REEB : MÉMOIRES INFORMATIQUES

J.P. REVEILLÈS

U.F.R de Mathématiques et d'Informatique, Strasbourg

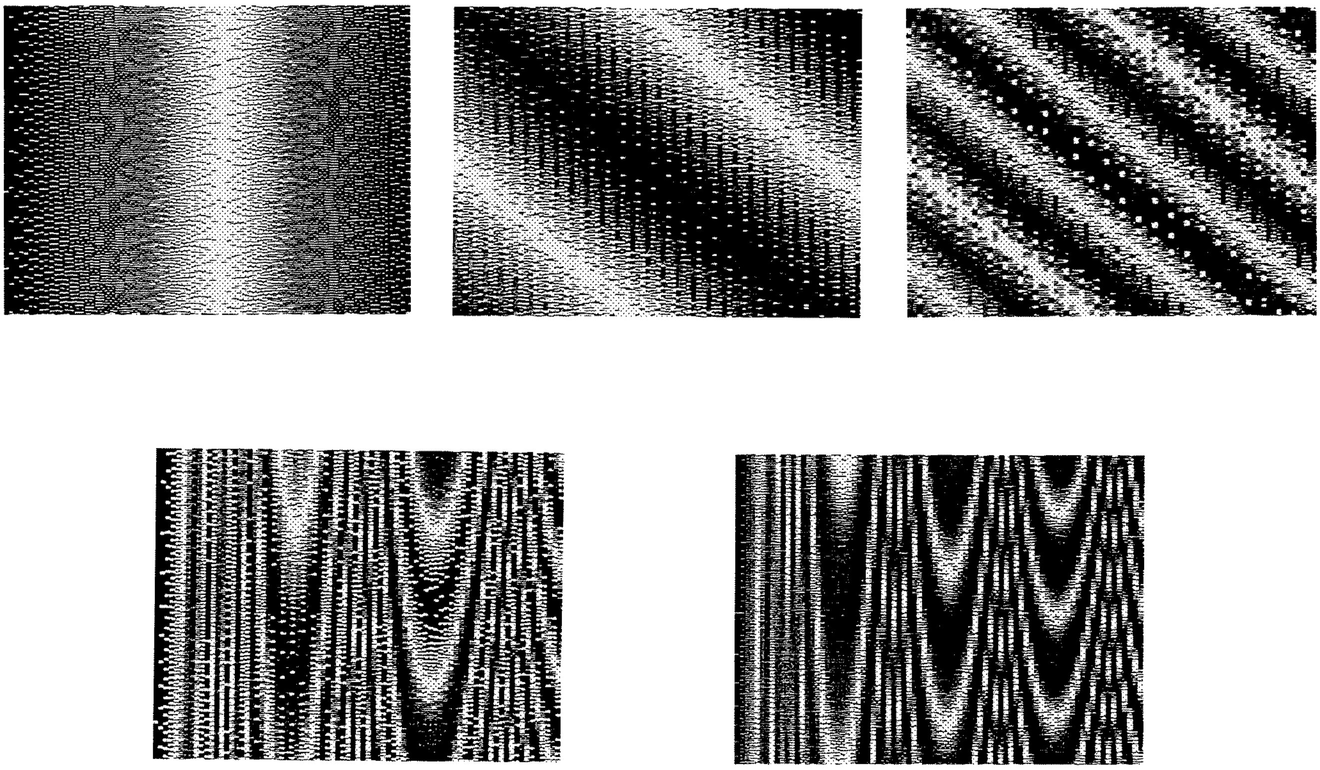
Les premières préoccupations non-standard dont G. Reeb m'ait parlé se classaient en deux rubriques : l'usage de réels non-standard, qui permettent de remplacer les limites par des valeurs infiniment voisines, et le théorème de Nelson d'existence d'un ensemble fini contenant tous les éléments standards. Alors que le premier thème continuait d'inspirer de nombreux travaux, il percevait dans le deuxième une potentialité sans commune mesure avec l'usage que l'on en faisait.

Nos discussions sur ce point avaient abouti au constat suivant : le théorème de Nelson décrit un processus de discrétisation parfait qui devrait intéresser l'informatique. Mais il y a loin de la coupe aux lèvres et il savait bien que de nombreux efforts devraient être consentis pour relier ce résultat très abstrait aux exigences des ordinateurs. Le bien fondé de cette idée fut assez vite confirmé par un résultat de Jacques Harthong permettant de considérer que « $\mathbb{R}$ n'est que $\mathbb{Z}$ vu de loin ». Ce fut la première étape d'une relation non-standard opératoire entre le discret et le continu. Elle donnait corps au théorème de Nelson et fut décisive pour l'intérêt que Reeb porta ultérieurement à l'informatique.

Dans l'équipe ANS, la conviction que le continu pouvait se traiter en ne recourant qu'aux nombres entiers s'était fermement établie. Il restait à mesurer jusqu'à quel point ce principe s'appliquait aux ordinateurs. Le problème n'est pas si simple qu'il y paraît, car la relation admise entre les nombres dits réels et les machines passe par leur représentation *flottante*, qui voile de nombreuses questions fondamentales.

Le premier test que Reeb entreprit de conduire fut de simuler le moiré. Beaucoup savent à quel point le travail de Harthong sur le moiré enthousiasma Georges et se le remémoreront, les lunettes à califourchon sur le crâne, en train d'examiner à contre-jour les franges produites par deux grilles superposées. Il n'est donc pas étonnant qu'il ait cherché à reproduire, sur un écran discret, ce qu'il appelait *les lois de Harthong* du moiré. Certains lecteurs seront peut-être surpris en apprenant que c'est sur un MO5 — micro-ordinateur scolaire — qu'il a conduit ses premières expériences.

Cette machine était dotée d'une gestion vidéo qui ne convenait pas du tout à ce qu'il souhaitait réaliser. Je me souviens fort bien de la quantité d'énergie et des



Quelques moirés de discrétisations de polynômes quadratiques et cubiques.

ruses qu'il a déployées pour réussir à contourner les limitations de cette machine rustique pour arriver à ses fins. Il avait réussi à écrire un programme qui tenait en 25 octets ! Comme aucun langage assembleur n'était fournit avec cette machine, il avait élaboré peu à peu, à la main, la liste des codes hexadécimaux des instructions. Il chargeait ces octets par des *pokes* à partir du Basic comme tous les *hackers* de cette époque.

Le résultat était extraordinaire. Il avait réussi, en triturant la RAM vidéo, à constituer deux très longues bandes striées (environ 2m selon lui), qui glissaient automatiquement l'une sur l'autre en recouvrant tout l'écran. Il n'y avait plus qu'à attendre l'apparition des lois de Harthong. Et, bien sûr, elles étaient au rendez-vous, bien que fût sous une forme très surprenante, qui le faisait s'exclamer : « C'est incroyable de voir comment l'arithmétique s'y prend pour singer le continu ! »

Cette discrétisation à la fois ingénieuse et brutale du phénomène de moiré avait encore le défaut — à ses yeux — de ne pas bien rendre compte de ces lois « aux petits angles ». Nous comprîment ce point beaucoup plus tard en utilisant des machines plus dociles et des discrétisations plus fines.

```
uses : turbo3, graph3, crt ;
const SegEcr = $B800 ; OfsEcr1 = 0 ;
      OfsEcr2 = 8192 ; OfsDernTr = 7999 ;
      Nbcr = 80 ; res = 100 ; noir = 255 ;
var TraitFinal : byte ;
    i, j : integer ;
    x : real ;
begin
  Hires ; {Haute résolution graphique}
  {Une fonction discrétisée dans la première bande}
  for i := 0 to OfsDernTr do begin
    x := i ;
    Mem[SegEcr : OfsEcr1 + (i div res) + Nbcr * (i mod res)]
                        := (trunc((x * x)/80000) mod 2) * noir
  end ;
  {Un deuxième exemplaire de la bande est entrelacé avec la première}
  for i := 0 to OfsDernTr
      do Mem[SegEcr : OfsEcr2 + i] := Mem[SegEcr : OfsEcr1 + i] ;
  repeat {Rotation de la première bande}
    TraitFinal := Mem[SegEcr : OfsDernTr] ;
    for j := Nbcr - 1 downto 0 do begin
      for i := res - 1 downto 1
          do Mem[SegEcr : j + Nbcr * i]
                        := Mem[SegEcr : j + Nbcr * (i - 1)] ;
      if j > 0
      then Mem[SegEcr : j] := Mem[SegEcr : +OfsDernTr - Nbcr + j]
      else Mem[SegEcr : j] := TraitFinal ;
    end ;
    delay(1000) ;
  until KeyPressed ;
end .
```

Simulation reebienne du moiré.

Les 25 octets reebiens sont certainement perdus, mais j'ai retrouvé sur l'une de mes disquettes de l'époque une version analogue à la sienne.

J'avais dû traduire en Pascal classique sa construction de deux bandes glissant l'une sur l'autre pour faire des comparaisons avec ses 25 octets. La superposition est en fait une juxtaposition, qui tire partie de la technologie des écrans entrelacés, où les octets pairs et impairs sont contenus dans des blocs mémoire séparés. Grâce à cette astuce, il suffit de faire tourner l'une des deux bandes *à côté* de l'autre pour

```
const unite = 100000 ;
var i, y, dy, reste : integer ;
begin {Calcul de 100.000 valeurs de l'exponentielle entre 0 et 1}
  y := unite ; dy := 0 ; reste := 0 ;
  for i := 0 to unite do begin
    y := y + dy ;
    dy := (y + reste) mod unite ;
    reste := (y + reste) div unite ;
  end ;
  write (y/unite : 10 : 5) ;
end .
```

L'algorithme de Reeb de tracé de la fonction exponentielle.

simuler le moiré. Il va sans dire que l'exécution est alors très rapide. L'instruction de temporisation delay(1000) a été rajoutée pour qu'une machine actuelle redonne la vitesse d'affichage originale; les incantations du type uses placées au début du programme permettent son exécution avec une version récente du compilateur.

Le phénomène de moiré se traduit par l'apparition de franges qui vont en se resserrant de plus en plus quand on accentue le décalage. Malheureusement, la juxtaposition atténue l'effet de moiré et avec la version jointe, il est préférable, de regarder le phénomène se dérouler sur l'écran, la reproduction sur papier de ces images n'étant pas bonne. (Celles qui sont reproduites ont été réalisées grâce à une variante plus élaborée qui utilise une vraie superposition et un écran à haute résolution.)

Peu à peu, la compréhension des modélisations finies du corps des réels au moyen de nombres entiers s'affermit. Reeb entreprit alors de traiter l'intégration numérique des équations différentielles par ce biais. Bien que la correspondance entre un écran d'ordinateur et un morceau de $\mathbb{Z}^2$ saute aux yeux, cette analogie n'avait jamais été exploitée comme moyen effectif de calcul depuis l'apparition des ordinateurs.

Un matin, nous le vîmes arriver en brandissant triomphalement une disquette. Je crois qu'il utilisait depuis peu le micro-ordinateur que l'équipe (étendue aux Diener) lui avait offert, ce qui simplifiait la communication : un chargement de fichier, un retour-chariot et nous avions sous les yeux un tracé ultra-rapide de courbe exponentielle; c'est le sentiment que j'ai gardé de cette expérience.

Devant notre surprise, il entama un discours d'épicier, parlant de valeurs trop grandes, d'économies, de gestion de restes. Il avait inséré un bout de multiprécision à l'intérieur du schéma d'intégration numérique d'Euler, alors que l'usage la place à l'extérieur des algorithmes pour des raisons d'indépendance.

Ce traitement inattendu du tracé de l'exponentielle eut une grande influence sur l'équipe ANS strasbourgeoise. Avec A. Troesch et E. Urlacher, nous étudiâmes et généralisâmes sans mal ce principe élémentaire, qui — nous le comprîmes peu à peu — rejoignait les méthodes dites de *virgule fixe*, en usage dans le calcul numérique avant l'apparition des ordinateurs et ajourd'hui dans les unités graphiques. Une version du programme de tracé de l'exponentielle est donnée ci-dessus.

Ces expérimentations montraient que les algorithmes de ce type, intéressants par leur rapidité et leur simplicité, n'apparaissaient nulle part dans la littérature numérique connue. Nous soupçonnions que ces méthodes étaient connues dans les laboratoires des constructeurs de composants ou des développeurs de logiciels très spécialisés (par exemple en graphique).

Une confirmation de l'originalité de l'idée de Georges vint pourtant assez vite. Comme clients IBM, nous eûmes le droit de tester une de leurs premières machines RISC qui avait la réputation de calculer très rapidement en nombres entiers.

L'expérience à conduire sautait aux yeux : faire exécuter les algorithmes en nombres entiers pour l'exponentielle, les lignes trigonométriques, etc. (que nous avions retravaillés et optimisés) pour les comparer aux calculs flottants réalisés par cette machine.

Il apparu immédiatement que cet ordinateur de pointe se trainait lamentablement dans ses calculs en nombres réels : les procédures étaient entre 100 et 200 fois plus lentes que les algorithmes en nombres entiers ! La bibliothèque numérique du compilateur ne connaissait visiblement pas ces méthodes.

Cette rencontre espérée, d'un domaine très particulier de l'informatique, à partir d'une quête purement mathématique renforçait dans l'esprit de Georges sa conviction de l'intérêt de l'emploi de l'analyse non standard dans le traitement du continu sur ordinateur. Il s'est très souvent exprimé sur ce point. Sous l'influence du moiré, ces préoccupations se sont peu à peu centrées autour de la discrétisation des lois de la géométrie euclidienne et nous avons pris conscience de l'immense quantité de travail cachée derrière cette question apparemment anodine.

La simulation du moiré «aux petits angles» (ou pour de petits décalages) est assez délicate et nous avait initialement troublés ; de même pour la construction sur écran de réseaux de finesse arbitraires. Longtemps après, nous avons compris que ces questions se résolvent assez bien à l'aide de droites discrètes lorsque les traits ne sont pas parallèles aux axes. En revanche, la simulation est difficile s'il y a parallélisme. Comme un moiré nécessite un exemplaire du réseau légèrement tourné, la discrétisation fait apparaître une juxtaposition de longs paliers qui détruisent le moiré attendu.

Ce problème n'est pas encore bien réglé aujourd'hui. C'était l'une des dernières préoccupations que Reeb mentionnait. Il cherchait à représenter des réseaux réguliers de *finesse non entière*, parallèles aux axes de coordonnées. Il est difficile, avec des pixels, de simuler un réseau de largeur 1,03 pixels par exemple. Là encore, Georges avait déployé sa ténacité coutumière pour aboutir à une notion de droite *bruitée* par une loi arithmétique qui mériterait certainement d'être formalisée et étudiée de près.

Quelques considérations sur les grands entiers, l'avaient mis sur une autre piste originale, qui consiste à *injecter* des parties de l'anneau des polynômes $\mathbb{Z}[x]$ dans $\mathbb{Z}$ pour tenter de remplacer le calcul formel par du calcul sur les entiers. Le lecteur pourra se convaincre facilement de cette idée en développant sur machine $(1+10^{25})^{37}$ pour constater que le résultat se découpe en tranches donnant les coefficients du binôme $\binom{37}{p}$. L'idée consiste donc à remplacer les polynômes par des entiers lacunaires dont les tranches de chiffres de mêmes longueurs contiennent les coefficients. Il avait entrepris, avec E. Urlacher, une exploration de cette piste.

Aujourd'hui, le moiré est toujours un sujet d'étude, tant théorique que pratique. Dans certains cas (télévision et traitement d'image par exemple), on veut le supprimer, ce qui nécessite une connaissance approfondie du phénomène. Les simulations de Reeb pourraient alors servir.

Reeb qualifiait le moiré de *machine à dériver*. Pour ma part, les discussions que j'ai eues avec lui sur ce sujet ont contribué à dégager quelques notions discrètes intéressantes : rotations, droites, transformations planes. On notera que ces recherches on finalement convergé avec les travaux de géométrie dite digitale entrepris indépendemment par A. Rosenfeld depuis vingt cinq ans.

Don Knuth m'avait dit, à propos du calcul en nombres entiers : « that's a very good idea ». Actuellement, ce calcul intervient directement dans divers domaines tels que l'algorithmique graphique, le traitement des erreurs, l'arithmétique des ordinateurs, la géométrie fractale, les codes de fonctions, etc. Ce calcul intègre les méthodes originales qui sont le témoignage de l'esprit pionnier de Georges Reeb.