
Coloriage du pavage dit « de TRUCHET »

Philippe ESPERET^a et Denis GIROU^b

^aLYCÉE HENRI IV
23 rue Clovis
75005 Paris

<esperet@marie.polytechnique.fr>

^bCNRS/IDRIS
BP 167
91403 Orsay cedex

<Denis.Girou@idris.fr>

Résumé. Un problème algorithmique de pavage du plan a fait l'objet d'un concours il y a trois ans. Après avoir présenté ce problème, nous donnons les principales solutions qui nous furent proposées. Le vainqueur désigné fut Rouben TER MINASSIAN.

Abstract. *Three years ago, an algorithmic problem on tiling of a plane was set as a contest puzzle. After presenting various aspects to the puzzle, we give the main answers received. The winner was Rouben Ter Minassian.*

1. Présentation

Un petit concours avait été proposé il y a déjà trois ans lors du tutoriel PSTricks du congrès GUTenberg 1995 et sur la liste de messagerie de la communauté francophone des utilisateurs de T_EX. Il s'agissait d'un problème algorithmique de pavage du plan (appelé abusivement « pavage de TRUCHET »), à résoudre via un langage de programmation graphique.

Dans sa version monochrome, ce pavage avait été trouvé au début de 1995 dans une revue consacrée au logiciel scientifique bien connu Mathematica [1], par Philippe ESPERET. Celui-ci s'était alors posé la question du coloriage des surfaces et des lignes, en vue d'en faire un problème d'informatique à poser à ses élèves de mathématiques spéciales. Il s'était en effet vite rendu compte que les solutions de ces questions n'étaient pas immédiates et demandaient le temps de la réflexion, constituant par là même d'excellents exercices !

Ce pavage était baptisé dans la revue « pavage de TRUCHET », du nom du Père Sébastien TRUCHET, un dominicain membre de l'Académie des sciences qui a vécu tout le siècle de LOUIS XIV et qui fut l'un des premiers à étudier la théorie des pavages, après les recherches pionnières du grand astronome KEPLER au début du XVII^e,

et ceci parmi une multitude d'intérêts et de travaux dans des domaines fort divers¹. Cette théorie n'allait connaître son véritable développement qu'à partir de la seconde moitié du XIX^e siècle, notamment en rapport avec le progrès des recherches en cristallographie. Mais TRUCHET fut le premier à tenter l'analyse exhaustive d'un pavage non trivial, à travers la mise en œuvre d'une méthode combinatoire.

Sébastien TRUCHET est cité, mais en une seule ligne, dans le livre de référence consacré à la théorie des pavages [9], impressionnante œuvre de synthèse de l'ensemble des connaissances théoriques accumulées dans ce domaine peu connu et difficile des mathématiques, mais c'est à Cyril SMITH [17] que revient le mérite d'avoir écrit la première (et seule...) étude récente consacrée aux recherches de TRUCHET sur les pavages, auxquelles on ne s'était plus intéressé depuis les travaux d'un contemporain de TRUCHET, le Père Dominique DOUAT, qui en avait publié le fruit en [5].

Laissons Cyril SMITH exposer l'origine et le contexte historique : « *En 1704, le prêtre dominicain Sébastien Truchet, qui s'intéressait aussi aux mathématiques et aux arts, a publié un texte où il montrait qu'une infinité de motifs peuvent être générés par l'assemblage de carreaux coloriés par moitié. Dans ce texte figurait en fait une toute première expression des principes de la théorie combinatoire et de la symétrie cristalline. De plus, ce texte de Truchet a été à l'origine d'un travail d'un autre dominicain du nom de Douat, qui a publié en 1722 un livre contenant une grande quantité de dessins juxtaposant des motifs de base. Ce livre a eu une influence sur tout l'art décoratif européen du XVIII^e siècle.* » (résumé d'après [17]).

En effet, TRUCHET considère une seule tuile de deux couleurs séparée par une ligne diagonale, admettant quatre positions possibles, et il assemble ces tuiles pour couvrir une surface tout en faisant varier l'orientation de celles-ci selon certaines règles précises (figures 1 et 2).

Quant au pavage abusivement baptisé « de TRUCHET » que nous avons donc utilisé dans notre problème, il est en fait une adaptation très simple de l'authentique « pavage de TRUCHET », où l'on remplace simplement, pour diviser chaque tuile, la diagonale de séparation par deux arcs de cercle, ce qui définit alors trois régions distinctes réduisant du même coup à deux orientations seulement les positions que peuvent prendre les tuiles, et où on détermine à chaque fois *au hasard* cette orientation, générant néanmoins dans tous les cas des segments qui joignent ceux des tuiles adjacentes, en créant ainsi des courbes ouvertes ou fermées sur une longueur souvent importante (figure 3). Car si nos recherches ultérieures ont permis de trouver quelques autres références où apparaissait ce très simple mais subtil pavage, nous avons néanmoins découvert que jamais Sébastien TRUCHET n'a utilisé, conçu et même pensé à ce pavage dont le nom lui a donc été donné abusivement... En fait, comme nous l'expliquons dans

1. Il fut notamment à son époque l'un des grands spécialistes des canaux fluviaux, mais aussi, comme on l'a redécouvert depuis peu d'années, le véritable inventeur du *point typographique* — voir [2] et <http://www.irisa.fr/faqtypo/truchet/truchet.html>

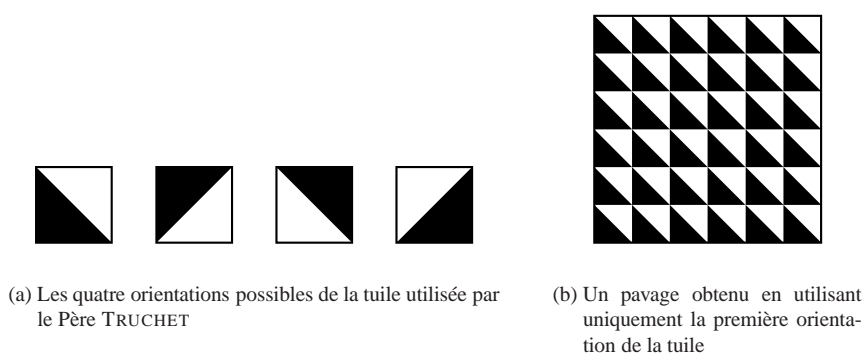


FIG. 1 – Tuile utilisée et exemple de pavage de Truchet

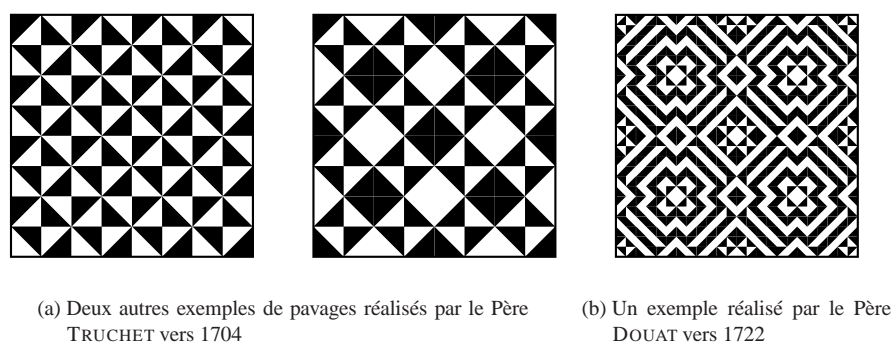
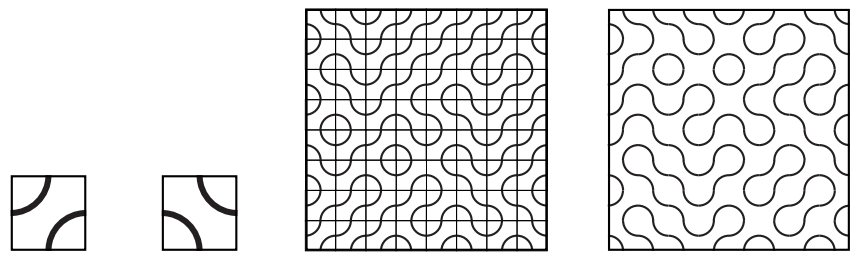


FIG. 2 – Autres exemples de pavages de Truchet

(Girou, à paraître), un nom plus judicieux serait « pavage de SMITH-LEWTHWAITE-PICKOVER-TRUCHET (SLPT) », parce que, après avoir exposé le travail de TRUCHET et reproduit un exemple de DOUAT, Cyril SMITH montre, parmi d'autres exemples de pavages « à la Truchet », celui que nous avons utilisé. De plus, Clifford PICKOVER, fécond chercheur dans le domaine de l'informatique graphique, a utilisé ce pavage proposé par SMITH et il est celui qui l'a popularisé à travers diverses publications [13, 14, 4]. Il a également à cette époque fait le lien avec un problème mathématique présenté sous forme de jeu de stratégie par G. LEWTHWAITE dans les années soixante [6], basé incidemment sur la manipulation des mêmes tuiles, et en rapprochant ce que montrait Cyril SMITH de ce problème, il a *pensé* ce dernier comme pavage. Mais s'il l'a initialement présenté comme *Truchet-like tiling*, la dénomination *Truchet tiling* a ensuite été constamment employée, d'où la confusion générale d'aujourd'hui... Depuis, on trouve en effet en divers endroits des références à un « pavage de Truchet » qui n'est

que le pavage SLPT² alors que le *vrai* « pavage de Truchet » est, lui, beaucoup plus méconnu...



(a) Les deux orientations possibles de la tuile du pavage « à la TRUCHET » SLPT

(b) Deux exemples de pavages monocolores SLPT

FIG. 3 – Tuile utilisée et exemples de pavages monocolores slpt

Le problème de notre concours était décomposé en trois parties :

1. tracé des courbes elles-mêmes, selon un procédé aléatoire (problème élémentaire, et cas particulier du problème suivant) ;
2. coloriage des lignes ;
3. coloriage des surfaces délimitées par les différentes courbes fermées et le bord du cadre³.

À titre illustratif, nous donnons ici l'algorithme simple correspondant au premier problème, qui s'implémente facilement dans n'importe quel langage graphique et qui est donc celui utilisé pour la figure 3 :

```

1 pour colonne=1,nb_colonnes
2   pour ligne=1,nb_lignes
3     si alea() = 0
4       poser(colonne,ligne){tuile}
5     sinon
6       poser(colonne,ligne){tourner{90}{tuile}}
7     fin si
8   fin pour
9 fin pour

```

2. On peut en découvrir un nombre assez surprenant, et dont certains font l'objet d'une grande diffusion : ainsi, sous le système Unix, des utilitaires de verrouillage de terminal (notamment `xlockmore` et `xscreensaver`) proposent un « pavage de Truchet » parmi leurs nombreuses animations d'écran ; d'autre part le logiciel de traitement d'images `Gimp`, aujourd'hui assez connu, offre parmi ses *motifs* un « pavage de Truchet », et même une variante de celui-ci avec un effet tridimensionnel...

3. Comme les tuiles de base sont monocolores, et non bicolores comme celles de TRUCHET, il n'était déjà pas immédiat de constater que seules deux couleurs suffisent pour colorier le pavage.

2. Réponses reçues

Nous avons reçu deux réponses complètes, toutes deux excellentes : l'une de Rouben TER MINASSIAN, professeur de mathématiques spéciales au lycée Janson de SAILLY à Paris, qui avait proposé une solution entièrement programmée en langage `PostScript`, et l'autre de Denis ROEGEL, maître de conférences à l'université de Nancy mais à l'époque doctorant en informatique au LORIA, qui avait quant à lui donné une solution dans le langage `METAPOST`. Nous présentons également ici notre propre solution en `PSTricks`. De plus, Jean-Marie TEULER, du CNRS/IDRIS, nous avait communiqué une solution partielle (écrite en langage `Fortran 90`), qui résout de façon non graphique le second problème, et nous avons également reçu, quelque temps après la clôture du concours, des solutions aux différents problèmes de la part de Jean-Bernard ROUX, professeur de mathématiques au collège de SAUSSURE en Suisse, solutions conçues avec le logiciel `MatLab` et basées en partie sur nos algorithmes alors diffusés [16]. D'autre part, Kees van der LAAN a publié en 1997 dans la revue `MAPS` du groupe néerlandais des utilisateurs de `TEX`, sa propre solution au troisième problème, programmée en `PostScript` [11].

Même si ce concours était essentiellement organisé pour le plaisir, et pour insister sur l'intérêt des langages graphiques *algorithmiques*, il était prévu de décerner un petit prix à la solution qui apparaîtrait la plus satisfaisante. En raison de son utilisation du langage `PostScript` comme spécifié dans notre règlement, qui n'autorisait que les solutions basées sur `PSTricks` et sur `PostScript` pour être certains que nous aurions les connaissances requises pour évaluer les solutions, ainsi que de l'originalité de son algorithme, ce prix avait été décerné à Rouben TER MINASSIAN, qui avait donc gagné un exemplaire du livre de Heinz-Otto PEITGEN, Hartmut JÜRGENS et Dietmar SAUPE *Chaos and Fractals — New Frontiers of Science*, Springer Verlag, 1992, livre qui est unanimement considéré comme l'actuel ouvrage de référence sur les questions qu'il traite.

3. Commentaires sur les réponses reçues

Ces problèmes comportaient en fait deux aspects très distincts : un aspect *algorithmique* et un aspect *langage de programmation graphique*.

Il est tout d'abord à noter que toutes les solutions fournies mettent en œuvre des algorithmes différents. Preuve en est de la richesse de ce problème !

Rouben TER MINASSIAN a utilisé un algorithme général pour résoudre simultanément les deux problèmes (le coloriage non bicolore des surfaces permettant de colorier les courbes). De plus, son implémentation offre des caractéristiques qui vont au-delà de celles que nous avons demandées : coloriage possible des surfaces en trois cou-

leurs non contiguës, repérage et coloriage spécifique des îlots de taille 1, utilisation de segments ou de flocons de Von KOCH à la place de cercles.

Pour le coloriage des lignes, Denis ROEGEL et Jean-Marie TEULER ont utilisé pour algorithme le principe de base suivant : on parcourt la grille abstraite représentant les arcs du pavage, et si un arc donné n'a pas déjà été colorié, on lui attribue une nouvelle couleur. Ensuite on regarde ses *suivants* potentiels pour leur attribuer la même couleur, et ceci jusqu'à ce qu'on arrive à un bord, ou qu'on obtienne un cycle (c'est-à-dire une courbe fermée). Denis ROEGEL a programmé cette solution de façon itérative et Jean-Marie TEULER de façon récursive.

Notre propre solution PSTricks est itérative. Dans le parcours de la grille abstraite des arcs du pavage, elle consiste à envisager dans chaque cas (fonction de la parité des numéros de ligne et de colonne) le seul voisin possible déjà colorié. Un seul cas particulier est à traiter à part, lorsqu'il y a deux voisins déjà coloriés, puisqu'il se peut que ces deux couleurs soient différentes.

Pour le coloriage des surfaces, Denis ROEGEL considère les voisins déjà coloriés au cours du parcours de la grille et calcule les couleurs de la tuile considérée à partir de ceux-ci.

Néanmoins, en dehors du fait que deux couleurs sont suffisantes, personne n'avait utilisé la propriété étonnante de ce pavage qui fait qu'en réalité il n'y a aucune dépendance entre les tuiles élémentaires pour ce qui est de la couleur de leurs surfaces ! Cela semble de prime abord surprenant, mais on peut pourtant s'en convaincre en examinant tous les (huit) cas des deux pavages 1x2 et 2x1. Les couleurs d'une tuile donnée ne dépendent que de la parité de ses numéros de ligne et de colonne. Aussi l'algorithme se réduit à :

```

1 pour colonne=1,nb_colonnes
2   pour ligne=1,nb_lignes
3     si impair(colonne+ligne)
4       si alea() = 0
5         poser(colonne,ligne){tuile(couleur1,couleur2)}
6       sinon
7         poser(colonne,ligne){tourner{90}
8                               {tuile(couleur2,couleur1)}}
9     fin si
10    sinon
11      si alea() = 0
12        poser(colonne,ligne){tuile(couleur2,couleur1)}
13      sinon
14        poser(colonne,ligne){tourner{90}
15                          {tuile(couleur1,couleur2)}}
16    fin si
17  fin si
18 fin pour
19 fin pour

```

4. Présentation des différentes réponses

4.1. Rouben Ter Minassian : solution basée sur PostScript

Il utilise pour son algorithme une idée originale : le coloriage non bicolore des surfaces induit le coloriage des courbes ; en effet, si g est la couleur de la surface « à gauche » et d celle « à droite », il suffit d'adopter sur la courbe frontière $f(g, d)$ où f est une application symétrique⁴ de $S \times S$ dans C , où S désigne l'ensemble des couleurs pour les surfaces et C celui des courbes. La symétrie est ici imposée parce qu'à chaque changement de concavité les notions de « droite » et de « gauche » sont modifiées.

Par exemple, si les couleurs des surfaces sont $(0, 1, 2, 3)$ (avec pour fixer les idées $0 \mapsto$ bleu, $1 \mapsto$ blanc, etc.) et celles des courbes $(0, 1, 2)$ (par exemple $0 \mapsto$ marron, $1 \mapsto$ vert, etc.), Rouben TER MINASSIAN adopte $f(a, b) = a + b \pmod 3$: si à gauche il y a du bleu et à droite du blanc, la courbe de séparation sera verte.

Cette approche lui permet de plus des sophistications comme le bi-coloriage de ces courbes : à la frontière entre une zone bleue et une zone rouge, on peut ainsi intercaler un double filet, rouge du côté bleu et bleu du côté rouge.

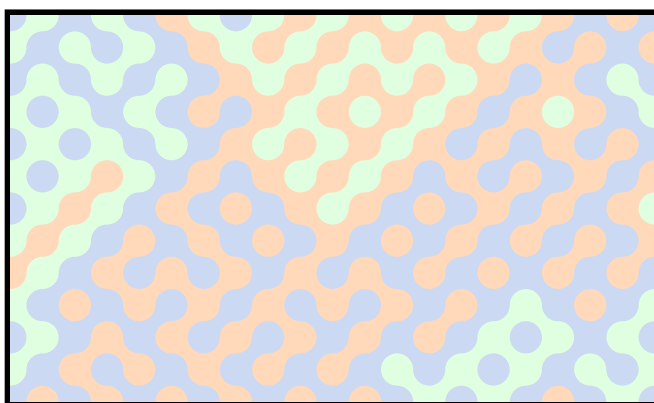


FIG. 4 – Rouben Ter Minassian : coloriage des seules surfaces

4. Telle que pour tout (x, y) , $f(x, y) = f(y, x)$.

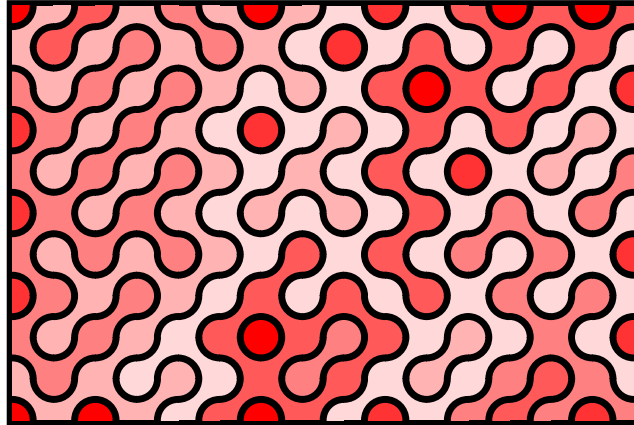


FIG. 5 – Rouben Ter Minassian : coloriage des surfaces, avec des lignes monocolors

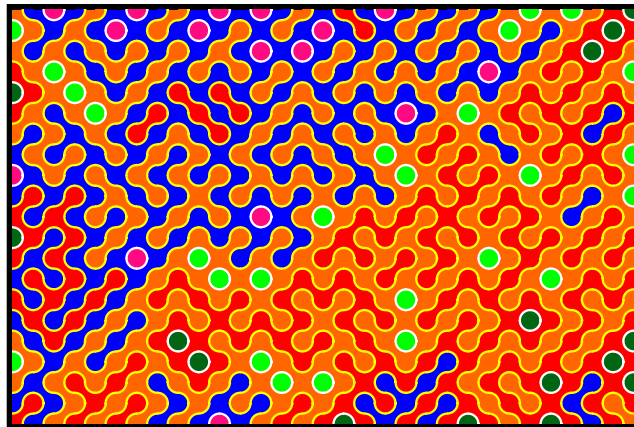


FIG. 6 – Rouben Ter Minassian : coloriage des surfaces et des lignes

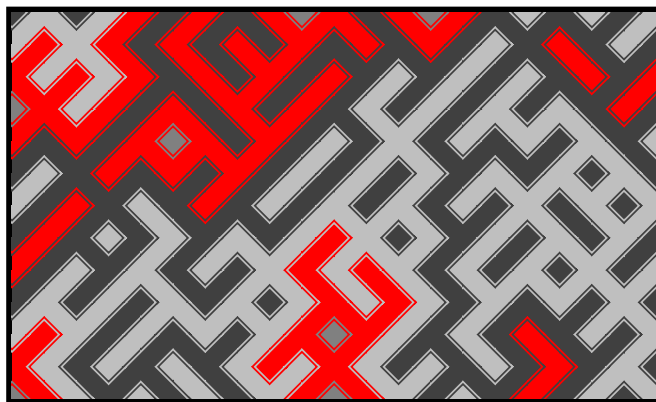


FIG. 7 – Rouben Ter Minassian : segments à la place de cercles

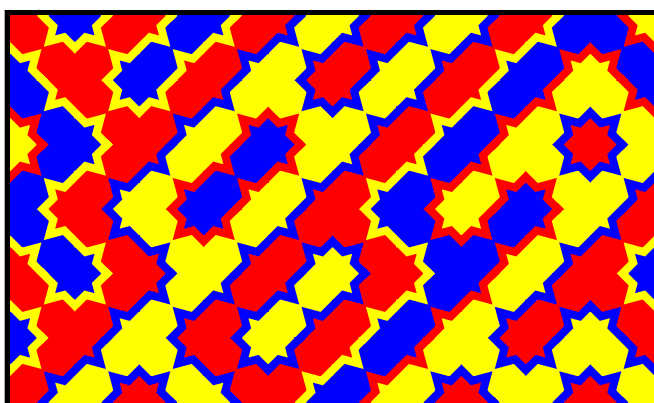


FIG. 8 – Rouben Ter Minassian : flocons de Von Koch à la place de cercles et bi-coloriage des lignes

4.2. Denis Roegel : solution basée sur METAPOST

Voici le commentaire général fourni à l'époque par Denis ROEGEL (le programme commenté, des exemples d'utilisation et diverses informations additionnelles se trouvent en [15]).

Le problème de TRUCHET devait être résolu soit par l'emploi de macros PSTricks, soit par une solution PostScript directe. Je n'ai suivi aucune de ces voies, non pas parce qu'elles m'apparaissaient difficiles, mais parce que je ne pouvais pas m'investir dans ces langages. J'ai préféré profiter de cet exercice pour perfectionner ma connaissance de METAPOST. Je crois qu'il vaut mieux maîtriser un langage unique, plutôt que d'en connaître plusieurs superficiellement.

METAPOST⁵ est un langage graphique dérivé de METAFONT et partage donc un certain nombre de caractéristiques avec PSTricks/TeX. C'est un *macrolangage* — tout comme TeX — et un programme METAPOST génère des fichiers PostScript pouvant être réutilisés par TeX ou éventuellement un autre logiciel. PSTricks génère aussi du PostScript mais les commandes y sont directement écrites dans le fichier .dvi. Parmi les acquis de METAFONT, on trouve en particulier la structure déclarative : il est possible de spécifier les coordonnées des points clés par des ensembles d'équations linéaires. Avec un peu d'habitude, cette structure déclarative devient très naturelle et accroît notablement la souplesse des figures produites.

5. METAPOST a été développé par John D. HOBBY, ATT Bell Laboratories, USA. Le manuel et d'autres documents sont disponibles sur CTAN ainsi que sur <http://www.loria.fr/tex>.

Ma solution au problème de TRUCHET n'est pas un exercice METAPOST typique, en ce sens qu'elle n'utilise pas ces potentialités déclaratives. J'ai plutôt tiré profit des caractéristiques suivantes :

- la possibilité de déclarer des tableaux de dimension quelconque, dont le nombre d'éléments n'est pas connu à l'avance ; par exemple la déclaration `numeric tilecolor[] [] []` crée un tableau de nombres (entiers ou réels) qui pourront être accédés par trois indices ; ceci n'est qu'un exemple, car en général, le nom d'une variable de tableau est composé de *tags* et de *suffixes*, un peu comme les paramètres d'une macro T_EX peuvent être séparés par des délimiteurs... ;
- la présence des structures de contrôle usuelles : `for`, `forever`, `exitif`, `endfor` ;
- la génération de nombres aléatoires (distributions uniforme et normale) ;
- le tracé de courbes de Bézier passant par certains points et soumis à des contraintes ; les courbes sont des objets de type `path`, elles peuvent subir diverses transformations (translations, rotations, déformations) et être coloriées, par défaut dans le système RGB ;
- le remplissage de zones délimitées par une courbe de type `path` ;
- le découpage en *macros* pour l'organisation des fonctionnalités et en particulier la possibilité d'évaluer ou non les paramètres d'une commande (type `expr` ou `text`).

Dans la solution que je propose, un pavage consiste en un tableau de pavés et pour chacun d'eux, la donnée de son orientation et le coloriage de ses lignes et parties. Les algorithmes employés sont les suivants :

lignes leur coloriage repose sur une opération plus simple, le *suivi* d'une ligne : une ligne encore non coloriée est choisie et parcourue d'un bout à l'autre ; on change de couleur à chaque nouvelle ligne.

surfaces leur coloriage repose sur l'observation que deux couleurs suffisent à colorier tout pavage du type considéré ; l'algorithme consiste à colorier un premier pavé et à diffuser ses couleurs sur ses voisins, jusqu'à avoir rempli tout le pavage ; le remplissage est donc réalisé pavé par pavé au sein même de METAPOST ; il eut été possible de définir une macro METAPOST prenant en argument un point d'un pavage, construisant la courbe bordant la région sélectionnée et remplissant alors cette région avec les commandes de METAPOST ; une telle macro peut s'avérer intéressante dans un cas plus général, mais pas pour un problème aussi simple et abordé avec une structure de tableau comme j'ai choisi de le faire.

METAPOST offre de très nombreuses autres possibilités, en particulier l'intégration de portions de texte dans les figures grâce à l'exécution de sous-processus T_EX.

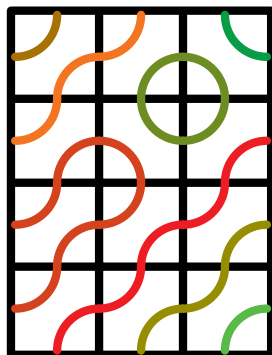


FIG. 9 – Denis Roegel :
coloriage des lignes

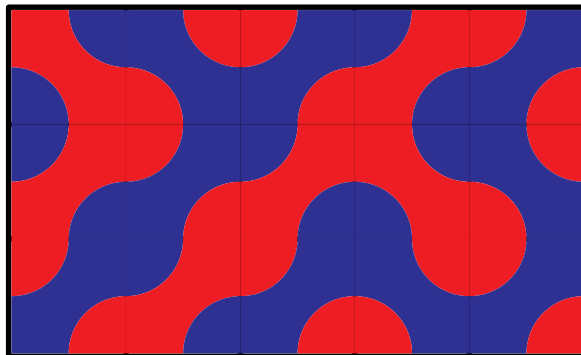


FIG. 10 – Denis Roegel : coloriage des surfaces

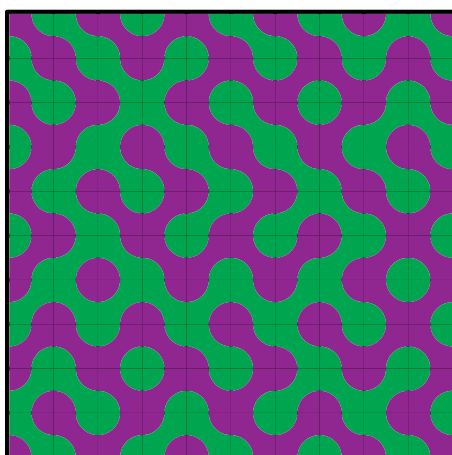


FIG. 11 – Denis Roegel : coloriage des surfaces

Les sorties générées sont des fichiers PostScript, pouvant être inclus directement par `\includegraphics` dans un fichier \LaTeX , sans que des calculs complexes doivent encore être faits par \TeX .

4.3. Denis Girou : solution basée sur PSTricks

Notre solution [7] demande l'utilisation de deux extensions particulières : l'une pour faire des tests aléatoires en *pile ou face* (nous avons alors repris la macro-commande proposée par Hans van der MEER⁶ [12]) et 'arrayjob', un ensemble de macros-commandes méconnu mais très puissant et commode, qui permet de manipuler avec \LaTeX des tableaux, au sens des langages Fortran ou C (nous avons extrait ces macros-commandes de l'extension 'formlett' proposée par Zhuhan JIANG [10]).

6. Aujourd'hui nous lui préférons la beaucoup plus puissante extension 'random' de Donald ARSENEAU [3].

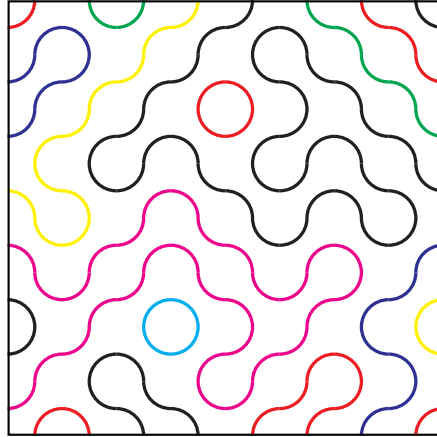


FIG. 12 – Denis Girou: coloriage des lignes

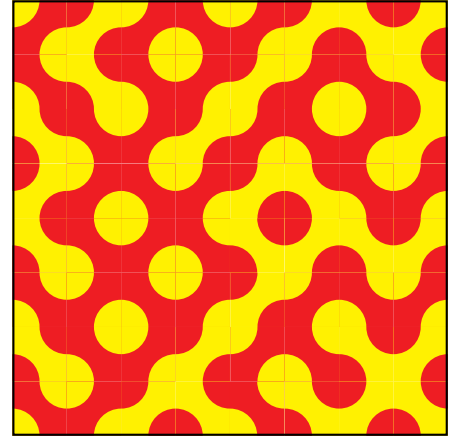


FIG. 13 – Denis Girou: coloriage des surfaces

Nous avons déjà expliqué plus haut l’algorithme très simple employé pour colorier les surfaces. Pour colorier les lignes, la solution que nous avons adoptée est itérative, et dans le parcours de la grille abstraite des arcs du pavage, consiste à considérer dans chaque cas le voisin parmi les arcs déjà coloriés (en fonction de la parité des numéros de ligne et de colonne, les positions des voisins possibles sont connues à chaque fois). Dans chaque cas, ou bien un tel voisin n’existe pas (on attribue alors une nouvelle couleur), ou bien l’arc en cours d’analyse doit avoir la même couleur. Un seul cas peut survenir qui pose problème (cas qui ne peut se produire que si les numéros de ligne et de colonne sont tous les deux impairs) : lorsqu’un arc peut avoir deux voisins déjà analysés (en *Ouest* et *Sud*, ou en *Nord-Ouest* et *Sud*). Dans ce cas, ces deux couleurs peuvent être différentes et donc on doit joindre ces deux segments. Pour ce faire, plutôt que de ré-analyser tous les arcs déjà coloriés pour faire les changements adéquats, on a résolu la difficulté en passant par un adressage indirect du tableau des indices de couleur, ce qui fait qu’on a dans ce cas des changements à opérer seulement sur le tableau de ces adresses, et non pas sur le tableau de la grille des arcs lui-même.

Néanmoins cette solution était lente en temps de calcul et limitée aussi à des tailles assez petites de pavages, à cause de la mémoire disponible dans les installations courantes de TEX . Mais depuis lors nous avons conçu une autre solution à partir des mêmes algorithmes, basée sur une extension de $\text{P}\text{S}\text{T}\text{ricks}$ diffusée assez récemment, ‘ pst-fill ’, qui permet de représenter avec facilité certaines catégories de pavages et rend possible dans ce cas précis, par une programmation mélangeant des commandes de $\text{P}\text{S}\text{T}\text{ricks}$ et des commandes natives de $\text{P}\text{o}\text{s}\text{t}\text{S}\text{c}\text{r}\text{i}\text{p}\text{t}$, de fournir une solution au pavage SLPT à la fois très rapide et très peu gourmande en mémoire, puisque l’essentiel du *travail* est fait non par TEX mais par $\text{P}\text{o}\text{s}\text{t}\text{S}\text{c}\text{r}\text{i}\text{p}\text{t}$ [8].

Bibliographie

- [1] Paul ABBOTT, « Tricks of the trade », *The Mathematica Journal*, Vol. 4, N° 2, page 25, spring 1994.
- [2] Jacques ANDRÉ and Denis GIROU, « Father Truchet, the typographic point, the *Romain du roi*, and tilings », *TUGboat*, Vol. 20, N° 1, March 1999 (à paraître).
- [3] Donald ARSENEAU, ‘*random*’ package: Generating “random” numbers in $\text{T}_{\text{E}}\text{X}$, CTAN/macros/generic/random.tex, 1995.
- [4] Alexander DEWDNEY, « Computer Recreations. Catch of the day: biomorphs on Truchet tiles, served with popcorn and snails », *Scientific American*, pages 92-95, July 1989.
- [5] Dominique DOUAT, *Méthode pour faire une infinité de desseins différents avec des carreaux mipartis de deux couleurs par une ligne diagonale: ou observations du Père Dominique Douat Religieux Carmes de la Province de Toulouse sur un mémoire inséré dans l’Histoire de l’Académie Royale des Sciences de Paris l’année 1704, présenté par le Révérend Père Sébastien Truchet religieux du même ordre, Académicien honoraire*, imprimé chez Jacques QUILLAU, Imprimeur Juré de l’Université, 1722.
- [6] Martin GARDNER, « Mathematical Games. Games of strategy for two players: star min, meander, dodgem and rex », *Scientific American*, pages 106-111, June 1975.
- [7] Denis GIROU, *Coloration of Truchet’s tiling*, <http://tug.org/applications/PSTricks/Tilings>, 1995-1998.
- [8] Denis GIROU, « pst-fill — a PSTricks package for filling and tiling areas », *TUGboat*, Vol. 19, N° 2, June 1998, pages 101-113.
- [9] Branko GRÜNBAUM et Geoffrey SHEPHARD, *Tilings and Patterns*, Freeman and Company, 1987.
- [10] Zhuhan JIANG, ‘*formlett*’ package: Make form letter for multiple receivers via plain $\text{T}_{\text{E}}\text{X}$, \LaTeX 2.09 or \LaTeX 2e, CTAN/macros/generic/formlett.sty, 1993-1995.
- [11] Kees van der LAAN, « Tiling in PostScript and METAFONT – Escher’s wink », *MAPS*, Vol. 19, N° 2, pages 39-67, 1997.
- [12] Hans van der MEER, « Random Bit Generator in $\text{T}_{\text{E}}\text{X}$ », *TUGboat*, Vol. 15, N° 1, March 1994.

-
- [13] Clifford PICKOVER, *Computers, Pattern, Chaos and Beauty*, St Martin's Press, 1990.
- [14] Clifford PICKOVER, « Picturing randomness with Truchet Tiles », *Journal of Recreational Mathematics*, Vol. 21, N° 4, pages 256-259, 1989.
- [15] Denis ROEGEL <Denis.Roegel@loria.fr>, *Truchet tiling*, CTAN/graphics/metapost/contrib/macros/truchet, juillet 1995.
- [16] Jean-Bernard ROUX <Jean-Bernard.Roux@Saussure.etat-ge.ch>, *Le pavage de Truchet*, <http://hypo.ge-dip.etat-ge.ch/www/math/html/node56.html>, 1995.
- [17] Cyril SMITH et Pauline BOUCHER, « The Tiling Patterns of Sébastien Truchet and the Topology of Structural Hierarchy », *Leonardo*, Vol. 20, N° 4, pages 373-385, 1987.
- [18] Rouben TER MINASSIAN <terminassian@cal.enst.fr>, *Pavage de Truchet. Travail présenté au concours PStricks 1995*, juin 1995, disponible auprès de l'auteur ou des rédacteurs du présent article.
- [19] Sébastien TRUCHET, « Mémoire sur les combinaisons », *Mémoires de l'Académie Royale des Sciences*, pages 363-372, 1704 (une partie du texte — en traduction anglaise... — et des reproductions de certaines planches figurent dans [17]).