le plan de projection est le plan équatorial.

Les cartes stéréographiques conservent les angles, les gnomoniques montrent les chemins les plus courts, les orthographiques permettent certaines mesures de distance... Mais aucune n'est assez globale ni assez pratique pour le marin.

Ce qui intéresse le marin c'est ce qu'il peut mesurer depuis son navire : en particulier l'angle de son cap avec le Nord, c'est-à-dire avec les méridiens, le plus pratique étant de naviguer à cap constant. En naviguant à cap constant, le marin suit les "lignes de Rhumb" : ce sont ces lignes qu'il veut retrouver sur la carte.

Les mathématiciens appellent les lignes de Rhumb des loxodromies. Ce ne sont pas des grands cercles sur la sphère : par exemple leurs projections sur une carte stéréographique sont des spirales équiangulaires (logarithmiques). Et donc pas les plus courts chemins d'un point à un autre. Mais c'est avec ça qu'on navigue : avec des arcs de loxodromies.

Donc une carte intéressante serait une carte où les rhumbs se lisent facilement, c'est-à-dire où les loxodromies seraient représentées par des droites. C'est l'idée de MERCATOR en 1569.

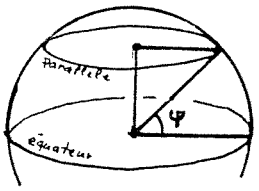
L'IDÉE DE MERCATOR

Voici, avant tout calcul, les idées qui sont à la base de la projection de MERCATOR :

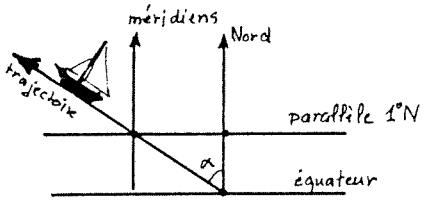
— les méridiens pointent tous vers le Nord : sur la carte ce seront donc des droites parallèles et équidistantes : les verticales,
— les parallèles sont toujours orthogonaux aux méridiens : ce seront donc les horizontales,
— les parallèles se retrouveront donc tous de longueur égale sur la carte : or, en réalité, ils sont de plus en plus courts à mesure qu'on s'approche du pôle. Pour préserver le rhumb (l'angle avec le méridien) sur la carte, on va les espacer de plus en plus.

(1) Encyclopédie Méthodique, 1790.

Précisons ce dernier point. Le parallèle situé à une latitude φ a pour longueur celle de l'équateur multipliée par $\cos\varphi$. Sur la carte, il a la même longueur que l'équateur : il est donc dilaté d'un facteur $1/\cos\varphi$.



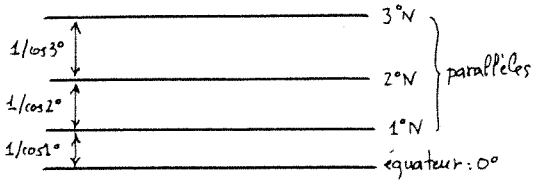
Vue du globe



Sur la carte

Par exemple, le parallèle $1°N$ est dilaté de $1/\cos 1°$ (qui est bien un facteur plus grand que 1 puisque $\cos 1° < 1$). Alors si on veut que l'angle α de la carte soit égal à celui qu'on mesure sur le bateau – le rhumb –, il faut que le triangle sur la carte soit semblable au triangle réel qu'il représente (2).

Pour que deux triangles soient semblables, il faut que leurs côtés soient proportionnels : si le côté "parallèle" est dilaté d'un facteur $1/\cos 1°$, il doit en être de même pour le côté "méridien" : autrement dit, le parallèle $1°N$ doit être "remonté". Et de la même façon pour le parallèle $2°N$, et les suivants ...



Sur la carte

Sur la carte, les parallèles sont donc de plus en plus espacés.

Sans tenir compte de l'échelle de la carte, le parallèle situé à la latitude n sera représenté à la hauteur $h(n)$ au dessus de l'équateur où $h(n) = \frac{1}{\cos 1} + \frac{1}{\cos 2} + \ldots + \frac{1}{\cos n}$.

On obtient par exemple :

$$h(30) = 31,55\ldots$$
$$h(45) = 50,71\ldots$$
$$h(60) = 75,96\ldots$$

(2) Le monde réel est sphérique mais sur une petite distance, il peut être considéré comme plat.

On ne peut pas aller jusqu'au pôle : il faudrait pour cela que la carte soit infinie. La carte s'arrêtera donc à une certaine latitude.

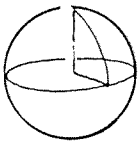
UN PEU DE SCIENCE FICTION

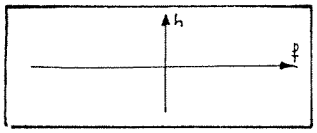
Pour le mathématicien, qui se situe volontiers hors du temps qui passe, tout cela peut ressembler à du bricolage. Pour le rassurer, habillons notre problème avec les vêtements que la science confectionnera plus tard :

30 ans après MERCATOR apparaissent les logarithmes, un siècle plus tard le calcul infinitésimal est au point; deux siècles après la notion d'application conforme, c'est-à-dire qui conserve les angles, apparaît.

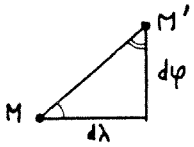
C'est bien de cela qu'il s'agit :

φ latitude	$\longmapsto$	f : latitude sur la carte
λ longitude		h : hauteur sur la carte

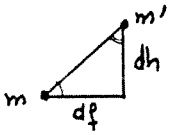




le globe $\longmapsto$ la carte



$\longmapsto$



un triangle infinitésimal $\longmapsto$ son image

L'application est conforme si et seulement si tout triangle infinitésimal est semblable à son image c'est-à-dire si : $\frac{df}{d\lambda} = \frac{dh}{d\varphi}$. Or, on a vu que (toujours sans tenir compte de l'échelle) : $\frac{df}{d\lambda} = \frac{1}{\cos\varphi}$.

D'où $\frac{dh}{d\varphi} = \frac{1}{\cos\varphi}$ et $h(\varphi) = Ln|\tan(\varphi/2 + \pi/4)|$ (3) (φ en radians).

On obtient par exemple :

$$h(30) = 0,5493\ldots$$
$$h(45) = 0,8813\ldots$$
$$h(60) = 1,3169\ldots$$

Quel rapport ont ces chiffres avec ceux de MERCATOR?

$$\frac{31,55\ldots}{0,5493\ldots} = 57,438\ldots \; ; \; \frac{50,71\ldots}{0,8813\ldots} = 57,533\ldots \; ; \; \frac{75,96\ldots}{1,3169\ldots} = 57,679\ldots \; ;$$

(3) N.D.L.R. : C'est ce qu'on appelle la fonction de MERCATOR!

Le rapport est sensiblement constant, et légèrement supérieur à 57. Pour le comprendre, utilisons un autre outil de science fiction – pour MERCATOR! – : les sommes de RIEMANN.

L'intégrale $\int_0^n \frac{d\varphi}{\cos\varphi}$, qui nous a donné la hauteur h pour n degrés, est approchée par la somme $\Sigma_1^n \frac{\pi}{180} \times \frac{1}{\cos p}$ (p en degrés) c'est-à-dire par $\frac{\pi}{180} \times \Sigma_1^n \frac{1}{\cos p}$, le coefficient $\frac{\pi}{180}$ correspondant à un pas de 1°. Or $\frac{\pi}{180}$ vaut 57,296 et $\Sigma_1^n \frac{1}{\cos p}$ est la hauteur calculée par MERCATOR.

n	l'intégrale	la somme	la tangente (à titre indicatif)
30	0,549	0,550	0,577
45	0,881	0,885	1
60	1,317	1,326	1,732

(La somme de RIEMANN est systématiquement en excès par rapport à l'intégrale à cause de la convexité de la fonction $1/\cos$.)

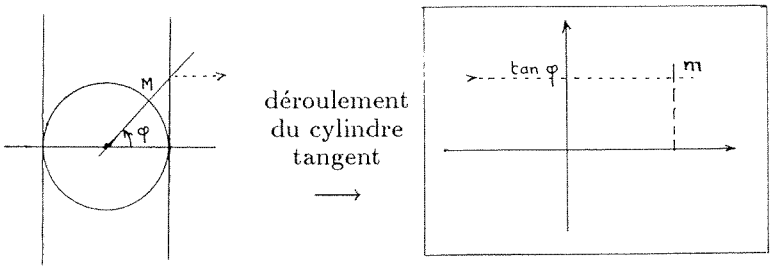
L'erreur commise est faible : se verrait-elle lors de la confection d'une carte? Sans doute pas.

Une conclusion s'impose : MERCATOR est un homme de son époque, mais dont l'idée rejoindra celle des mathématiciens du siècle suivant : CARALIERI, ROBERVAL, LEIBNIZ, NEWTON ...

ATTENTION AUX PROJECTIONS

Dans "projection de MERCATOR", le mot "projection" est trompeur : le système de MERCATOR n'est pas une projection au sens géométrique du terme : c'est une reconstruction par le calcul.

En particulier, contrairement à ce qu'on peut lire dans des ouvrages de bon aloi, ce n'est pas une projection cyclindrique tangente où le point de latitude φ est représenté à la hauteur $\tan\varphi$.



La colonne des valeurs de $\tan n$ dans le tableau ci-dessus diffère notablement des deux autres : c'est normal puisque les fonctions $\mathrm{Ln}|\tan(\varphi/2+\pi/4)|$ et $\tan\varphi$ sont distinctes. Ce qui se retrouve bien entendu sur les dérivées de chacune :

$1/\cos\varphi$ pour la première et $1/\cos^2\varphi$ pour la seconde.

Et la projection cylindrique tangente, si elle est facile à imaginer, ne serait d'aucune utilité pour retrouver les "rhumbs" sur la carte.

Méfions-nous des idées simplistes...

ET AUJOURD'HUI?

Comment se dirige un marin d'aujourd'hui ? Quelles cartes utilise-t-on pour trouver son cap ?

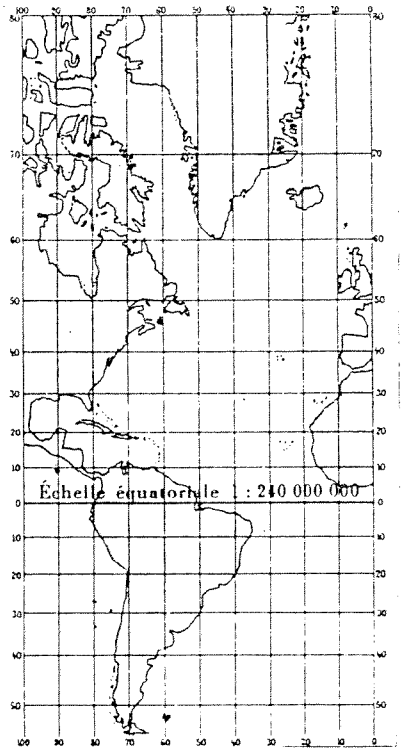
Principalement des cartes de MERCATOR, qui ont la préférence de la plupart des navigateurs, même si on sait faire des cartes plus sophistiquées. Mais il faut préciser que s'ajoutent à ces cartes deux autres outils performants :
— pour les itinéraires usuels : un annuaire donnant, pour chaque liaison, la liste des caps successifs à prendre;
— pour les itinéraires exceptionnels : le guidage par satellite, comme la balise Argos (mais il y en a d'autres), qui donne la position et la direction du navire quand on la demande.

La recherche de la "meilleure" route passe aussi par des impératifs météorologiques, économiques et géographiques. En particulier la navigation à cap constant emprunte les loxodromies qui ne sont pas les plus courts chemins : la différence en distance – et donc en carburant – peut atteindre 10% sur une traversée transatlantique. Mais on peut réduire cette différence en approchant la géodésique (chemin le plus court) par une suite d'arcs de loxodromies ...

Pour terminer, citons quelques problèmes mathématiques ayant des liens plus ou moins forts avec la navigation:

— liens historiques : la recherche des géodésiques sur une surface, au sein des géométries non euclidiennes;
— liens esthétiques : la description des rivages en termes de courbes fractales;
— liens technologiques : pour la navigation dans l'espace, le problème de la transmission des informations avec le moins d'erreurs possibles (par exemple : une photo envoyée de la fusée à la terre). Ce sont des problèmes d'algèbre – codes autocorrecteurs – ou de géométrie – épuration automatique des formes –, liés aux moyens informatiques utilisés.

Il ne restera plus alors qu'une question : "*quel est l'âge du capitaine ?*"



Une carte de MERCATOR

MATHÉMATIQUES SANS FRONTIÈRES

Rémy JOST

Un élève a posé une devinette :
Mon premier : 52 classes de troisième et 35 classes de seconde, soit près de 2400 élèves d'Alsace du Nord.
Mon deuxième : Un entraînement assidu et motivé.
Mon troisième : Un cocktail tonitruant d'exercices mathématiques au goût de chacun, même pour les linguistes.
Mon quatrième : De l'organisation, un esprit d'équipe et d'initiative.
Mon tout : Une compétition mathématique interclasse... hors classe!

La réponse est bien sûr :

"Mathématiques sans frontières" - Nord-Alsace 1990

qui est une compétition nouvelle interclasses pour les classes entières de 3ème et 2nde,
Organisée par l'Inspection Pédagogique Régionale de Mathématiques et l'Institut de Recherches sur l'Enseignement des Mathématiques de l'académie de Strasbourg.
Favorisant le travail d'équipe et les initiatives de liaison collège-lycée, l'ouverture des mathématiques aux langues étrangères.
Intéressant tous les élèves d'une classe à une activité mathématique diversifiée **désirant** susciter des vocations scientifiques, contribuer à faire bouger l'enseignement des mathématiques.

Quel type de compétition?

Il s'agit de confier à une classe entière la résolution et la gestion de 12 exercices en troisième et de trois de plus en seconde.
Un des exercices est à rédiger en langue **étrangère**.
La classe s'organise comme elle l'entend pour se répartir les tâches de recherche, de confrontation, de rédaction (1).
Au bout de deux heures, elle doit rendre douze (ou quinze) feuilles réponses.
Le professeur peut entraîner sa classe avant la compétition officielle. Il va surveiller une autre classe le jour J.

---

(1) Voir annexe 1.

Cette année : Mathématiques sans frontières : Nord Alsace 1990.

Pour lancer la compétition, la région Nord-Alsace a été retenue en 1990. La campagne d'information de septembre à octobre 1989 a permis l'inscription de 87 classes entières (52 classes de troisième et 35 classes de seconde) soit près de 2400 élèves au grand étonnement mais à la satisfaction de l'équipe d'organisation qui n'en attendait pas tant.
Une épreuve d'entraînement a été proposée à toutes ces classes en décembre 89. Le 8 mars 1990 de 14 heures et 17 heures a eu lieu l'épreuve officielle (2) dans l'enthousiasme général. Le 17 mai 1990 est prévue à Haguenau une distribution des prix festive en présence de Monsieur le Recteur de l'Académie de Strasbourg, des élus locaux, de la presse, des parrains de la compétition, des chefs d'établissements, des professeurs et naturellement des élèves des classes primées. Des lots de participation seront tirés au sort, chaque élève d'une classe primée bénéficie d'une part du lot attribué, que ce soit un voyage, une montre, une mallette, ... etc.
L'équipe d'organisation (3) est constituée de professeurs, d'inspecteurs et de chefs d'établissement. Elle a été aidée par tous les professeurs des classes inscrites qui ont également participé à deux réunions de bilan pédagogique.

L'année prochaine et ... après :

"Mathématiques sans frontières - Nord Alsace" continuera sur sa lancée. Une compétition analogue nommée "Mathématiques sans frontières - Haute Alsace" sera organisée dans le sud de l'Alsace. En 1992 les régions de Strasbourg et de Colmar pourraient elles aussi accueillir cette compétition interclasses et en 1993 toute l'Alsace serait couverte. Par ailleurs des contacts sont pris avec des établissements européens, allemands en particulier.
Une association a été créée (4) pour promouvoir tous ces ambitieux projets.
A voir l'engouement des élèves pour ces mathématiques, à entendre tous les encouragements de nos partenaires, "Mathématiques sans frontières" paraît répondre à une attente actuelle.
Mathématiques à vivre et à suivre ...

Pour l'équipe organisatrice : R. JOST

(2) Voir annexe 2.
(3) Font partie de l'équipe d'organisation de "Mathématiques sans frontières" en 1990 : Gérard BARBANÇON, Michel BARTHELET, Jean-Claude BIENAERD, Michel de COINTET, Jean-Paul GULLY, Pierre HUBER, Rémy JOST, Marie-Anne KEYLING, Eliane LEGRAND, Jean SAMSON, Henri SILVESTRE, Nicole VOGEL.
(4) Le prix de l'adhésion 1990 est fixé à 10 FF.

Pour tout renseignement sur la compétition ou sur l'association écrire à :
"Mathématiques sans frontières"
B.P. 214
F – 67506 HAGUENAU CEDEX

ANNEXE 1 : Ce qu'en pensent les élèves [1]

Quelques impressions d'élèves (Collège Foch de Haguenau) :

"...Le bilan de ce concours est déjà positif. Nous avons su nous responsabiliser en prenant en main notre travail. Nous n'étions, pour une fois, pas motivés par la seule ambition d'une note, mais nous nous confrontions à d'autres classes. Et même si nous n'emportons pas la victoire, nos efforts ont déjà été couronnés de succès, les maths n'ont pas été un casse tête solitaire mais un jeu d'équipe..."

"...La classe s'est organisée librement, par petits groupes équilibrés et mobiles. Grâce à cette épreuve nous avons découvert une nouvelle façon de travailler — chacun s'est senti utile et responsable — et une autre conception des mathématiques pour les jeunes. C'est une expérience étonnante, à renouveler!

Un extrait d'interviews d'élèves de 3ème (Collège Foch de Haguenau) :

...
Q : Quel genre d'entraînement?
R : En travail d'équipe. Nous nous sommes répartis par groupes équilibrés. Malgré nos différences de caractères et de talents.
Q : Et le jour de la compétition officielle?
R : Le jour du concours? Pas de problèmes, enfin nous avions tout de même le trac. Nous connaissions les règles du jeu et nous étions surveillés par un autre professeur.
Q : Vous aviez emporté des documents?
R : Seulement notre petit matériel, de quoi écrire, une calculatrice et des dictionnaires.
Q : Des dictionnaires?
R : Un dictionnaire général et un dictionnaire Français-Allemand; n'oubliez pas : Maths **sans** frontières. D'autres classes ont fait la même chose, mais en anglais ou en espagnol.
Q : Bien, dites moi ...chacun des groupes a-t-il progressé comme prévu?
R : Oui dans l'ensemble. Nos conditions de travail sont souples. Chaque membre du groupe peut rejoindre une autre unité si cela arrange la classe. Nous avons ainsi travaillé plus rapidement! et librement!

(1) Ces textes ont été réalisés en collaboration avec le professeur de français dans le cadre d'un P.A.E. interdisciplinaire.

Q : Précisez votre pensée.
R : Nous prenions des initiatives... Nous nous entendions bien; on nous a traité comme des adultes. Nous nous sentions responsables.
Q : C'est la confiance réciproque?
R : Oui, même si parfois c'est dur. Il a fallu de la patience mais maintenant on aime les maths.
Q : Tous?
R : Chacun à sa manière ... même ceux qui ne s'y intéressaient pas.
Q : Pourquoi?
R : Nous avons travaillé d'une autre façon, ensemble. Nous avions des objectifs précis. Chacun a contribué à la recherche des solutions.
Q : Et si jamais vous ne gagniez pas?
R : Ce serait dommage, mais ce n'est pas le plus important.
Q : Mais vous attendez les résultats?
R : Oui, oui! Bien sûr! Avec impatience! Mais le bilan est déjà positif.
Q : L'expérience a réussi?
R : Exactement ... puisqu'on considère les mathématiques autrement, avec plaisir ... on s'y intéresse, on les aime. Et nous sommes prêts à recommencer la compétition l'année prochaine en seconde. Les mathématiques sans limites!

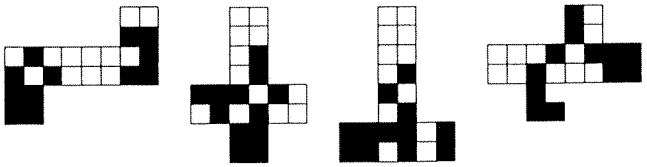
ANNEXE 2 : Sujets

Sujets d'entraînement

L'ENVERS DU DECOR

Un cube a été fabriqué à partir d'un matériau transparent, quadrillé, dont certaines cases ont été noircies.

Deux des figures ci-dessous sont l'intérieur et l'extérieur de ce même cube, développé de deux manières différentes. Lesquelles?



n° 1 n° 2 n° 3 n° 4

LE CHAT ET LA SOURIS

Une souris est à vingt pas de son trou. Un chat est à cinq bonds de la souris. Pendant que le chat fait un bond, la souris fait trois pas, et un bond de chat a la même longueur que dix pas de souris.

Le chat rattrapera-t-il la souris?

UNE GRANDE ET BELLE FAMILLE

Dominique affirme : "J'ai autant de frères que de sœurs". Sa sœur déclare : "J'ai deux fois plus de frères que de sœurs".

Combien y-a-til d'enfants dans cette famille?

LES FICELLES DU PHARAON

Voulant récompenser ses deux géomètres Ahmés et Thoutmés, qui avaient établi le cadastre du Royaume, Pharaon leur fit remettre deux très longs fils de lin de même longueur, un à chacun, et leur dit qu'il leur donnerait le domaine de la forme de leur choix qu'ils pourraient entourer de leur fil.

Thoutmés voulut un domaine en forme de triangle équilatéral, dont la superficie se trouva être de 20 dounams. Ahmès choisit pour sa part la forme de l'hexagone régulier :

Quelle fut la superficie de son bien?

Pharaon renvoya le premier pour manque d'ambition, puis le second quelques temps après, pour la même raison. Pourquoi?

EXERCICE 1
5 POINTS

SOMMET EUROPEEN

Les participants à une rencontre européenne ont échangé des poignées de mains. L'un d'entre eux a compté qu'il y a eu en tout 78 poignées de mains.
Sachant que chaque personne a échangé une poignée de mains avec chacune des autres, combien de personnes se sont serré la main ?
Rédiger la réponse en allemand, en espagnol ou en anglais.

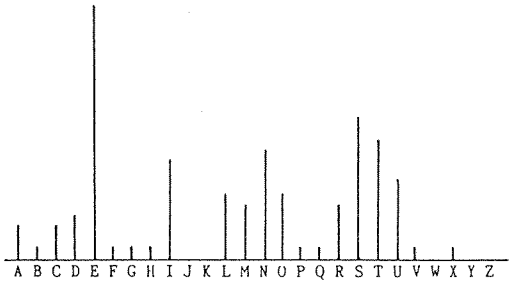
EXERCICE 2
5 POINTS

TOP SECRET

Voici une citation. Chaque lettre de l'alphabet a été remplacée par un signe. La ponctuation et les espaces ont été supprimés.

4+□ ● 5xx+□ △ → 3 □□+ △★+★ ● +x+! — → △★431 — + □ + ★ + ! — ! ! +△ ● — 53★□4+□ ● 3□★3△☆★35△□□5☆3 — 4+□ △ + ⊙ +!6+△★+★—+25△ ● ++□▼!+□!—4! ★ 343 ★ + ☆ 5 xx ! △ +

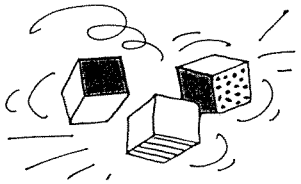
Décoder cette citation en utilisant le diagramme en bâtons qui indique la fréquence d'apparition de chaque lettre dans le texte.



EXERCICE 3
5 POINTS

ON EN VOIT DE TOUTES LES COULEURS

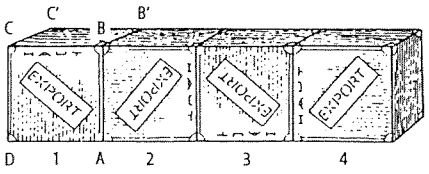
Trois cubes identiques sont posés sur une table. Les six faces de chaque cube sont de couleurs différentes.
Paul, Maurice et Albert regardent chacun un cube et énoncent les couleurs des trois faces qu'ils voient autour d'un sommet :
Paul : "bleu, blanc, jaune"
Maurice : "orange, bleu, rouge"
Albert : "vert, orange, blanc"
Quelle est la couleur de la face opposée à la face blanche ?
On ne demande pas de justification.



EXERCICE 4
5 POINTS

ET POURTANT ELLE TOURNE

On doit déplacer une caisse cubique de face avant ABCD et d'arêtes perpendiculaires à cette face [AA'], [BB'], [CC'], [DD'].
Pour cela, on la tourne autour de (AA') jusqu'à ce que (DC) soit horizontale, puis autour de (BB'), etc...
On a représenté sur une même figure la place initiale de la caisse ainsi que ses positions à la fin des trois premiers mouvements.
On demande de dessiner, dans le plan de la face avant de la caisse, la courbe suivie par le point B au cours des cinq premières rotations, la courbe suivie par le milieu de [AB] et celle suivie par le centre du carré ABCD.



EXERCICE 5
10 POINTS

LE GROS CHENE

Ce gros arbre a un tronc cylindrique de 4 mètres de circonférence. Un escargot l'escalade verticalement. Il est à 47 cm au-dessus du sol. De l'autre côté, sur la verticale diamétralement opposée, un autre escargot grimpe. Il ne lui reste plus que 3 cm pour être à 2 mètres au-dessus du sol.
Mais soudain, dans leur langage secret, nos deux escargots décident d'abandonner leur escalade et d'aller l'un vers l'autre par le plus court chemin.
Quelle distance chacun a-t-il parcourue à partir de ce moment-là sachant que la rencontre a eu lieu à mi-chemin ?

Toute solution, même partielle, sera examinée. Le soin sera pris en compte.
Ne prendre qu'une seule feuille réponse par exercice.

EXERCICE 6
10 POINTS

UNE DIVISION D'UN AUTRE AGE

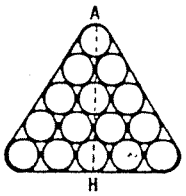
Au XIII[e] siècle on disposait la division de 4019 par 87 de la manière suivante :

QUOTIENT	4 4 4 4 4 6 4 6 4 6	QUOTIENT
DIVIDENDE	4019 4019 819 539 539 539 59 17	RESTE
DIVISEUR	87 87 87 87 87 87 87 87	DIVISEUR

Disposer de la même manière, sur un quadrillage, la division de 5435 par 76.

EXERCICE 7
10 POINTS

BOULES ET BILLARD

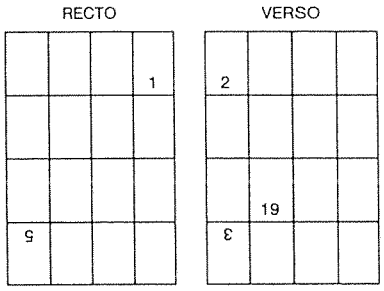


Avant une partie de "billard américain", les boules (toutes de même diamètre) sont serrées dans un cadre comme l'indique la figure.
La longueur AH mesure 169,6 mm. Quel est le diamètre d'une boule ?

EXERCICE 8
5 POINTS

BONNE IMPRESSION

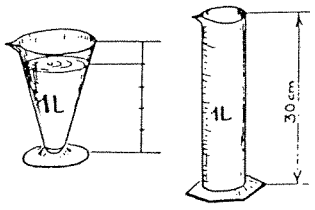
Un imprimeur fabrique un livre de 32 pages. Ces 32 pages sont imprimées sur une grande feuille avec 16 pages au recto et 16 pages au verso.
Cette grande feuille est ensuite pliée simplement en deux, quatre fois de suite.
On obtient ainsi un cahier de 16 épaisseurs de papier qui sera cousu suivant le dernier pli et massicoté sur trois côtés.
Sur le recto et le verso de la grande feuille représentés ci-contre, cinq numéros de pages ont été placés.
Recopier le recto et le verso de la grande feuille et y placer les 27 numéros manquants de façon que les pages du livre soient numérotées de 1 à 32, en bas, au milieu.



EXERCICE 9
10 POINTS

LE VERRE ET L'EPROUVETTE

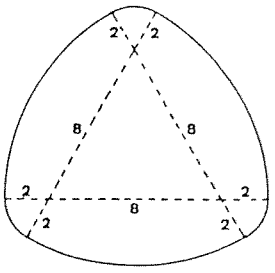
L'intérieur d'un verre de laboratoire a la forme d'un cône de révolution, celui d'une éprouvette a la forme d'un cylindre de révolution; leur contenance commune est 1 litre.
On remplit le verre aux 4/5[e] de la hauteur du cône et on transvase le liquide du verre dans l'éprouvette.
Sachant que la hauteur de l'éprouvette est 30 cm, calculer la hauteur du liquide dans l'éprouvette.



EXERCICE 10
15 POINTS

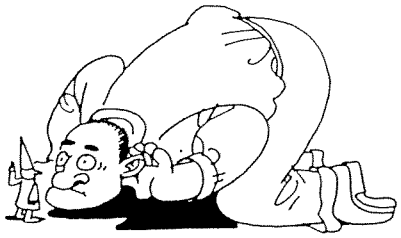
UNE ROUE A LA HAUTEUR

La figure ci-contre représente une courbe constituée d'arcs de cercles centrés aux trois sommets d'un triangle équilatéral et de rayons 10 cm ou 2 cm selon le cas.
Construire la figure en vraie grandeur. La découper.
La poser de différentes manières à l'intérieur d'une bande de papier de 12 cm de large, en la faisant rouler.
Constater que la courbe reste tangente aux deux bords de la bande de papier. Justifier.
Coller la roue sur la bande et le tout sur la feuille réponse.



EXERCICE 11
15 POINTS

LE PLUS GRAND DES NAINS ET LE PLUS PETIT DES GEANTS

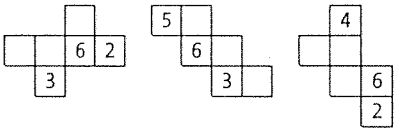


Soixante-douze nombres, tous différents, sont placés dans un tableau de 8 lignes et 9 colonnes.
On appelle nombre géant, le plus grand des nombres de chaque colonne : il y a 9 nombres géants.
On appelle nombre nain, le plus petit nombre de chaque ligne : il y a 8 nombres nains.
Comparer le plus grand des nombres nains au plus petit des nombres géants. Expliquer.

EXERCICE 12
5 POINTS

DES DES

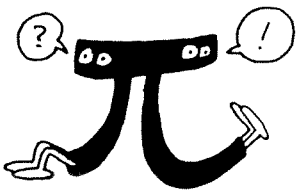
Chacun de ces patrons peut être plié pour former un dé. Sur chacun il manque trois numéros. La somme des points de deux faces opposées est, dans tous les cas, égale à 7.
Recopier ces patrons et inscrire sur chaque face le numéro manquant.



SPECIAL SECONDE

EXERCICE 13
5 POINTS

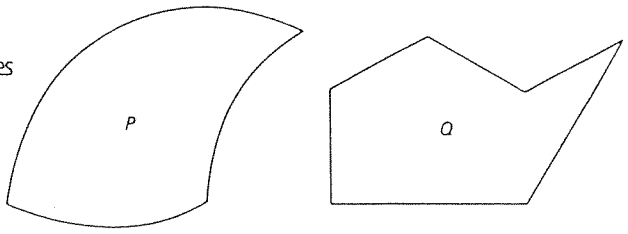
TOUT PRES DE π



Robert prétend que la longueur obtenue en ajoutant le côté du carré et le côté du triangle équilatéral inscrits dans un cercle de rayon 10 cm est égale à la longueur d'une demi-circonférence de ce cercle...à 0,5 mm près. A-t-il raison ? Justifier la réponse.

EXERCICE 14
10 POINTS

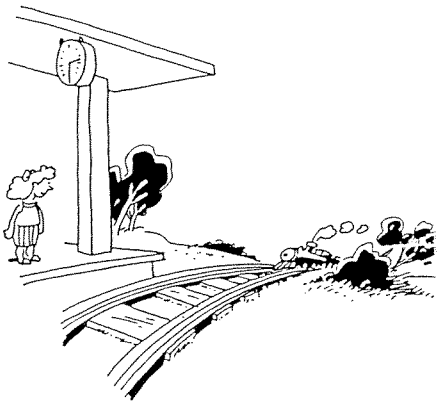
DEUX PAR DEUX



Un même procédé a permis de construire les figures P et Q de façon que chacune d'elles puisse être découpée en deux pièces superposables.
Reproduire P et Q et indiquer comment diviser chacune d'elles en deux pièces superposables.
Utiliser le même procédé pour construire une troisième figure R.

EXERCICE 15
15 POINTS

AU TRAIN OU ÇA VA



Il est 2 h 30. Le train de Frédérique entre en gare de Gérardvillé. Sa camarade Camille est venue la chercher. Frédérique lui raconte que son train a parcouru 100 km durant chaque heure de son trajet et que sa vitesse moyenne a pourtant été de 104 km/h. Camille refuse de la croire. Frédérique insiste. Elle dit qu'elle est partie à minuit (0 heure), que le train ne s'est pas arrêté avant Gérardvillé, que de 0 h 30 à 1 h et de 1 h 30 à 2 h il a roulé à la vitesse constante de 80 km/h alors que le reste du temps il a roulé à la vitesse constante de 120 km/h.
Représenter graphiquement la distance parcourue par le train en fonction de l'heure. Frédérique a-t-elle raison ou non ? Expliquer pourquoi.

NOUS AVONS LU

"Aventures mathématiques"

par M. de GUZMÁN aux Presses Polytechniques romandes.

Vulgariser les mathématiques est un art et les divertissements, récréations ou amusements mathématiques, forment un fond commun dans lequel tout bon vulgarisateur puise. De temps en temps seulement, un nouvel exercice apparaît et c'est pourquoi il ne faut pas lire le livre de Miguel de GUZMÁN pour y trouver beaucoup de nouveautés, non! il faut le dire pour réfléchir à la façon dont il a classé les exercices : selon les méthodes de démonstration : la descente infinie de FERMAT, l'utilisation des transformations, le principe de DIRICHLET, ... chaque chapitre se terminant par une note historique qui peut être soit une courte biographie, soit une réflexion sur le développement de tel domaine des mathématiques. C'est l'occasion d'avoir une vision un peu globale des maths, vision que l'on peut mettre entre les mains des lycéens, ce qui n'arrive pas tous les jours.

Il est regrettable qu'un tel ouvrage, qui est déjà traduit en de nombreuses langues, n'ait pas eu, pour la version française, un traducteur compétent. Que de barbarismes, voire de contre-sens! Le lecteur, même non-hispanisant, pourra sans doute corriger cela de lui-même mais c'est désagréable.

"Arithmétique"
Cours, Exercices et Travaux Pratiques sur Micro-ordinateur

Publication IREM de Strasbourg, aux éditions Ellipses

Par Jacques FARAUT et Elisabeth KHALILI

Un cours d'arithmétique a été introduit en première année du premier cycle de mathématiques de l'université Louis Pasteur en 1986. Cet enseignement comprend des travaux pratiques sur micro-ordinateur qui donnent aux étudiants l'occasion d'analyser des algorithmes, comme le crible d'Eratosthène, de faire des expérimentations, sur la répartition des nombres premiers par exemple, ou de se faire une idée des applications de l'arithmétique, comme la cryptographie.

Un chapitre du cours est consacré aux fractions continues. C'est un sujet qui habituellement ne fait pas partie d'un cours d'arithmétique élémentaire, mais il a l'avantage de permettre un point de rencontre avec le cours d'analyse. De plus ce sujet ancien connait de nouveaux développements.

Sont rassemblées les notes de cours, les énoncés d'exercices et de travaux pratiques préparés par les personnes qui ont assuré cet enseignement pendant ces trois dernières années. Elles sont publiées principalement pour les travaux pratiques qui seuls présentent un caractère original.

Le texte a été saisi en TEX par Madame Evelyne LE GUYADER, qui assure également la saisie des articles de votre revue préférée (*'L'Ouvert'* bien sûr).

COMMENT COMPILER OU INTERPRÉTER UNE EXPRESSION ARITHMÉTIQUE

Raymond SEROUL

1. Le problème

Que se passe-t-il dans un ordinateur ou une calculatrice entre le moment où je tape sur le clavier

$$\sin(3 * x + \exp(1 + \cos(x * x))) + 3 * x * x - 7 * x + 1$$

et le moment où s'affiche le résultat ?
En entrée, l'ordinateur (ou la calculatrice) reçoit une *chaîne de caractères*. En sortie, j'obtiens la *valeur* de l'expression arithmétique que j'ai tapé au clavier.

Dans cet article, nous allons étudier les différentes étapes qui séparent l'entrée de la sortie. Et, chemin faisant, nous apprendrons à écrire un *interpréteur*, un *compilateur* d'expressions arithmétiques et à *produire* diverses sortes de code intermédiaire. Cela vous permettra de saisir directement au clavier une expression arithmétique ou une fonction sans être obligé de recompiler votre programme à chaque fois.

Le langage utilisé sera le PASCAL.

2. Expressions arithmétiques

2.1 — Par "expression arithmétique" nous entendrons ici une chaîne de caractères obtenue en concaténant :

- les lettres 'a' jusqu'à 'z';
- les symboles '+' et '$*$';
- les parenthèses '(' et ')'.

Nous avons appris petit à petit à discerner les "bonnes" expressions arithmétiques. Nous savons par exemple que $a*x*x+b*x+c$ est une "bonne" expression et que $a*x+)b$ n'est pas une "bonne" expression. Et cela, sans avoir jamais rencontré de définition précise. Vous allez comprendre pourquoi....

2.2 — Les "bonnes" expression arithmétiques se subdivisent en trois catégories : les *expressions* proprement dites, les *termes* et les *facteurs* dont voici la définition.

2.2.1 Nous noterons $\mathcal{E}$ (et nous appellerons *expression arithmétique*) une chaîne de caractères qui est un *terme* ou une *somme de termes*, c'est-à-dire une chaîne de caractères qui s'obtient en concaténant un terme, un signe '+', un autre terme, etc.

2.2.2 Nous noterons $\mathcal{T}$ (et nous appelerons *terme*) une chaîne de caractères qui est un *facteur* ou un *produit de facteurs*, c'est-à-dire une chaîne qui s'obtient en concaténant un facteur, un signe '$*$', un autre facteur, etc.

2.2.3 Enfin, nous noterons $\mathcal{F}$ (et nous appellerons *facteur*) une *variable* (c'est-à-dire une des lettres 'a', 'b',...,'z') ou une expression arithmétique parenthésée (c'est-à-dire la chaîne de caractères que l'on obtient en concaténant une parenthèse ouvrante, une expression arithmétique et une parenthèse fermante).

De manière imagée, nous pouvons exprimer les définitions précédentes par les "équations" :

$$\begin{array}{lll} \mathcal{E} = T & \text{ou} & \mathcal{E} = T + \cdots + T, \\ \mathcal{T} = F & \text{ou} & \mathcal{T} = F * \cdots * F, \\ \mathcal{F} = a & \text{ou} & \mathcal{F} = (E). \end{array}$$

Dans toute la suite, "expression" sera synonyme de "expression arithmétique".

REMARQUE : la définition que nous venons de donner laisse un certain malaise quand on la lit pour la première fois. Mais c'est le prix à payer avec les définitions récursives... Il est possible de donner une définition plus agréable d'une expression arithmétique, mais pour cela, il faut apprendre la théorie des grammaires. (La définition qui vient d'être donnée est d'ailleurs directement inspirée de cette théorie.)

EXEMPLE : la chaîne de caractères

$$a + (b + c * x) * u * v + e + f$$

est-elle une bonne expression arithmétique au sens de notre définition? Procédons petit à petit :

a) Si les chaînes a, $(b + c * x) * u * v$, e et f sont des termes, alors la chaîne donnée est une bonne expresssion arithmétique en vertu de (2.2.1).

b) En combinant (2.2.2) et (2.2.3), nous voyons tout de suite que les chaînes a, e et f sont des termes, car ce sont des facteurs.

c) La définition (2.2.2) nous indique que $(b + c * x) * u * v$ est un terme si nous savons prouver que $(b + c * x)$, u et v sont trois facteurs.

d) (Accélérons un peu.) Si la chaîne $b + c * x$ est une bonne expression, nous saurons que $(b + c * x)$ est un facteur en vertu de (2.2.3). Pour savoir si $b + c * x$ est une expression, il faut recommencer : si b et $c * x$ sont deux termes, alors $b + c * x$ est une expression. Mais b est un facteur, donc un terme, etc.

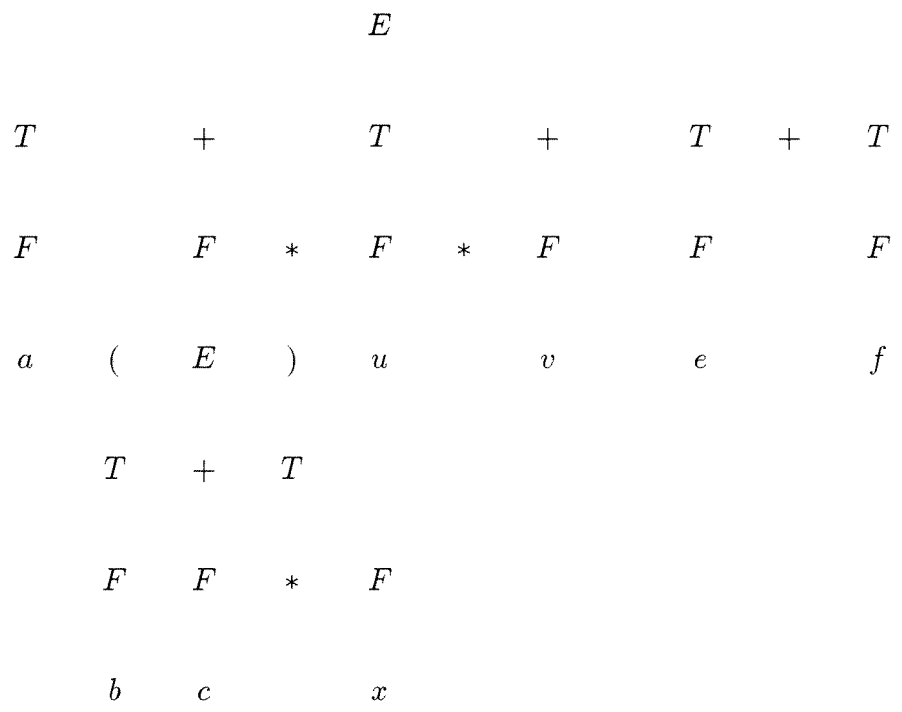


Figure 1

e) Enfin u et v sont deux facteurs.

Nous pouvons illustrer et résumer la démarche que nous venons de suivre sous la forme d'une arborescence (voir Figure 1).

REMARQUE : il importe de bien comprendre la portée des définitions (2.2.1), (2.2.2) et (2.2.3). Une chaîne de caractères est une bonne expression arithmétique s'il est possible de la décomposer en une somme de termes; une chaîne est un terme s'il est possible de la décomposer en un produit de facteurs, etc. Nous n'avons aucune protection contre les manières farfelues de fabriquer une expression, un terme ou un facteur. Par exemple, la concaténation des chaînes $a*($ et $b+c)$ est une expression. Mais nous savons que $a*(b+c)$ est une expression parce qu'il existe une décomposition "légale" en un produit de facteurs.

3. Reconnaître une expression arithmétique

3.1 — Nous allons écrire un programme PASCAL qui nous dira si une chaîne de caractères donnée est une bonne expression arithmétique. Nous aurons besoin des déclarations :

```
label
  999;
type
  str255 = string[255];
```

```
var
  expression : str255;
  token      : char;
  k          : integer;
```

La variable *expression* contient la chaîne à examiner et *token* contiendra un des caractères de cette chaîne. Pour éviter "d'arriver au bord" de *expression*, nous concatènerons la chaîne d'entrée avec le caractère '$'. Ce caractère sert d'indicateur de fin de chaîne; d'un point de vue technique, il nous assure de l'existence d'un caractère *après* l'expression qu'on cherche à reconnaître.

3.2 — La chaîne d'entrée n'est pas obligée d'être une bonne expression. Aussi avons-nous besoin d'une procédure *erreur*. La première erreur sera fatale, il n'y aura aucune tentative de traitement d'erreur. Pour simplifier, cette procédure n'envoie qu'un message lapidaire (sans diagnostic) :

```
procedure erreur ;
begin ;
    writeln('expression incorrecte') ;
    goto 999 ;
end ;
```

3.3 — La variable *token* sera successivement égale à tous les caractères de la chaîne *expression* (y compris le caractère '$' qui borde désormais cette chaîne). C'est la procédure *next_token* qui modifie *token*. Pour simplifier l'écriture des procédures que nous verrons plus loin, il est utile de demander à la procédure *next_token* de vérifier d'abord si la valeur courante de *token* est égale à un caractère qu'on lui transmet. Si c'est le cas, *token* change de valeur. Sinon, *next_token* tire la sonnette d'alarme :

```
procedure next_token(tok : char) ;
begin ;
    if tok <> token then erreur ;
    k := k + 1 ;
    token := expression[k] ;
end ;
```

Cas particulier : *next_token*(*token*) change la valeur de *token* sans jamais déclencher d'erreur.

3.4 — Avant de commencer à programmer, demandons-nous quels caractères peuvent commencer ou terminer une expression, un terme ou un facteur :

- une expression, un terme ou un facteur ne peuvent débuter que par une lettre (i.e. un nom de variable) ou une parenthèse ouvrante;
- une expression ne peut être suivie que par l'un des deux caractères ')' ou '$';
- un terme ne peut être suivi que par l'un des trois caractères '+', ')' ou '$';
- un facteur ne peut être suivi que par l'un des quatre caractères '+', '*', ')' ou '$'.

3.5 — Nous allons maintenant écrire trois procédures E, T et F qui ne font que traduire très fidèlement les définitions (2.2.1), (2.2.2) et (2.2.3). Le cahier des charges est le suivant :

- Les procédures E, T et F modifient la valeur de la variable *token*.
- La procédure E "lit" (c'est-à-dire passe successivement en revue) les caractères de la chaîne *expression* en partant du caractère *token* et avance dans la chaîne donnée jusqu'à la fin de la plus grande expression arithmétique qui commence avec *token*. Quand la procédure E s'arrête, la variable *token* doit être égale au premier caractère non examiné. De manière imagée, nous dirons que la procédure E *reconnaît* les expressions arithmétiques.
- De manière analogue, la procédure T reconnaît le plus grand terme de *expression* qui débute par *token*.
- Enfin, la procédure F reconnaît le plus grand facteur de *expression* qui débute par *token*.

3.6 — Comme chacune des procédures E, T ou F pourra appeler l'une des trois autres, nous devons utiliser la déclaration **forward** :

```
procedure E ; forward ;
procedure T ; forward ;
procedure F ; forward ;
```

3.7 — La procédure E commence par "consommer" un terme, puis consomme autant qu'elle le peut des chaînes de la forme $+\,T$. On examine alors la valeur de *token* : si cette variable est différente d'une parenthèse fermante (cas d'une sous-expression) ou d'un signe dollar (lorsque la chaîne à traiter a été entièrement lue), on déclenche une alarme :

```
procedure E ;
begin
   T ;
   while token = '+' do
   begin next_token(token) ; T end ;
   if not (token in [')', '$']) then erreur ;
end ;
```

3.8 — On procède de manière analogue pour la procédure T (l'alarme est donnée si le caractère qui suit le terme n'est pas le signe '+', une parenthèse fermante ou la fin de la chaîne) :

```
procedure T ;
begin
   F ;
   while token = '*' do
   begin next_token(token) ; F end ;
   if not (token in ['+', ')', '$']) then erreur ;
end ;
```

3.9 — La procédure F examine si le caractère courant de la chaîne d'entrée est un nom de variable. Si c'est le cas, elle le consomme (procédure *next_token*). Dans le cas contraire, elle vérifie qu'elle a affaire à une expression parenthésée. Enfin, si *token* n'est pas une lettre ou une parenthèse ouvrante, l'alarme est déclenchée :

```
procedure F ;
begin
   if token in ['a'..'z']
   then next_token(token)
   else begin next_token('(') ; E ; next_token(')') end ;
   if not (token in ['+', '*', ')', '$']) then erreur ;
end ;
```

3.10 — REMARQUES :

1) Une lecture superficielle suggère que seule la procédure F s'assure que le premier caractère de la sous-chaîne à traiter est correct. Mais le premier travail de la procédure E est d'appeler T, qui appelle F. Par conséquent, E et T s'assurent bien que la sous-chaîne à traiter commence par un caractère légal (lettre ou parenthèse ouvrante).

2) Chacune des procédures E, T et F teste si le caractère qui suit la sous-chaîne qu'elle traite est correct. On peut alléger ces tests en cascade. En effet, le test qui figure à la fin de la procédure T

if not (*token* **in** [')','$']) **then** *erreur*

est inutile, car il fait double emploi avec le test correspondant qu'effectue la procédure E appelant T. De même, le test

if not (*token* **in** ['+','*',')','$']) **then** *erreur*

qui figure à la fin de la procédure F est inutile (car T ou E s'en chargent). Ces tests seront donc supprimés dans les versions ultérieures des procédures T et F.

3.11 — Le corps du programme est minuscule : il consiste à entrer la chaîne *expression* qu'on borde ensuite avec le symbole '$', puis à initialiser les variables k et *token*. Ceci fait, la procédure E traite la chaîne. Par construction, E reconnaît la plus grande chaîne qui commence par *expression*[1]. Par exemple, si la chaîne d'entrée est $a + bc$, la procédure E reconnaît $a + b$ et *token* vaut c. Il est donc indispensable de s'assurer que la chaîne *expression* a été entièrement lue (d'où le test **if** *token* <>' $'**then** *erreur*) :

```
begin
   write('expression = ') ; read(expression) ;
   expression := concat(expression, '$') ;
   k := 1 ; token := expression[k] ;
   E ;
   if token <> '$' then erreur ;
   999 :
end.
```

3.12 — Nous comprenons maintenant beaucoup mieux la Figure 1 de la section précédente. L'arborescence associée à la chaîne $\omega = a + (b + c * x) * u * v + e + f$ n'est pas autre chose que *l'arbre des appels* du programme principal lorsqu'on lui fournit la chaîne ω. (Une branche telle que "F — a" ou "E — $+$" correspondant à l'appel de *next_token*.)

3.13 — EXERCICE : enrichir le programme précédent pour qu'il émette un diagnostic raisonnable en cas d'erreur.

4. Interpréter ou évaluer une expression arithmétique

Nous supposons maintenant que les variables a à z possèdent des valeurs mémorisées dans le tableau $variables['a'..'z']$.

4.1 — Pour *calculer* la valeur d'une expression, il suffit d'introduire une variable globale *valeur* que les procédure E, T, F mettent à jour. Plus précisément, après appel de $E(valeur)$, la variable *valeur* contiendra la valeur de l'expression reconnue par E. De même, après l'appel de $T(valeur)$ (resp. $F(valeur)$), la variable *valeur* contiendra la valeur du terme (resp. facteur) reconnu par T (resp. F). La déclaration de ces trois procédures est donc :

```
procedure E(var valeur : real) ; forward ;
procedure T(var valeur : real) ; forward ;
procedure F(var valeur : real) ; forward ;
```

4.2 — Si une expression se réduit à un terme, sa valeur est celle du terme. Si une expression est une somme de termes, sa valeur est la somme des valeurs des différents termes qu'on ajoute au fur et à mesure :

```
procedure E ;
var new_val : real ;
begin
   T(valeur) ;
   while token = '+' do begin
      next_token(token) ;
      T(new_val) ;
      valeur := valeur + new_val ;
      end ;
   if not (token in [')', '$']) then erreur ;
end ;
```

4.3 — La procédure T est calquée sur la procédure E, à ceci près qu'il s'agit de multiplications et pas d'additions :

```
procedure T ;
var new_val : real ;
begin
   F(valeur) ;
   while token = '*' do begin
      next_token(token) ;
      F(new_val) ;
      valeur := valeur * new_val ;
      end ;
end ;
```

4.4 — Si un facteur est un nom de variable, sa valeur est celle de la variable et si un facteur est une expression parenthésée, sa valeur est celle de l'expression elle-même :

```
procedure F ;
begin
   if token in ['a'..'z'] then
   begin valeur := variables[token] ; next_token(token) end
   else
   begin next_token('(') ; E(valeur) ; next_token(')') end ;
   end ;
```

4.5 — Le corps du programme principal est le même que dans la section précédente, à ceci près qu'il faut penser à donner aux variables 'a', 'b', etc. une valeur avant de lancer l'appel de $E(valeur)$.

5. Compiler une expression arithmétique

Dans la partie précédente, nous avons vu comment interpréter (c'est-à-dire évaluer) une expression arithmétique entrée au clavier. Nous connaissons tous l'inconvénient de cette démarche : si l'expression donnée doit être calculée 1000 fois de suite, le programme de la partie précédente *reconnaîtra et évaluera* la même chaîne 1000 fois de suite.
Dans le programme d'interprétation, le *calcul* de la valeur se fait en même temps que la *reconnaissance*. Ne serait-il pas possible de reconnaître la chaîne une fois pour toutes et de séparer ainsi la reconnaissance du calcul de la valeur? Dans l'exemple de la boucle, nous remplacerions 1000 reconnaissances et 1000 calculs de valeur par une seule reconnaissance et 1000 calculs de valeur.

5.1 — Mais quelle forme donner au résultat de la reconnaissance ? Nous choisirons ici la notation *polonaise suffixée*. Rappelons rapidement les propriétés de cette notation. L'expression arithmétique

$$\omega = a + \big((b + c * d) + (e + f)\big)$$

s'écrit en polonais suffixé

$$\pi = abcd * + ef + + + .$$

Si nous disposons de la chaîne π, *l'évaluation* de l'expression arithmétique de l'expression ω est très facile.

L'algorithme est le suivant. Il faut disposer d'une pile (qu'on appelle la *pile d'évaluation*). Au départ, la pile est vide. On lit successivement les caractères de la chaîne "polonaise" :

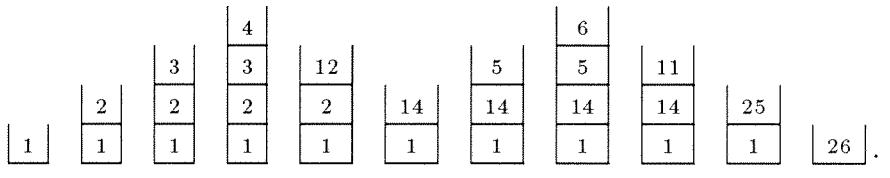
- si le caractère est une lettre (c'est-à-dire un nom de variable), on empile la *valeur* de cette variable dans la pile d'évaluation ;
- si le caractère est un signe '+', on dépile les deux dernières valeurs empilées dans la pile d'évaluation, on fait la somme des deux nombres ainsi obtenus, puis on redépose le résultat au sommet de la pile ;
- les opérations sont analogues si le caractère est le signe '$*$' (en remplaçant naturellement addition par multiplication).

Lorsque la chaîne a été entièrement lue, la pile d'évaluation ne contient plus qu'un seul nombre : c'est la valeur de l'expression.

EXEMPLE : évaluons l'expression $abcd * + ef + + +$ sachant que l'on a

$$a = 1, \quad b = 2, \quad c = 3, \quad d = 4, \quad e = 5 \quad \text{et} \quad f = 6.$$

Les états successifs de la pile d'évaluation sont



5.2 — Reprenons nos trois procédures E, T et F. Le résultat de leur action — c'est-à-dire leur "valeur" — ne sera pas un nombre, mais une *chaîne de caractères*. Plus exactement le résultat de l'appel $E(polonais)$ sera la traduction "en polonais" de l'expression que E a reconnu. De même, le résultat de l'appel $T(polonais)$ (resp. $F(polonais)$) sera la traduction "en polonais" du terme (resp. facteur) que T (resp. F) a reconnu. Au niveau des déclarations, cela donne :

procedure E(**var** *polonais* : *str*255) ; **forward** ;
procedure T(**var** *polonais* : *str*255) ; **forward** ;
procedure F(**var** *polonais* : *str*255) ; **forward** ;

5.3 — Lorsqu'une expression arithmétique est réduite à un terme ($\mathcal{E} = \mathcal{T}$), la procédure E n'a rien à faire; elle se contente de transmettre le résultat qu'a calculé la procédure T. Par contre, si $\mathcal{E} = \mathcal{T}_1 + \mathcal{T}_2$ et si π_1 et π_2 sont les traductions en polonais des termes $\mathcal{T}_1$ et $\mathcal{T}_2$, la procédure E renvoie la chaîne $\pi_1\pi_2+$.

```
procedure E ;
var new_pol : str255 ;
begin
   T(polonais) ;
   while token = '+' do begin
      next_token ;
      T(new_pol) ;
      polonais := concat(polonais, new_pol, '+') ;
      end ;
   if not (token in [')', '$']) then erreur ;
end ;
```

5.4 — La procédure T se traite de manière analogue, à ceci près qu'on y remplace les additions par des multiplications :

```
procedure T ;
var new_pol : str255 ;
begin
   F(polonais) ;
   while token = '*' do begin
      next_token ;
      F(new_pol) ;
      polonais := concat(polonais, new_pol, '*') ;
      end ;
end ;
```

5.5 — La traduction en polonais du facteur a est bien évidemment la chaîne 'a' et la traduction de $(\mathcal{E})$ est celle de $\mathcal{E}$. La procédure F est, une fois de plus, la plus facile à écrire :

```
procedure F ;
begin
   if token in ['a'..'z'] then
      begin polonais := token ; next_token(token) end
   else
      begin next_token('(') ; E(polonais) ; next_token(')') end ;
end ;
```

5.6 — Question : que sont les parenthèses devenues? Considérons par exemple l'expression $a + b + c$. La procédure E la traite comme une somme de trois termes consécutifs, ce qui donne successivement a, $ab+$ et $ab + c+$. Introduisons des parenthèses : $a + (b + c)$. La procédure E traite cette chaîne comme la somme des termes $\mathcal{T}_1 = a$ et $\mathcal{T}_2 = (b + c)$. La traduction en polonais de $\mathcal{T}_2$ est $bc+$. Par conséquent, la traduction de $a + (b + c)$ est $abc++$. Les parenthèses servent donc uniquement à rythmer les appels de procédures.

6. Production de code intermédiaire

Voici une variante de la partie précédente. Il s'agit de remplacer une expression arithmétique par du *code intermédiaire*. On entend par là une suite d'ordres très simples (*résultat = opérande* ou *résultat = opérande1 opérateur opérande2*) qui permet l'évaluation d'une expression arithmétique donnée. Par exemple, l'expression arithmétique

$$a + \big((b + c * d) + (e + f)\big) * \big(x + (y + z) * (u + v)\big)$$

sera remplacée par le code intermédiaire

```
100 : t1=c*d      |  105 : t6=u+v
101 : t2=b+t1     |  106 : t7=t5*t6
102 : t3=e+f      |  107 : t8=x+t7
103 : t4=t2+t3    |  108 : t9=t4*t8
104 : t5=y+z      |  109 : t10=a+t9.
```

Comme nous ne nous autorisons qu'une seule opération à la fois, il est nécessaire d'introduire des *variables temporaires* (qui sont ici les variables $t1,\ldots,t10$). Ces variables mémorisent les calculs intermédiaires. La valeur de l'expression se trouve dans la dernière mémoire temporaire créée (ici $t10$).

Rappelons que nous choisissons d'écrire le code intermédiaire sur l'écran. Toujours dans un but de simplification, nous confondrons un nom de variable avec son emplacement en mémoire.

6.1 — Il est nécessaire de prévoir une procédure *new_temp* qui engendrera les temporaires nécessaires. Puisque nous envoyons tous nos résultats sur écran, la variable globale *temp*, modifiée par la procédure *new_temp*, sera une variable de type chaîne. Nous utiliserons aussi la variable *nb_temp* qui est une variable globale de type entier. Comme son nom l'indique, elle dénombre les temporaires déjà créés :

```
procedure new_temp(var temp : str255) ;
begin
   nb_temp := nb_temp + 1 ;
   NumToString(nb_temp, temp) ;
   temp := concat('t', temp) ;
end ;
```

6.2 — La "valeur" d'une expression sera ici le *code intermédiaire* correspondant à l'évaluation de l'expression reconnue. Plus précisémment, l'appel de $E(place)$ engendrera le code intermédiaire nécessaire pour évaluer l'expression reconnue par E. En outre, la variable *place* contiendra l'emplacement mémoire où sera déposé la valeur de l'expression reconnue par E quand on exécute le code intermédiaire correspondant. Par exemple, lorsque la procédure E reconnaît la chaîne $a + b * c$, elle engendre le code

```
100 : t1=b*c ; 101 : t2=a+t1
```

et retourne l'emplacement `t1` dans *place*. Énoncés analogues pour T et F.
Vu notre choix (tout écrire sur l'écran), la variable *place* est une chaîne de caractères qui contient le nom d'un temporaire ou d'une variable. C'est une variable globale du programme. Au niveau des déclarations, cela nous donne :

procedure E(**var** *place* : $str255$) ; **forward** ;
procedure T(**var** *place* : $str255$) ; **forward** ;
procedure F(**var** *place* : $str255$) ; **forward** ;

6.3 — Lorsque nous avons une somme de termes $\mathcal{T}_1 + \mathcal{T}_2$ ou un produit de facteurs $\mathcal{F}_1 * \mathcal{F}_2$, soient $p1$ et $p2$ les emplacements où seront déposés les valeurs des termes $\mathcal{T}_1$, $\mathcal{T}_2$ ou celui des facteurs $\mathcal{F}_1$, $\mathcal{F}_2$.
Nous devons produire du code pour une addition ou une multiplication. Pour cela, on réserve d'abord un emplacement en mémoire (appel de *new_temp* qui mémorise l'emplacement dans *place*). Cet emplacement accueillera le résultat de l'addition ou de la multiplication. On produit ensuite le code correspondant :

procedure *gen_code*(**var** *place* : $str255$;
$p1 : str255$; $op : char$; $p2 : str255$) ;
begin
new_temp(*place*) ;
writeln(*next_quad* : 0, ' : ', *place*, ' = ', $p1$, op, $p2$) ;
next_quad := *next_quad* + 1 ;
end ;

Remarque : la variable *next_quad* est une variable globale du programme. Elle est de type entier et sert à numéroter les lignes écrites sur l'écran.

6.4 — Lorsqu'une expression arithmétique est réduite à un terme, la procédure E n'a rien à faire; elle se contente de transmettre l'emplacement calculé par la procédure T. Lorsque l'expression est une somme de termes, elle appelle la procédure *gen_code* pour calculer l'emplacement réservé à la somme (remarquer

les modifications successives de la variable *place*) :

```
procedure E ;
var new_place : str255 ;
begin
   T(place) ;
   while token = '+' do begin
      next_token ;
      T(new_place) ;
      gen_code(place, place, '+', new_place) ;
      end ;
   if not (token in [')', '$']) then erreur ;
end ;
```

6.5 — La procédure T se traite de manière analogue :

```
procedure T ;
var new_place : str255 ;
begin
   F(place) ;
   while token = '*' do begin
      next_token ;
      F(new_place) ;
      gen_code(place, place, '*', new_place) ;
      end ;
end ;
```

6.6 — L'emplacement en mémoire de la variable a est, vu nos conventions, le caractère 'a'. Quant à celui de $(\mathcal{E})$, c'est celui de $\mathcal{E}$:

```
procedure F ;
begin
   if token in ['a'..'z'] then
   begin place := token ; next_token(token) end
   else
   begin next_token('(') ; E(place) ; next_token(')') end
end ;
```

6.7 — Le corps du programme consiste à initialiser correctement les différentes variables globales, puis de demander à la procédure E de calculer l'emplacement

de la valeur de l'expression proposée :

```
begin
   write('expression arithmétique = ') ; readln(expression) ;
   expression := concat(expression, '$') ;
   k := 1 ; token := expression[k] ;
   nb_temp := 0 ; next_quad := 100 ;
   E(place) ;
   if token <> '$' then erreur ;
   999 :
end .
```

Comment utiliser moins de temporaires

6.8 — Nous venons d'apprendre à produire du code intermédiaire. Mais la solution proposée souffre d'un grave défaut : les temporaires ne sont pas recyclés. Or il est souvent possible d'évaluer une expression arithmétique compliquée avec peu de temporaires. Pour évaluer l'expression

$$a + \big((b + c * d) * (e * f + g)\big) * \big((x + (y + z)) * (u + v + w)\big),$$

trois temporaires suffisent :

```
100 : t1=c*d      |  106 : t2=x+t2
101 : t1=b+t1     |  107 : t3=u+v
102 : t2=e*f      |  108 : t3=t3+w
103 : t2=t2+g     |  109 : t2=t2*t3
104 : t1=t1*t2    |  110 : t1=t1*t2
105 : t2=y+z      |  111 : t1=a+t1.
```

Avec le programme précédent, il nous en faudrait 12!

6.9 — Voyons comment nous pouvons minimiser le nombre total des temporaires. Pour des raisons qui deviendront claires un peu plus loin, commençons par écrire une fonction booléenne capable de reconnaître si un emplacement mémoire est un temporaire ou celui d'une variable :

```
function is_temp(place : str255) : boolean ;
begin
   if length(place) > 1
   then is_temp := true
   else is_temp := false ;
end ;
```

6.10 — La procédure qui recycle les temporaires est des plus sommaires :

```
procedure release(temp : str255) ;
begin
   nb_temp := nb_temp − 1 ;
end ;
```

6.11 — Lorsque la procédure *gen_code* est activée, celle-ci examine les paramètres $p1$ et $p2$. Il y a quatre cas possibles :

- $p1$ et $p2$ sont deux temporaires. Comme $p1$ a été créé avant $p2$, on produit le code

$$*p1 = *p1 \ * op \ * p2 \ ;$$

et on recycle la variable $p2$ qui ne sert plus à rien (l'étoile indique qu'il faut prendre le *contenu* de la variable correspondante) ;

- $p1$ est un temporaire et $p2$ n'est pas un temporaire. On produit alors le code

$$*p1 = *p1 \ * op \ * p2 \ ;$$

- $p1$ n'est pas un temporaire et $p2$ est un temporaire. On produit alors le code

$$*p2 = *p1 \ * op \ *p2 \ ;$$

- $p1$ et $p2$ ne sont pas des temporaires (ce sont donc deux noms de variables). Dans ce cas, il est nécessaire de d'engendrer un temporaire qui recevra le résultat de l'opération.

$$new\_temp(place) \ ; \ *place = *p1 \ * op \ * p2 \ ;$$

Il est crucial ici d'utiliser un temporaire. En effet, si nous utilisions un nom de variable pour y déposer le résultat d'un calcul intermédiaire, la traduction de $x + y + x * z$ risquerait d'être quelque chose comme $x := x + y$; $x := x * z$; $x := x * x$!

On remarquera qu'en procédant ainsi on confère aux temporaires une structure de *pile* (le dernier créé est le premier détruit).

Si on prend la précaution de donner à la variable *place* la bonne valeur dans chaque cas de figure, nous pourrons nous borner à n'engendrer qu'un seul type de

message :

```
procedure gen_code( var place : str255 ; p1 : str255 ;
                        op : char ; p2 : str255) ;
begin
   if is_temp(p1) then begin
      place := p1 ;
      if is_temp(p2) then release(p2)
      end
   else if is_temp(p2)
         then place := p2
         else new_temp(place) ;
   writeln(next_quad : 0, ' : ', place, ' = ', p1, op, p2) ;
   next_quad := next_quad + 1 ;
end ;
```

6.12 — Il n'y a rien à changer au reste du programme.

7. Un préprocesseur : le TRRUmpilateur (*)

Supposons que nous voulions travailler en virgule fixe sur une machine. Si le langage de programmation implanté sur cette machine est assez riche, il existe probablement un type *virgule fixe* qui autorise des additions, des multiplications et des divisions. Cette solution — très souvent utilisée en graphisme — conduit à des calculs beaucoup plus rapides qu'en virgule flottante.

Nous allons ici explorer une autre voie. Au lieu d'utiliser un type virgule fixe et de laisser gérer les retenues par la machine, nous allons tout faire nous-mêmes. Les calculs seront aussi rapides qu'en virgule fixe (puisque ce seront uniquement des calculs portant sur des entiers). Mais nous aurons la maîtrise complète des opérations.

7.1 — Mettons en place notre arsenal théorique. Soit $N > 1$ un nombre entier. Nous lui associons deux ensembles :

- l'ensemble VF (pour *virgule fixe*) des nombres rationnels x de la forme

$$x = x_0 + \frac{x_1}{N},$$

où $x_0, x_1 \in \mathbb{Z}$ sont deux entiers avec $0 \leq x_1 < N$;

- l'ensemble RM (pour *représentation en machine*) l'ensemble des couples (x_0, x_1) appartenant à $\mathbb{Z} \times \mathbb{N}$ tels que $0 \leq x_1 < N$.

(*) Se reporter à L'OUVERT **48** pages 25–30.

Le nombre rationnel $x \in \mathrm{VF}$, s'il est de taille raisonnable, sera représenté en machine par le couple $(x_0, x_1) \in \mathrm{RM}$. Pécisons ce que nous entendons par "représenter" x par le couple (x_0, x_1). On dispose de deux applications naturelles

$$\mathrm{code} : \mathrm{VF} \to \mathrm{RM} \quad \text{et} \quad \mathrm{valeur} : \mathrm{RM} \to \mathrm{VF}$$

définies par les formules

$$\mathrm{code}(x) = (x_0, x_1) \quad \text{avec} \quad \begin{cases} x_1 = (N * x) \ \mathbf{mod}\ N, \\ x_0 = (N * x - x_1) \ \mathbf{div}\ N; \end{cases}$$
$$\mathrm{valeur}(x_0, x_1) = x_0 + \frac{x_1}{N}\cdot$$

L'application de *normalisation* $[\![\ ]\!] : \mathbb{Z} \times \mathbb{Z} \to \mathrm{RM}$ définie par

$$[\![(x_0, x_1)]\!] = \big(x_0 + (x_1 \ \mathbf{div}\ N), x_1 \ \mathbf{mod}\ N\big).$$

permet de préciser les liens qui relient les applications code et valeur :

$$\mathrm{valeur}(\mathrm{code}(x)) = x \quad \text{et} \quad \mathrm{code}(\mathrm{valeur}(x_0, x_1)) = [\![(x_0, x_1)]\!].$$

7.2 — Pour trouver le code (dans RM) associé à la somme de deux nombres de VF, nous devons écrire, avec $t = x_1 + y_1$,

$$(1) \qquad \begin{aligned} x + y &= x_0 + y_0 + \Big(\frac{x_1 + y_1}{N}\Big) \\ &= x_0 + y_0 + (t \ \mathbf{mod}\ N) + \frac{t \ \mathbf{div}\ N}{N}\cdot \end{aligned}$$

Comme on s'y attend, trouver le code du produit de deux éléments de VF est beaucoup plus compliqué car il y a deux retenues. En posant $u = x_1 * y_1$ et $t = x_1 * y_0 + x_0 * y_1 + (u \ \mathbf{div}\ N)$, on a

$$(2) \qquad \begin{aligned} x * y &= \Big(x_0 + \frac{x_1}{N}\Big) * \Big(y_0 + \frac{y_1}{N}\Big) \\ &= x_0 * y_0 + \frac{x_0 y_1 + x_1 y_0}{N} + \frac{x_1 * y_1}{N^2} \\ &= x_0 * y_0 + (t \ \mathbf{div}\ N) + \frac{t \ \mathbf{mod}\ N}{N} + \frac{u \ \mathbf{mod}\ N}{N^2}\cdot \end{aligned}$$

7.3 — Les formules (1) et (2) suggèrent de munir l'ensemble RM d'une addition $\oplus$ et d'une multiplication $\otimes$ telles que :

$$(\mathrm{TRRU}) \qquad \begin{cases} (x \oplus y)_0 = x_0 + y_0 + (x_1 + y_1) \ \mathbf{div}\ N, \\ (x \oplus y)_1 = (x_1 + y_1) \ \mathbf{mod}\ N; \\ (x \otimes y)_0 = x_0 * y_0 + (t \ \mathbf{div}\ N), \\ (x \otimes y)_1 = t \ \mathbf{mod}\ N, \end{cases}$$

où, pour simplifier la définition de la multiplication, on a posé

$$u = x_1 y_1 \quad \text{et} \quad t = x_0 y_1 + x_1 y_0 + (u \ \mathbf{div}\ N).$$

La loi $\oplus$ est "exacte" en ce sens que l'on a

$$x + y = \text{valeur}\big(\text{code}(x) \oplus \text{code}(y)\big)$$

pour tous $x, y \in \text{VF}$. Par contre, la loi $\otimes$ n'est qu'une approximation de la multiplication des nombres rationnels :

$$x * y = \text{valeur}\big(\text{code}(x) \otimes \text{code}(y)\big) + \frac{u \ \mathbf{mod}\ N}{N^2}.$$

L'erreur commise en remplaçant $x * y$ par $\text{valeur}(\text{code}(x) \otimes \text{code}(y))$ est donc majorée par N^{-1}.

Il résulte de cela que RM, muni de la loi $\oplus$, est un groupe abélien. Remarquons au passage que l'opposé de $(x_0, x_1) \in \text{RM}$ est

$$-x = (-x_0 - 1, N - x_1).$$

Par contre, la loi $\otimes$ n'est pas associative; RM n'est donc pas un anneau.

7.4 — La loi $\oplus$ gère automatiquement la retenue. Introduisons donc une variante de cette loi qui ne le fait pas :

$$x \oplus' y = (x_0 + y_0, x_1 + y_1).$$

Cette loi est définie sur $\mathbb{Z} \times \mathbb{N}$ et l'ensemble RM n'est pas stable pour cette addition. Les additions $\oplus$ et $\oplus'$ sont liées par la relation

$$x \oplus y = [\![x \oplus' y]\!].$$

7.5 — Donnons-nous maintenant une expression arithmétique

$$\omega = (a * x + b) * (c * x * x + d * x + e).$$

Si nous connaissons les codes (dans RM) des nombres a, b, etc, nous pouvons calculer la valeur de l'expression

$$\omega' = (a \otimes x \oplus b) \otimes (c \otimes x \otimes x \oplus d \otimes x \oplus e).$$

ou, si nous désirons gérer "à la main" les retenues, calculer la valeur de l'expression

$$\omega'' = [\![a \otimes x \oplus' b]\!] \otimes [\![c \otimes x \otimes x \oplus' d \otimes x \oplus' e]\!].$$

Et il y a encore beaucoup d'autres variantes!

Précisons ce que nous entendons par *calculer* la valeur de ω' ou de ω'' : on fait comme si $\oplus$ et $\otimes$ étaient une "vraie" addition et une "vraie" multiplication. Autrement dit, les règles usuelles d'évaluation de gauche à droite, de priorité, de parenthésage, etc, sont respectées.

7.6 — Considérons l'expression très simple

$$\omega = (a * x + b) * (c * y + d).$$

Pour calculer la valeur de $\omega' = (a \otimes x \oplus b) \otimes (c \otimes y \oplus d)$, nous devrons générer à partir de ω un source PASCAL qui peut ressembler à ceci :

$$\begin{aligned}
&temp := a0 * x1 + a1 * x0 + (a1 * x1) \textbf{ div } N;\\
&TA0 := a0 * x0 + (temp \textbf{ div } N);\\
&TA1 := temp \textbf{ mod } N;\\
&temp := TA1 + b1;\\
&TB0 := TA0 + b0 + (temp \textbf{ div } N);\\
&TB1 := temp \textbf{ mod } N;\\
&temp := c0 * y1 + c1 * y0 + (c1 * y1) \textbf{ div } N;\\
&TC0 := c0 * y0 + (temp \textbf{ div } N);\\
&TC1 := temp \textbf{ mod } N;\\
&temp := TC1 + d1;\\
&TD0 := TC0 + d0 + (temp \textbf{ div } N);\\
&TD1 := temp \textbf{ mod } N;\\
&temp := TB0 * TD1 + TB1 * TD0 + (TB1 * TD1) \textbf{ div } N;\\
&TE0 := TB0 * TD0 + (temp \textbf{ div } N);\\
&TE1 := temp \textbf{ mod } N;
\end{aligned}$$

Explications : la variable temporaire $TA = (TA_0, TA_1)$ recueille le résultat du produit $a * x$ puis TB recueille la valeur de $a * x + b$, etc. Le code de la valeur de ω (ou plutôt le code d'une valeur approchée de ω) se trouve dans la dernière variable temporaire TH.

Générer un tel code est ennuyeux et mécanique, ce qui est source d'erreurs. Pourquoi ne pas écrire un programme PASCAL qui lirait l'expression ω et qui écrirait le code à notre place?

Pour cela, commençons par modifier la procédure *new_temp*. En effet, les temporaires s'appellent maintenant TA, TB, etc. :

```
procedure new_temp(var temp : str255) ;
begin
    temp := chr(ord('A') + nb_temp) ;
    temp := concat('T', t) ;
    nb_temp := nb_temp + 1 ;
end ;
```

7.7 — La procédure *gen_add* se charge de d'écrire sur l'écran le code correspondant

à une addition $\oplus$ et de réserver un emplacement pour le résultat :

```
procedure gen_add(var place : str255 ; p, q : str255) ;
var s : str255 ;
begin
    s := concat(p, '1 + ', q, '1') ;
    new_temp(place) ;
    writeln('temp := ', s, ' ; ') ;
    writeln(place, '1 := temp mod N ; ') ;
    writeln(place, '0 := ', p, '0 +', q, '0 + (temp div N ) ; ') ;
end ;
```

7.8 — Nous allons demander à la procédure E de :

- d'écrire sur l'écran un code PASCAL correspondant à l'expression (ou la sous-expression) qu'elle a reconnu ;
- renvoyer le nom de la variable qui contient le résultat de l'évaluation du code PASCAL

```
procedure E ;
var new_place : str255 ;
begin
    T(place) ;
    while token = '+' do begin
        next_token(token) ;
        T(new_place) ;
        gen_add(place, place, new_place) ;
        end ;
    if not (token in [')', '$']) then erreur ;
end ;
```

7.9 — La procédure *gen_mult* se charge d'écrire sur l'écran le code correspondant à une multiplication $\otimes$ et de réserver un emplacement pour le résultat :

```
procedure gen_mult(var place : str255 ; p, q : str255) ;
var u : str255 ;
begin
    u := concat(p, '1 * ', p, '1') ;
    writeln('temp := ',  p, '0 * ', q, '1 + ',
                         p, '1 * ', q, '0 + (', u, 'div N ) ; ') ;
    new_temp(place) ;
    writeln(place, '1 := temp mod N ; ') ;
    writeln(place, '0 := ', p, '0 *', q, '0 + (temp div N ) ; ') ;
end ;
```

7.10 — La procédure T est analogue à la procédure E : on remplace simplement les additions par des multiplications.

```
procedure T ;
var new_place : str255 ;
begin
   F(place) ;
   while token = '*' do begin
      next_token(token) ;
      F(new_place) ;
      gen_mult(place, place, new_place) ;
      end ;
end ;
```

7.11 — Enfin, la procédure F se contente d'inspecter les lieux et de prendre note, comme d'habitude :

```
procedure F ;
begin
   if token in ['a'..'z'] then
   begin place := token ; next_token(place) end
   else if token = '(' then
   begin next_token('(') ; E(place) ; next_token(')') end ;
end ;
```

7.12 — Il n'y a rien à changer au corps du programme principal.

7.13 — EXERCICE : modifier le programme pour qu'il engendre le moins possible de variables temporaires.

(à suivre ...)

A VOS STYLOS

PROBLÈME 11

Enoncé

Trouver le plus petit entier positif k pour lequel il existe un polynôme à coefficients entiers, de degré k, de la forme

$$P(x) = x^k + a_1 x^{k-1} + \cdots + a_k$$

et tel que, pour tout entier $x \in \mathbb{Z}$, $P(x)$ soit divisible par un milliard.

Solution (de "*L'Ouvert*") :

Soient n et t des entiers strictement positifs. Nous allons montrer qu'il existe un polynôme, à coefficients entiers, de degré k, de la forme

$$P(x) = x^k + a_1 x^{k-1} + \cdots + a_k$$

vérifiant $P(\mathbb{Z}) \subset n\mathbb{Z}$ si et seulement si n divise $k!$

Pour $n = 10^9$, comme $k = 40$ est le plus petit nombre tel que un milliard divise $k!$ (car 40! contient neuf fois le facteur premier 5 alors que 39! ne le contient que huit fois), ceci entraînera que la réponse est 40.

a) Si n divise $k!$, il existe un tel polynôme. Il suffit de prendre $P(x) = x(x+1)\ldots(x+k-1)$. Pour x entier positif, $P(x)/k!$ est un coefficient binomial, donc $P(x)$ est divisible par n; comme la propriété "$P(x)$ est divisible par n" ne dépend que de la valeur de x modulo n, elle s'étend à tout $x \in \mathbb{Z}$.

b) Si P est un tel polynôme, par récurrence sur m il est clair que pour tout m positif, $\Delta^m P$ envoie $\mathbb{Z}$ dans $n\mathbb{Z}$ (où Δ est l'opérateur de différence finie $\Delta Q(x) = Q(x+1) - Q(x)$); c'est donc vrai en particulier pour $m = k$. Mais Δ^k appliqué au polynôme $x^k + a_1 x^{k-1} + \cdots + a_k$ donne le polynôme constant $k!$, donc n divise $k!$.

PROBLÈME 12

Enoncé

Soit Ω un ouvert non vide du plan. Deux points C (chat) et S (souris) sont mobiles dans Ω et choisissent chacun à chaque instant leur vitesse, le module de cette dernière étant toutefois limité à un intervalle $[O, V]$, où la vitesse maximale V est la même pour C et S. On demande, selon la forme de Ω, si C a une stratégie imparable pour finir par rattraper S, si au contraire S a un moyen certain de toujours échapper à C, ou si ni l'un ni l'autre de ces deux cas ne se présente.

Indication

Pourvu que les positions initiales C_0 et S_0 soient différentes, il existe une stratégie permettant à S d'échapper indéfiniment à C, quel que soit l'ouvert non vide Ω.

PROBLÈME 13

Enoncé

Un réel x est algébrique si et seulement s'il existe des polynômes à coefficients entiers $P(u,v)$ et $Q(u,v)$ et des entiers u_o, v_o tels que, en posant

$$u_{n+1} = P(u_n, v_n) \quad ; \quad v_{n+1} = Q(u_n, v_n)$$

on ait $v_n \neq 0$ pour tout n et $(u_n/v_n) \longrightarrow x$, quand n tend vers ∞.

PROBLÈME 14 (proposé par D. DUMONT)

Enoncé

Démontrer l'égalité suivante pour $|x| < 1$:

$$\frac{x}{1+x} + \frac{2x^2}{1+x^2} + \frac{3x^3}{1+x^3} + \frac{4x^4}{1+x^4} + \cdots = \frac{x}{1-x} + \frac{3x^3}{1-x^3} + \frac{5x^5}{1-x^5} + \frac{7x^7}{1-x^7} + \cdots$$

Question complémentaire : Comparer à l'aide d'un micro-ordinateur les vitesses de convergence des deux séries. Comment croit la somme $S(x)$ de ces séries quand x tend vers 1^- ? (problème dont le résultat n'est pas connu par l'auteur).

DEVINETTE :

Qui a dit : "*Les mathématiques sont comme le porc, tout en est bon*" ?
La première bonne réponse aura une surprise!
