Université de Montréal

Dynamique bicomplexe et théorème de Bloch pour fonctions hyperholomorphes

par

Dominic Rochon

Département de mathématiques et de statistique

Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures

en vue de l'obtention du grade de

Philosophiæ Doctor (Ph.D.)
en mathématiques fondamentales

novembre 2000

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-61393-3

Canada

Université de Montréal

Faculté des études supérieures

Cette thèse intitulée

Dynamique bicomplexe et théorème de Bloch pour fonctions hyperholomorphes

présentée par

Dominic Rochon

a été évaluée par un jury composé des personnes suivantes :

Marlène Frigon
(président-rapporteur)

Paul M. Gauthier
(directeur de recherche)

Quazi Ibadur Rahman
(membre du jury)

John Ryan
(examinateur externe)

David London
(représentant du doyen)

Thèse acceptée le :
30 mars 2001

La liberté peut être entravée par le mal de ne pouvoir explorer tous les mondes rencontrés.

-Sharina

SOMMAIRE

Le premier article de cette thèse vise à introduire la dynamique bicomplexe en utilisant une généralisation commutative des nombres complexes appelée les nombres bicomplexes. En particulier, nous donnons une généralisation de l'ensemble de Mandelbrot et des ensembles de "Julia-rempli" en dimension trois et quatre. Aussi, nous établissons que notre version de l'ensemble de Mandelbrot établie avec des polynômes quadratiques en nombres bicomplexes de la forme $w^2 + c$ est identiquement l'ensemble des points pour lesquels la généralisation de l'ensemble de "Julia-rempli" est connexe. De plus, nous prouvons que notre généralisation de l'ensemble de Mandelbrot de dimension quatre est connexe.

Dans le second article, nous utilisons aussi les nombres bicomplexes afin de montrer que la sous-classe des applications holomorphes de $\mathbb{C}^2$, satisfaisant les équations de Cauchy-Riemann complexifiées, a une constante de Bloch dans $\mathbb{C}^2$. De plus, nous trouvons des estimations de cette constante lorsque les applications sont sur la boule unité et nous donnons un domaine spécifique de $\mathbb{C}^2$ où la constante de Bloch a la même valeur que la constante de Bloch classique d'une variable complexe.

Table des matières

Table des figures

INTRODUCTION AU CHAPITRE 1

En 1982, A. Norton [**23**] nous présente certains algorithmes simples pour la génération et la représentation de formes fractales en 3-D. Pour la première fois, l'itération de quaternions [**19**] apparaît. Par la suite, certains résultats théoriques ont été traités dans [**13**] pour l'ensemble de Mandelbrot quaternionique défini avec le polynôme quadratique en quaternions de la forme q^2+c. Cependant, dans [**3**], S. Bedding and K. Briggs ont établi qu'il n'existe pas de dynamique intéressante avec cette approche et qu'elle ne joue aucun rôle fondamental analogue à l'application $z^2 + c$ pour le plan complexe. Nous notons qu'une autre définition de l'ensemble de Mandelbrot pour les quaternions a été introduite par J. Holbrook dans [**18**]. Cette définition nous donne un ensemble de Mandelbrot qui n'est pas une tranche de l'ensemble de Mandelbrot quaternionique défini avec le polynôme quadratique.

Dans ce premier article, nous utilisons une généralisation commutative des nombres complexes appelés nombres bicomplexes ([**24**], [**25**], [**26**], [**28**]) afin de donner une nouvelle version de l'ensemble de Mandelbrot en dimension trois et quatre. De plus, nous prouvons que notre généralisation en dimension quatre, notée $\mathcal{M}_2$, est un ensemble connexe. Aussi, nous définissons le concept d'ensemble de "Julia-rempli" pour les nombres bicomplexes et nous prouvons qu'un point est à l'intérieur de $\mathcal{M}_2$ si et seulement si l'ensemble de "Julia-rempli" à ce point est connexe. Ces deux résultats sont parfaitement analogues aux résultats correspondants pour le plan complexe.

Notre généralisation de l'ensemble de Mandelbrot en dimension trois est établie à partir d'une tranche de $\mathcal{M}_2$. Nous donnons aussi une représentation graphique de notre ensemble, appelé le "tétrabrot", dans $\mathbb{R}^3$ et portons spécialement notre attention sur les couches de divergence à l'infini pour approcher l'ensemble. De plus, nous donnons des représentations graphiques d' ensembles de "Julia-rempli" associés à des points sur le tétrabrot et nous notons que la forme de certains ensembles de "Julia-rempli" sont le reflet de la forme du tétrablrot près des points correspondants. Cette caractéristique avait aussi été remarquée pour l'ensemble de Mandelbrot dans le plan complexe.

Finalement, nous remarquons que le tétrabrot pourrait possiblement être non-connexe et nous établissons des hypothèses sur la géométrie de l'ensemble de Mandelbrot pour lesquelles le tétrabrot serait non-connexe.

Chapitre 1

ARTICLE 1: "A GENERALIZED MANDELBROT SET FOR BICOMPLEX NUMBERS"

Article accepté pour publication dans la revue *"Fractals"*. [1]

1.1. INTRODUCTION

In 1982, A. Norton [**23**] gave some straightforward algorithms for the generation and display in 3-D of fractal shapes. For the first time, iteration with quaternions [**19**] appeared. Subsequently, theoretical results have been treated in [**13**] for the quaternionic Mandelbrot set defined with quadratic polynomial in the quaternions of the form $q^2 + c$. However, in [**3**], S. Bedding and K. Briggs established that there is no interesting dynamics for this approach and it does not play any fundamental role analogous to that for the map $z^2 + c$ in the complex plane. We note that another definition of a Mandelbrot set for the quaternions was introduced by J. Holbrook in [**18**]. This definition gives a Mandelbrot set in $\mathbb{R}^3$ which is not a slice of the quaternionic quadratic Mandelbrot set.

In this article, we use a commutative generalization of the complex numbers called bicomplex numbers ([**24**], [**25**], [**26**], [**28**]) to give a new version of the

---

1. Recherche supportée par le FCAR Québec.

Mandelbrot set in dimensions three and four. Moreover, we prove that our generalization in dimension four, noted $\mathcal{M}_2$, is a connected set. We also define the concept of "filled-Julia" set for bicomplex numbers and we prove that a point is inside $\mathcal{M}_2$ if and only if the "filled-Julia" set at that point is connected. These two results are perfectly analogous to the corresponding results in the complex plane.

Our generalization of the Mandelbrot set in dimension three is established from a slice of $\mathcal{M}_2$. We also give a graphics representation of our set, called the Tetrabrot, in $\mathbb{R}^3$ and we especially focus our attention on the infinite divergence layers to approach this set. Moreover, we give a graphics representation of the associated "filled-Julia" set for points on the Tetrabrot and we note that shapes of certain "filled-Julia" sets are reflected in the shape of the Tetrabrot near the corresponding points. This feature had also been remarked for the Mandelbrot set in the complex plane.

Finaly, we remark that the Tetrabrot could possibly be unconnected and we establish hypotheses about the geometry of the Mandelbrot set for which the Tetrabrot would be unconnected.

1.2. PRELIMINARIES

Here, we introduce some of the basic results of the theory of bicomplex numbers. First, we define bicomplex numbers as follows:
$\mathbb{C}_2 := \{a + bi_1 + ci_2 + dj : {i_1}^2 = {i_2}^2 = -1, j^2 = 1$ and $i_2 j = j i_2 = -i_1, i_1 j = j i_1 = -i_2, i_2 i_1 = i_1 i_2 = j\}$ where $a, b, c, d \in \mathbb{R}$. In this article, the norm used on $\mathbb{C}_2$ is the Euclidean norm (also noted $|\ \ |$) of $\mathbb{R}^4$.

We remark that we can write a bicomplex number $a + bi_1 + ci_2 + dj$ as $(a+bi_1)+(c+di_1)i_2 = z_1 + z_2 i_2$ where $z_1, z_2 \in \mathbb{C}_1 := \{x + y i_1 : {i_1}^2 = -1\}$. Thus,

$\mathbb{C}_2$ can be veiwed as the complexification of the usual complex numbers $\mathbb{C}_1$ and a bicomplex number can be seen as an element of $\mathbb{C}^2$. Moreover, the norm of the bicomplex number is the same as the norm of the associated element (z_1, z_2) of $\mathbb{C}^2$. It is easy to see [24] that $\mathbb{C}_2$ is a commutative unitary ring with the following characterization for the noninvertible elements:

Proposition 1. *Let $w = a + bi_1 + ci_2 + dj \in \mathbb{C}_2$. Then w is noninvertible iff $(a = -d$ and $b = c)$ or $(a = d$ and $b = -c)$ iff $z_1{}^2 + z_2{}^2 = 0$.*

It is also important to know that every bicomplex number $z_1 + z_2 i_2$ has the following unique idempotent representation:

$$z_1 + z_2 i_2 = (z_1 - z_2 i_1)e_1 + (z_1 + z_2 i_1)e_2$$

where $e_1 = \frac{1+j}{2}$ and $e_2 = \frac{1-j}{2}$.

This representation is very useful because: addition, multiplication and division can be done term-by-term. Also, an element will be noninvertible iff $z_1 - z_2 i_1 = 0$ or $z_1 + z_2 i_1 = 0$. The next definition will be useful to construct a natural "disc" in $\mathbb{C}_2$.

Definition 1. *We say that $X \subseteq \mathbb{C}_2$ is a $\mathbb{C}_2$-cartesian set determined by X_1 and X_2 if $X = X_1 \times_e X_2 : = \{z_1 + z_2 i_2 \in \mathbb{C}_2 : z_1 + z_2 i_2 = w_1 e_1 + w_2 e_2, (w_1, w_2) \in X_1 \times X_2\}$.*

In [24] it is shown that if X_1 and X_2 are domains of $\mathbb{C}_1$ then $X_1 \times_e X_2$ is also a domain of $\mathbb{C}_2$. Then, a manner to construct a natural "disc" in $\mathbb{C}_2$ is to take the $\mathbb{C}_2$-cartesian product of two discs in $\mathbb{C}_1$. Hence, we define the natural "disc" of $\mathbb{C}_2$ as follows [24]: $D(0,r) : = B^1(0,r) \times_e B^1(0,r) = \{z_1 + z_2 i_2 : z_1 + z_2 i_2 = w_1 e_1 + w_2 e_2, |w_1| < r, |w_2| < r\}$ where $B^n(0,r)$ is the open ball of $\mathbb{C}_1^n \backsimeq \mathbb{C}^n$ with radius r.

1.3. The Generalized Mandelbrot Set

In this section, we want to give a version of the Mandelbrot set for the bicomplex numbers. First, we recall the definition of the Mandelbrot set for the complex plane:

Definition 2. *Let* $P_c(z) = z^2 + c$ *where* $z, c \in \mathbb{C}$ *and* $P_c^{\circ n} := (P_c^{\circ(n-1)} \circ P_c)(z)$. *Then the Mandelbrot set is defined as follows:* $\mathcal{M} = \{c \in \mathbb{C} : P_c^{\circ n}(0)$ *is bounded* $\forall n \in \mathbb{N}\}$. *When we take* $z, c \in \mathbb{C}_1$, *we denote the Mandelbrot set by* $\mathcal{M}_1$.

Figure 1 gives an illustration of the Mandelbrot set with some of its "filled-Julia" sets. In fact, our figure is a rotation by 90° of the original Mandelbrot set. This rotation will give a better vantage point when we shall work on our version of the Mandelbrot set in $\mathbb{R}^3$. Also, the colours around the Mandelbrot set have been determined by the number of iterations needed before $|P_c^{\circ n}(0)| > 2$. This is well justified by the fact that the Mandelbrot set can also be characterized as follows: $\mathcal{M} = \{c \in \mathbb{C} : |P_c^{\circ n}(0)| \leq 2 \ \forall n \in \mathbb{N}\}$ [**7**]. Then, the colours give information about the manner in which the algorithm for the Mandelbrot set diverges to infinity. This information will be almost the only possible one to approach our version of the Mandelbrot set in dimension three.

We also recall the following beautiful property of the Mandelbrot set [**10**]:

Theorem 1 (Douady and Hubard, 1982). *The Mandelbrot set* $\mathcal{M}$ *is connected.*

Now, to give a version of the Mandelbrot set for the bicomplex numbers we have only to reproduce the algorithm of Definition 2 for the bicomplex numbers. This is the next definition.

Definition 3. *Let* $P_c(w) = w^2 + c$ *where* $w, c \in \mathbb{C}_2$ *and* $P_c^{\circ n}(w) := (P_c^{\circ(n-1)} \circ P_c)(w)$. *Then the generalized Mandelbrot set for bicomplex numbers is defined as follows:* $\mathcal{M}_2 = \{c \in \mathbb{C}_2 : P_c^{\circ n}(0)$ *is bounded* $\forall n \in \mathbb{N}\}$.

The next lemma is a characterization of $\mathcal{M}_2$ using only $\mathcal{M}_1$. This lemma will be useful to prove that $\mathcal{M}_2$ is also a connected set.

Lemma 1. $\mathcal{M}_2 = \mathcal{M}_1 \times_e \mathcal{M}_1$.

Proof. First, we prove that $\mathcal{M}_2 \subseteq \mathcal{M}_1 \times_e \mathcal{M}_1$. Let $c \in \mathbb{C}_2$ such that $P_c^{\circ n}(0)$ is bounded $\forall n \in \mathbb{N}$. We have

$$P_c(w) = w^2 + c = [(z_1 - z_2 i_1)^2 + (c_1 - c_2 i_1)]e_1 + [(z_1 + z_2 i_1)^2 + (c_1 + c_2 i_1)]e_2$$

where $w = (z_1 - z_2 i_1)e_1 + (z_1 + z_2 i_1)e_2$ and $c = (c_1 - c_2 i_1)e_1 + (c_1 + c_2 i_1)e_2$. Then,

$$P_c^{\circ n}(w) = P_{c_1 - c_2 i_1}^{\circ n}(z_1 - z_2 i_1)e_1 + P_{c_1 + c_2 i_1}^{\circ n}(z_1 + z_2 i_1)e_2.$$

By hypothesis,

$$P_c^{\circ n}(0) = P_{c_1 - c_2 i_1}^{\circ n}(0)e_1 + P_{c_1 + c_2 i_1}^{\circ n}(0)e_2 \text{ is bounded } \forall n \in \mathbb{N}.$$

Hence, $P_{c_1 - c_2 i_1}^{\circ n}(0)$ and $P_{c_1 + c_2 i_1}^{\circ n}(0)$ are also bounded $\forall n \in \mathbb{N}$. Then $c_1 - c_2 i_1$, $c_1 + c_2 i_1 \in \mathcal{M}_1$ and $c = (c_1 - c_2 i_1)e_1 + (c_1 + c_2 i_1)e_2 \in \mathcal{M}_1 \times_e \mathcal{M}_1$.

Conversely, if we take $c \in \mathcal{M}_1 \times_e \mathcal{M}_1$, we have $c = (c_1 - c_2 i_1)e_1 + (c_1 + c_2 i_1)e_2$ with $c_1 - c_2 i_1$, $c_1 + c_2 i_1 \in \mathcal{M}_1$. Hence, $P_{c_1 - c_2 i_1}^{\circ n}(0)$ and $P_{c_1 + c_2 i_1}^{\circ n}(0)$ are also bounded $\forall n \in \mathbb{N}$. Then $P_c^{\circ n}(0)$ is bounded $\forall n \in \mathbb{N}$, that is $c \in \mathcal{M}_2$. □

Theorem 2. *The generalized Mandelbrot set* $\mathcal{M}_2$ *is connected.*

Proof. Define a mapping e as follows:

$$\mathbb{C}_1^2 = \mathbb{C}_1 \times \mathbb{C}_1 \xrightarrow{e} \mathbb{C}_1 \times_e \mathbb{C}_1 = \mathbb{C}_2$$

$$(w_1, w_2) \longmapsto w_1 e_1 + w_2 e_2.$$

The mapping e is clearly a homeomorphism. Then, if X_1 and X_2 are connected subsets of $\mathbb{C}_1$ we have that $e(X_1 \times X_2) = X_1 \times_e X_2$ is also connected. Now, by Lemma 1, $\mathcal{M}_2 = \mathcal{M}_1 \times_e \mathcal{M}_1$. Moreover, by Theorem 1, $\mathcal{M}_1$ is connected. It follows, if we let $X_1 = X_2 = \mathcal{M}_1$, that $\mathcal{M}_2$ is connected.□

1.4. THE TETRABROT

In the previous section, we established a version of the Mandelbrot set in dimension four. Now, we want to give a version of the Mandelbrot set in dimension three using the definition for $\mathcal{M}_2$. The idea here is to preserve the Mandelbrot set inside $\mathcal{M}_2$. Then, if we restict the algorithm to the points of the form $a + bi_1 + ci_2$ where $a, b, c \in \mathbb{R}$, we preserve the Mandelbrot set on two perpendicular complex planes and we stay in $\mathbb{R}^3$. This is the first argument to justify the following definition.

Definition 4. *The "Tetrabrot" is defined as follows:* $\mathcal{T} = \{c = c_1 + c_2 i_2 \in \mathbb{C}_2 : Im(c_2) = 0$ *and* $P_c^{\circ n}(0)$ *is bounded* $\forall n \in \mathbb{N}\}$.

We wish to give an illustration of the Tetrabrot in $\mathbb{R}^3$. The next result will give a manner to approach the Tetrabrot with the Euclidian norm in $\mathbb{R}^4$.

Theorem 3. $\mathcal{M}_2 \subset \overline{D}(0,2) \subset \overline{B^2(0,2)}$ *where* $\overline{D}(0,2) = \overline{B^1(0,2)} \times_e \overline{B^1(0,2)}$. *Moreover, the radius* 2 *is the best possible in each case.*

Proof. By Lemma 1, $\mathcal{M}_2 = \mathcal{M}_1 \times_e \mathcal{M}_1$. Moreover, $\overline{D}(0,2) = \overline{B^1(0,2)} \times_e \overline{B^1(0,2)}$ and $\mathcal{M}_1 \subset \overline{B^1(0,2)}$ with a point of $\mathcal{M}_1$ which touches the boundary of

this disc [7]. Then, $\mathcal{M}_2 \subset \overline{D}(0,2)$ and the radius 2 is the best possible. Finaly, it is proven in [24] that $\overline{D}(0,2) \subset \overline{B^2(0,2)}$ with points of $\overline{D}(0,2)$ which touche the boundary of $\overline{B^2(0,2)}$.□

Then, it is possible to compute the infinite divergence layers of the Tetrabrot from the number of iterations needed to have $|P_c^{\circ n}(0)| > 2$. We have to remark here that each divergence layer will hide the others. For example, Fig. 2 is an illustration for the Tetrabrot of one of its divergence layers in correspondence with the divergence layer illustrated in Fig. 1.A for the Mandelbrot set. In fact, the Tetrabrot is inside Fig. 2. It is possible to see a part of the Tetrabrot (see Fig. 3) if we cut a piece of Fig. 2. In Fig. 3, the colours are an illustration of the other divergence layers. It is also possible to compute others divergence layers (see Figs. 4, 5 ,6 and 7). Figure 7 begins to be close to the set that we wish to approach; then Fig. 7 with its cut plane gives certainly a good idea of the Tetrabrot.

To define the Tetrabrot, we have put the last coordinate in "j" equal to zero. In fact it is possible to do the same things if we fix the last coordinate equal to a number different from zero. However, if we do that, we lose the beautiful symmetry of the Tetrabrot. Figures 8 and 9 gives an illustration of this phenomenon for two different fixed "dj" with $d \neq 0$.

An interesting exploration of the Tetrabrot is now possible. For example Figure 10 is an enlargement of Fig. 7.A. It is also possible to be more specific. For example, Fig. 14 is an enlargement of 10.A and Figs. 11 and 12 are an enlargement of deep zones above the zone of Fig 7.A. Also, Fig. 13 is an enlargement of Fig 7.B. The colour in the background of Fig. 14 has been added to give a better 3-dimensional view. Each figure has been illustrated with a selected divergence layer and striations have been added to give an illustration of the "level curves" of each figure.

1.5. THE GENERALIZED "FILLED-JULIA" SET

It is now interesting to see what happens with the Julia set. First, we recall the definition of the "filled Julia" set in the complex plane:

Definition 5. *The "filled-Julia" set is defined as follows:* ($c \in \mathbb{C}$)

$$\mathcal{K}_c = \{z \in \mathbb{C} : P_c^{\circ n}(z) \text{ is bounded } \forall n \in \mathbb{N}\}.$$

Moreover, the Julia set $\mathcal{J}_c := \partial\mathcal{K}_c$.

We recall also the following beautiful relationship between the Mandelbrot set and the "filled-Julia" set:

Theorem 4. $c \in \mathcal{M} \Leftrightarrow \mathcal{K}_c$ *is connected.*

It is possible to generalize the "filled-Julia" set for bicomplex numbers:

Definition 6. *The generalized "filled-Julia" set for bicomplex numbers is defined as follows:* ($c \in \mathbb{C}_2$)

$$\mathcal{K}_{2,c} = \{w \in \mathbb{C}_2 : P_c^{\circ n}(w) \text{ is bounded } \forall n \in \mathbb{N}\}.$$

The next lemma gives a charaterization of the "filled-Julia" set for bicomplex numbers in terms of the "filled-Julia" set for the complex plane. This lemma will be useful to give an analogue of Theorem 4 for the bicomplex numbers.

Lemma 2. $\mathcal{K}_{2,c} = \mathcal{K}_{2,(c_1-c_2i_1)e_1+(c_1+c_2i_1)e_2} = \mathcal{K}_{c_1-c_2i_1} \times_e \mathcal{K}_{c_1+c_2i_1}$.

Proof. The proof is along the same lines as the proof of the Lemma 1.□

Theorem 5. $c \in \mathcal{M}_2 \Leftrightarrow \mathcal{K}_{2,c}$ *is connected.*

Proof. By Lemma 2, we know that $\mathcal{K}_{2,c} = \mathcal{K}_{c_1-c_2i_1} \times_e \mathcal{K}_{c_1+c_2i_1}$. Also, by the homeomorphism in the proof of Theorem 2, $\mathcal{K}_{c_1-c_2i_1} \times_e \mathcal{K}_{c_1+c_2i_1}$ is connected if and only if $\mathcal{K}_{c_1-c_2i_1} \times \mathcal{K}_{c_1+c_2i_1}$ is connected. Then, $\mathcal{K}_{c_1-c_2i_1} \times_e \mathcal{K}_{c_1+c_2i_1}$ is connected

if and only if $\mathcal{K}_{c_1-c_2i_1}$ and $\mathcal{K}_{c_1+c_2i_1}$ are connected. Hence, by Theorem 4, $\mathcal{K}_{2,c}$ is connected if and only if $c_1 - c_2i_1$, $c_1 + c_2i_1 \in \mathcal{M}_1$. Therefore, by Lemma 1, $\mathcal{K}_{2,c}$ is connected if and only if $c = (c_1 - c_2i_1)e_1 + (c_1 + c_2i_1)e_2 \in \mathcal{M}_2$.□

1.6. THE "FILLED-JULIA" SET FOR THE TETRABROT

The same process as for the Tetrabrot yields a version of the "filled-Julia" set in $\mathbb{R}^3$. We define the "filled-Julia" set for the Tetrabrot.

Definition 7. *The associated "filled-Julia" set for the Tetrabrot is defined as follows:* ($c \in \mathbb{C}_2$)

$$\mathcal{L}_{2,c} = \{w = w_1 + w_2i_2 \in \mathbb{C}_2 : Im(w_2) = 0 \text{ and } P_c^{\circ n}(w) \text{ is bounded } \forall n \in \mathbb{N}\}.$$

Figure 15 is an illustration of the "filled-Julia" set for the Tetrabrot at the same point $c = -1.754878$ as the "filled-Julia" set B of Fig. 1. Hence, Fig. 15 is a kind of generalization of the "filled-Julia" set $\mathcal{K}_c$ in the complex plane. In the same manner, Figure 16 is the generalization of Fig. 1.C, and Fig. 17 the generalization of Fig. 1.D.

In [7], L. Carleson and T. W. Gamelin have remarked this interesting fact: "One striking feature of $\mathcal{M}$ is that shapes of certain of the Julia sets $\mathcal{J}_c$ in dynamic space (z-space) are reflected in the shape (c-shape).". For the Tetrabrot, we seem to have something similar. For example, Fig. 18 is Fig. 17 with the same kind of cut as for the Tetrabrot in Fig. 7. Hence we see that inside Fig. 17 we have the same shape as inside the Tetrabrot near the point $c = 0,25$. It is also possible to see the same phenomenon with the "filled-Julia" set of Fig. 16. This phenomenon has been illustrated in Fig. 19 where we have put together the border of the Tetrabrot and the associated "filled-Julia" set at the point $c = -1.16 - 0.25i_1$ on

the border. We see clearly that this "filled-Julia" set imitates the border of the Tetrabrot.

Finaly, in Figs. 20, 21, 22 and 23 we show the "filled-Julia" set at $c = i_1$ for different infinite divergence layers. We remark that Fig. 23 is a good approximation of this set and an interesting generalization of Fig. 1.E.

1.7. CONJECTURE

We have proved in Sec. 3 that $\mathcal{M}_2$ is a connected set. It is natural to ask whether the Tetrabrot is also connected. Until now, the exploration of the Tetrabrot seems to confirm that the Tetrabrot is connected. However, if we enlarge Fig. 7 in the centre of the Tetrtabrot above the zone B, we notice (see Fig. 24.A) that there is a piece which seems to be disconnected from the main figure (Fig. 25 focuses on this piece). Because we work with divergence layers and a computational approximation, we are far from knowing if the piece is really unconnected or if there is point inside the piece which is also inside the Tetrabrot. However this is enough to formulate a conjecture:

Conjecture 1. *The Tetrabrot is unconnected.*

It is possible to translate the conjecture into a question about the geometry of the Mandelbrot set. For this, we need to prove the following lemma which is itself of interest:

Lemma 3. *The Tetrabrot can be characterized as follows:*

$$\mathcal{T} = \bigcup_{y \in [-m,m]} \{[(\mathcal{M}_1 - yi_1) \cap (\mathcal{M}_1 + yi_1)] + yi_2\}$$

where $m := \sup\{q \in \mathbb{R} : \exists p \in \mathbb{R} \text{ such that } p + qi_1 \in \mathcal{M}_1\}$.

Proof. By definition,

$$\mathcal{T} = \{c = c_1 + c_2 i_2 \in \mathbb{C}_2 : Im(c_2) = 0 \text{ and } P_c^{on}(0) \text{ is bounded } \forall n \in \mathbb{N}\}.$$

Let $c = (c_1 - c_2 i_1)e_1 + (c_1 + c_2 i_1)e_2$ with $c_1 = c_{11} + c_{12} i_1$ and $c_2 = c_{21} + c_{22} i_1$ where $c_{11}, c_{12}, c_{21}, c_{22} \in \mathbb{R}$. Now, if $Im(c_2) = 0$, we have $c_2 = c_{21} + 0 i_1$ and therefore, $c = (c_1 - c_{21})e_1 + (c_1 + c_{21} i_1)e_2$ whenever $Im(c_2) = 0$. Hence $\mathcal{T} = \{(c_1 - c_{21} i_1)e_1 + (c_1 + c_{21} i_1)e_2 : P_c^{on}(0) \text{ is bounded } \forall n \in \mathbb{N}\} = \{(c_1 - c_{21} i_1)e_1 + (c_1 + c_{21} i_1)e_2 : P_{c_1 - c_{21} i_1}^{on}(0) \text{ and } P_{c_1 + c_{21} i_1}^{on}(0) \text{ are bounded } \forall n \in \mathbb{N}\}$.To continue the proof, we need to remark the following fact: $\forall z \in \mathbb{C}_1$,

$$\{c \in \mathbb{C}_1 : P_{c+z}^{on}(0) \text{ is bounded } \forall n \in \mathbb{N}\} = \mathcal{M}_1 - z.$$

By definition, $P_{c_1 - c_{21} i_1}^{on}(0)$ and $P_{c_1 + c_{21} i_1}^{on}(0)$ are bounded $\forall n \in \mathbb{N}$ if and only if $c_1 - c_{21} i_1$, $c_1 + c_{21} i_1 \in \mathcal{M}_1$, and by the remark, it is also if and only if $c_1 \in (\mathcal{M}_1 - c_{21} i_1) \cap (\mathcal{M}_1 + c_{21} i_1)$. Hence, if we express $(c_1 - c_{21} i_1)e_1 + (c_1 + c_{21})e_2 = c_1 + c_{21} i_2 = c_{11} + c_{12} i_1 + c_{21} i_2$, the Tetrabrot can be characterized as follows:

$$\mathcal{T} = \{c_{11} + c_{12} i_1 + c_{21} i_2 : c_{11} + c_{12} i_1 \in (\mathcal{M}_1 - c_{21} i_1) \cap (\mathcal{M}_1 + c_{21} i_1)\}$$

$$= \bigcup_{y \in \mathbb{R}} \{[(\mathcal{M}_1 - y i_1) \cap (\mathcal{M}_1 + y i_1)] + y i_2\}.$$

It is possible to be more precise with the last expression. In fact,

$$= \bigcup_{y \in [-m,m]} \{[(\mathcal{M}_1 - y i_1) \cap (\mathcal{M}_1 + y i_1)] + y i_2\}$$

because $(\mathcal{M}_1 - y i_1) \cap (\mathcal{M}_1 + y i_1) = \emptyset$ whenever $y \in [-m, m]^c$. This comes from the fact that $\mathcal{M}_1 \subset \{z \in \mathbb{C}_1 : |Im(z)| \leq m\}$.

Moreover, $(\mathcal{M}_1 - y i_1) \cap (\mathcal{M}_1 + y i_1) \neq \emptyset \; \forall y \in [-m, m]$. To see this, we just have to prove that $E_y := \{c = c_{11} + 0 i_1 + y i_2 : P_c^{on}(0) \text{ is bounded } \forall n \in \mathbb{N}\}$ is nonempty $\forall y \in [-m, m]$ because $E_y \subset \{c = c_{11} + c_{12} i_1 + c_{21} i_2 : P_c^{on}(0) \text{ is bounded } \forall n \in \mathbb{N}\} = \{c_{11} + c_{12} i_1 + c_{21} i_2 : c_{11} + c_{12} i_1 \in (\mathcal{M}_1 - c_{21} i_1) \cap (\mathcal{M}_1 + c_{21} i_1)\}$. In fact, the

set E_y is the algorithm for the Mandelbrot set iterates, with the imaginary part in "i_2" fixed at y. By the compactness and the symmetry of the Mandelbrot set $\mathcal{M}_1$, there must exist x_m such that $x_m - mi_2$, $x_m + mi_2 \in E_m$. Therefore, because $\mathcal{M}_1$ is connected, we must have $E_y \neq \emptyset \ \forall y \in [-m, m]$.

Theorem 6. *Let*

$$R_1 : = R(-1.3939 + 0.0848i_1; \ -1.3893 + 0.0803i_1),$$

$$L : = R(-1.3939 + 0.0803i_1; \ -1.3939 + 0.0728i_1),$$

$$R_2 : = R(-1.3939 + 0.0728i_1; \ -1.3893 + 0.0683i_1),$$

$$L_3 : = R(-1.3939 + 0.1304i_1; \ -1.3893 + 0.1304i_1),$$

$$L_4 : = R(-1.3939 + 0.1259i_1; \ -1.3893 + 0.1259i_1),$$

$$L_5 : = R(-1.3893 + 0.1304i_1; \ -1.3893 + 0.1259i_1),$$

$$L_6 : = R(-1.3939 + 0.1304i_1; \ -1.3939 + 0.1259i_1),$$

where

$$R(a+bi_1, c+di_1) : = \{\alpha_1(a+bi_1)+\alpha_2(c+bi_1)+\alpha_3(a+di_1)+\alpha_4(c+di_1) | \sum_{ß=1}^{4} \alpha_i = 1, \ \alpha_i \geq 1\}.$$

If

$$F_1 : = R_1 \cup L \cup R_2$$

and

$$F_2 : = L_3 \cup L_4 \cup L_5$$

are disjoint from the Mandelbrot set and,

$$z_1^* : = -1.391816306 + 0.129472959i_1$$

and

$$z_2^* : = -1.392873019 + 0.077172405i_1$$

are inside the Mandelbrot set, then the Tetrabrot is unconnected.

Proof. The goal of the proof is to construct, from the hypothesis about the Mandelbrot set, a box where the algorithm for the Tetrabrot diverges with, inside the box, a point where the algorithm converges. The box that we want to construct is a box around the piece of Fig. 25. Also, to understand better with which zones we work, Fig. 26.A gives an indication where the zones of the hypothesis are on the Mandelbrot set (Figs. 27, 28 and 29 are enlargements of Fig. 26.A where the specific sets F_1, F_2, z_1^* and z_2^* are illustrated).

The "box of divergence" is constructed as follows: let $y_1 : = 0.0228$ and $y_2 : = 0.0288$,

$$B_i : = R_i + y_i i_1 - y_i i_2 \text{ for i=1,2,}$$

$$B_i : = \bigcup_{y \in [y_1, y_2]} (L_i - y i_1 - y i_2) \text{ for i=3,...,6.}$$

Each "B_i" is a side of the box; then the "box of divergence" is $B : = \bigcup_{i=1}^{6} B_i$.

First, we have to confirm that each "B_i" is a set where the algorithm for the Tetrabrot diverges. This is possible with Lemma 3 and the assumption that $F_1 \cup F_2$ is not in the Mandelbrot set.

For B_1, by Lemma 3, we just need to prove that:

$$B_1 \cap ([(\mathcal{M}_1 - y_1 i_1) \cap (\mathcal{M}_1 + y_1 i_1)] - y_1 i_2) = \emptyset.$$

This is clear because if $B_1 \cap ([(\mathcal{M}_1 - y_1 i_1) \cap (\mathcal{M}_1 + y_1 i_1)] - y_1 i_2) \neq \emptyset$, we obtain that there exists $z_1 \in R_1$ such that $z_1 + y_1 i_1 \in [(\mathcal{M}_1 - y_1 i_1) \cap (\mathcal{M}_1 + y_1 i_1)]$. However, this is impossible because if $z_1 + y_1 i_1 \in \mathcal{M}_1 + y_1 i_1$ we obtain that $z_1 \in \mathcal{M}_1$ and this contradicts the hypothesis.

A similar proof is possible for B_2. The cases of B_3, B_4 and B_5 are also along the same lines. For example:

$$B_3 \cap ([(\mathcal{M}_1 - yi_1) \cap (\mathcal{M}_1 + yi_1)] - yi_2) = \emptyset \ \forall y \in [y_1, y_2],$$

because if it is not true, there must exist $y \in [y_1, y_2]$ and $z_3 \in L_3$ such that $z_3 - yi_1 \in [(\mathcal{M}_1 - yi_1) \cap (\mathcal{M}_1 + yi_1)]$. However, this is impossible because if $z_3 - yi_1 \in \mathcal{M}_1 - yi_1$ we obtain that $z_3 \in \mathcal{M}_1$ and this contradicts the hypothesis. For B_6, the same argument is possible if we remark that $L_6 - yi_1 \subset F_1 + yi_1$ $\forall y \in [y_1, y_2]$.

Now, we have to confirm that each B_i is on the same box. For this, we will just remark the following fact:

$$L_i - y_k i_1 - y_k i_2 \subset B_k = R_k + y_k i_1 - y_k i_2 \ \forall i = 3, ..., 6 \text{ and } \forall k = 1, 2$$

because $L_i - y_k i_1 \subset R_k + y_k i_1$ $\forall i = 3, ...6$ and $\forall k = 1, 2$. Then each B_i, for $i = 3, ...6$, touches B_1 and B_2 at their extremity. To be more specific and to confirm that the B_i form a box, we have to check directely with the exact coordinates given in the hypothesis of the theorem.

Finally, we have to prove, with the assumptions of the theorem, that there is a point inside the box B for which the algorithm for the Tetrabrot converges. For this, we remark that z_2^* is between R_1 and R_2. Moreover, $Re(z_1^*) > Re(z_2^*)$; then we must have in the set $A := \{x + yi_1 \in \mathbb{C}_1 : x = Re(z_1^*) \text{ and } 0.0728 < y < 0.0803\}$ a point $z^* \in \mathcal{M}_1$. If not, by hypothesis, the Mandelbrot set would not be connected because z_2^* would be separated from z_1^* by $F_1 \cup A$ since $(F_1 \cup A) \cap \mathcal{M}_1 = \emptyset$.

Now, let

$$y^* := \frac{Im(z_1^*) - Im(z^*)}{2}$$

where $z_1^*, z^* \in \mathcal{M}_1$. We note that $z_1^* - y^* i_1 = z^* + y^* i_1$ and $y^* \in [-1, 1] \subset [-m, m]$. Then by Lemma 3, $z_1^* - y^* i_1 - y^* i_2 \in \mathcal{T}$ because $z_1^* - y^* i_1 - y^* i_2 = z^* + y^* i_1 - y^* i_2 \in$

$[(\mathcal{M}_1 - y^* i_1) \cap (\mathcal{M}_1 + y^* i_1)] - y^* i_2 \subset \mathcal{T}$. Moreover, $z_1^* - y^* i_1 - y^* i_2$ is inside the "box of divergence" because $y_1 < y^* < y_2$ and z_1^* is inside the rectangle formed by $L_3 \cup L_4 \cup L_5 \cup L_6$.□

1.8. Conclusion

The last theorem is a good indication that the conjecture is true because the hypothesis about the Mandelbrot set can be approximately confirmed by computers with a high level of precision. To confirm that the conjecture is true, we have two choices: to demonstrate theoretically the hypothesis about the geometry of the Mandelbrot set or to prove more directly that the Tetrabrot is unconnected. If the conjecture is proven to be true, a new question could be to know the cardinality of the family of the connected components. Also, it could be interesting to know whether the "filled-Julia" set associated with points on an unconnected piece of the Tetrabrot have some specific proporties such as to be also unconnected. Finaly, another pertinent question could be to know the local fractal dimension of the boundary of the Tetrabrot.

Figure 1

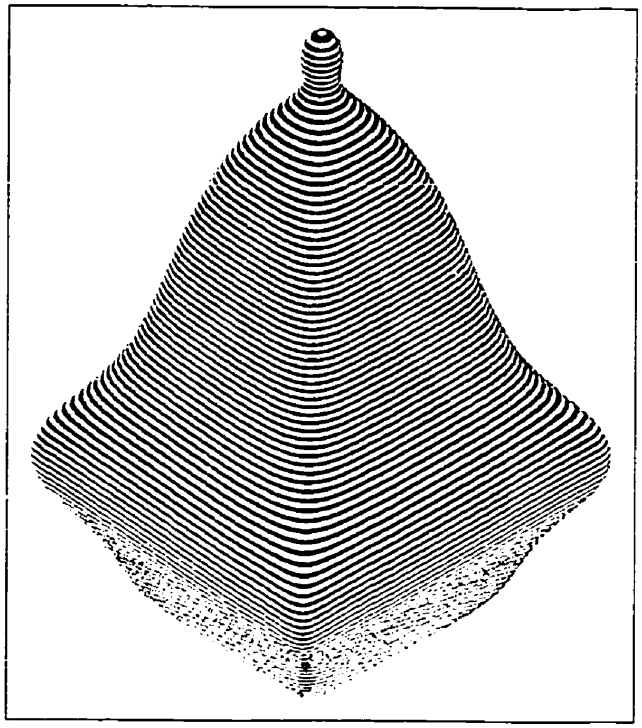

Figure 2

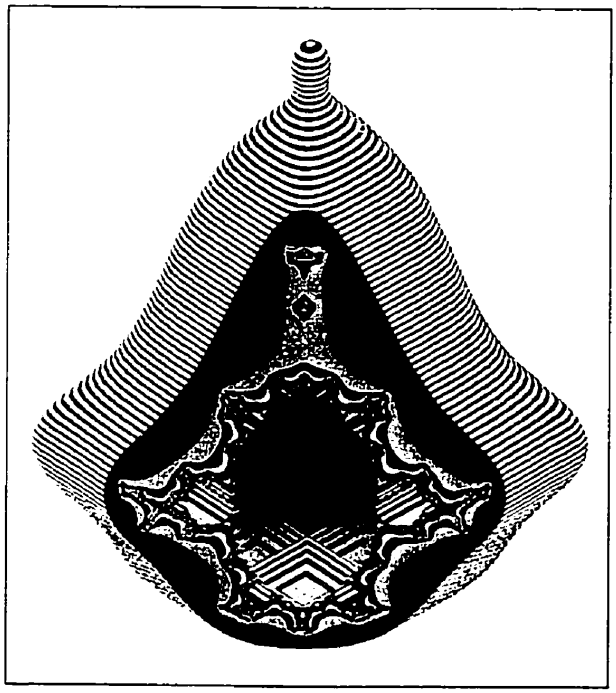

Figure 3

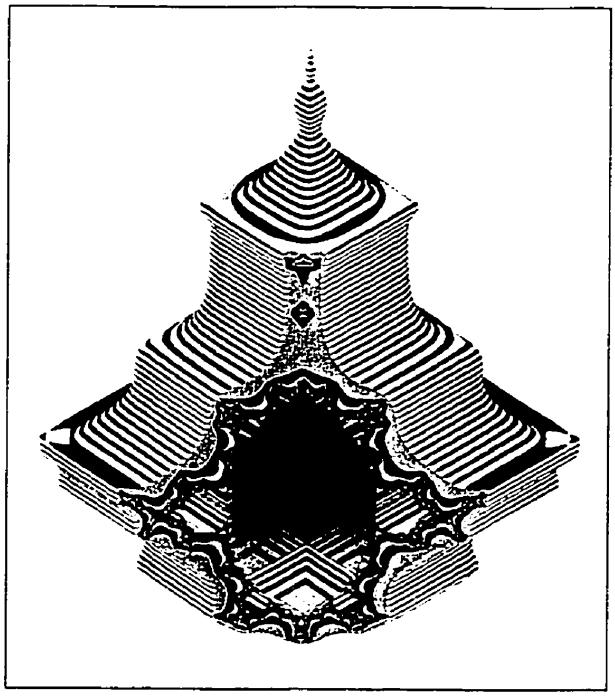

Figure 4

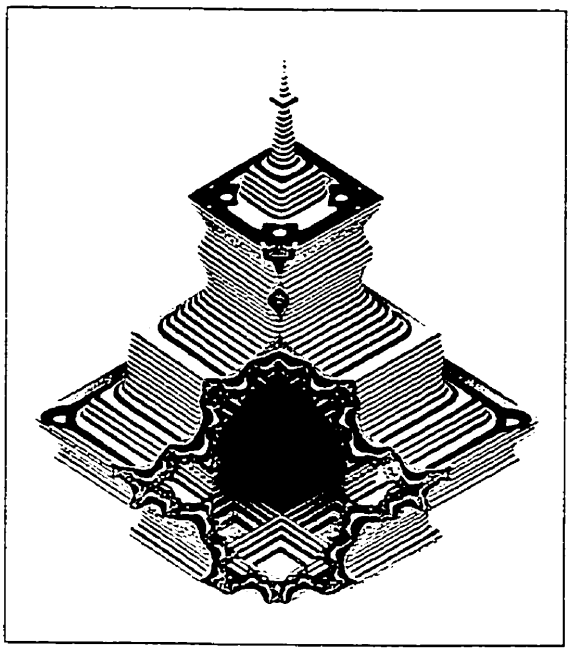

Figure 5

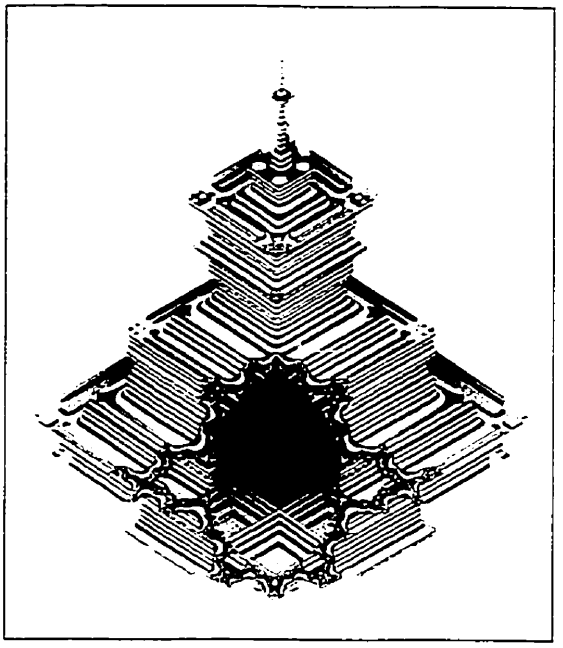

Figure 6

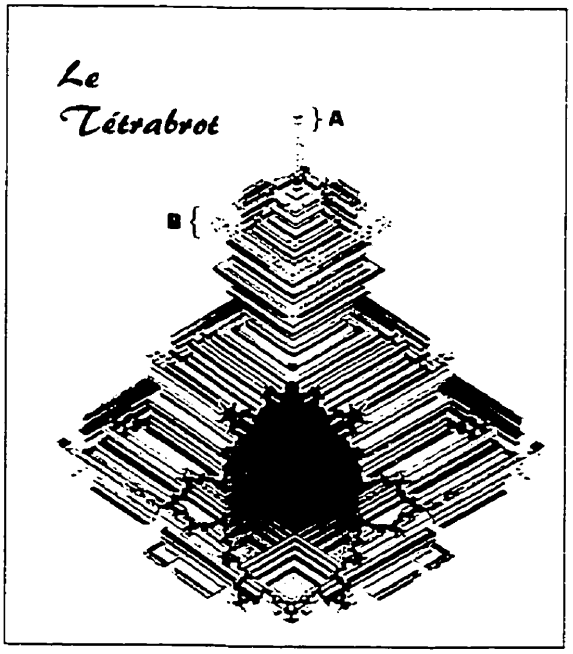

Figure 7

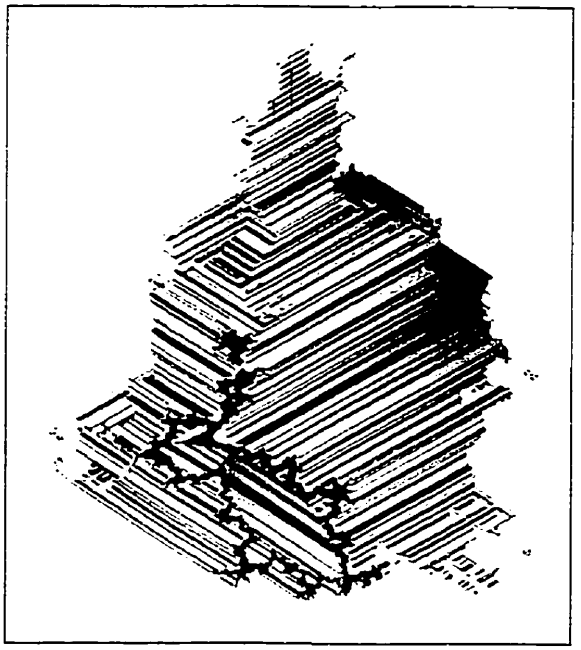

Figure 8

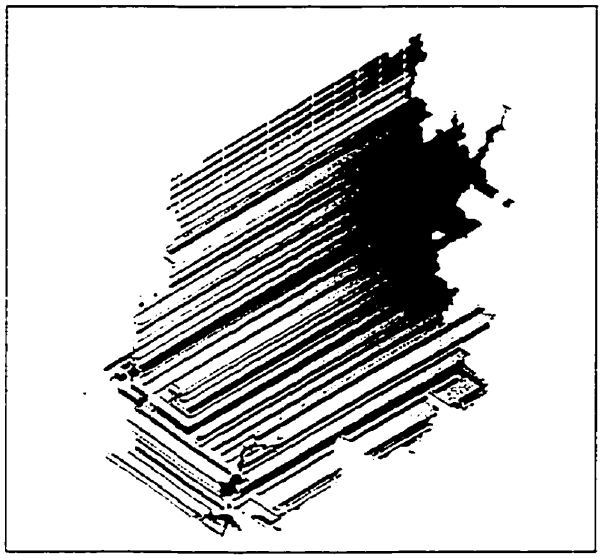

Figure 9

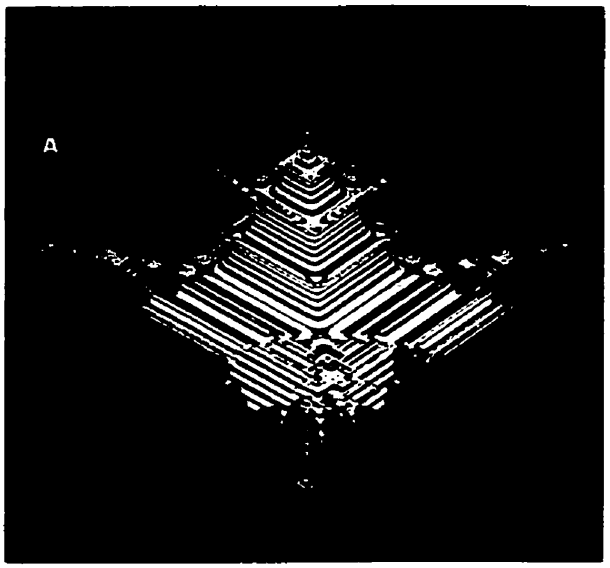

Figure 10

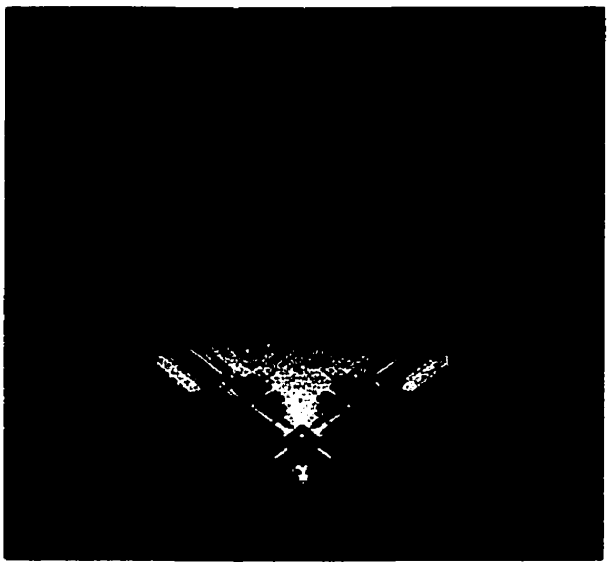

Figure 11

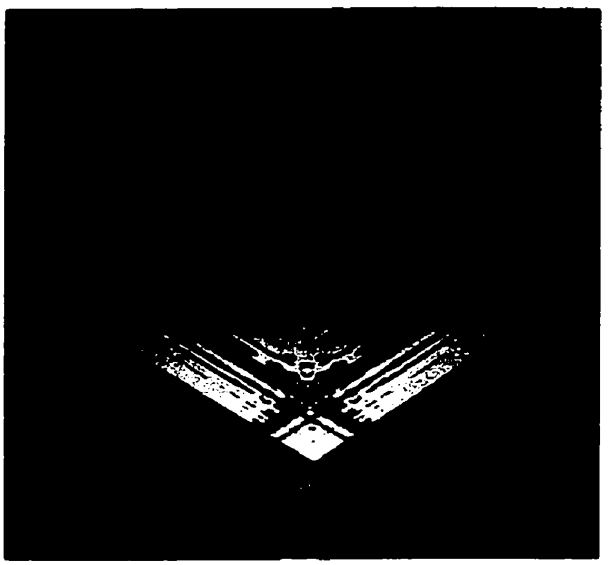

Figure 12

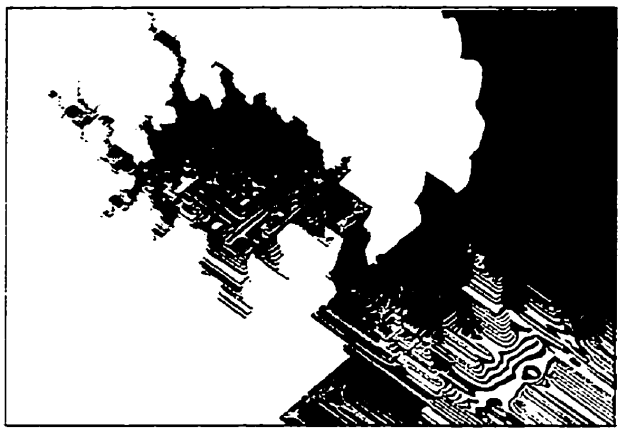

Figure 13

Figure 14

Figure 15

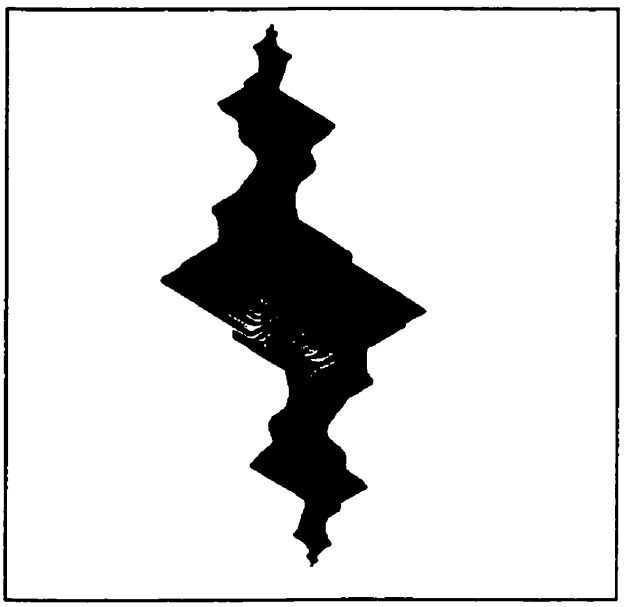

Figure 16

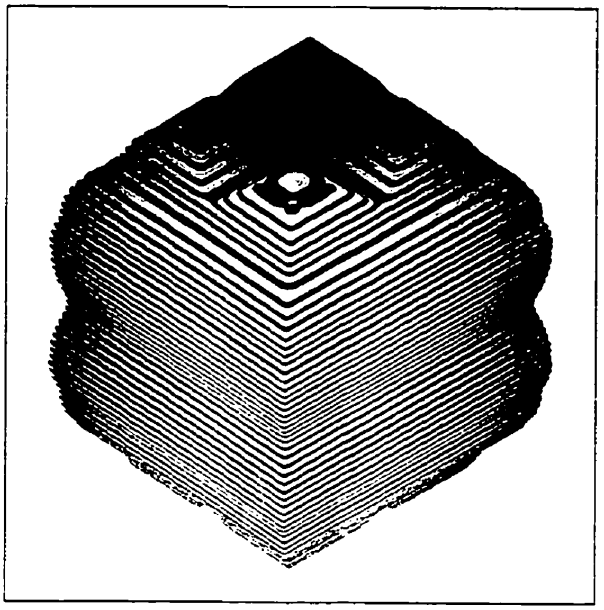

Figure 17

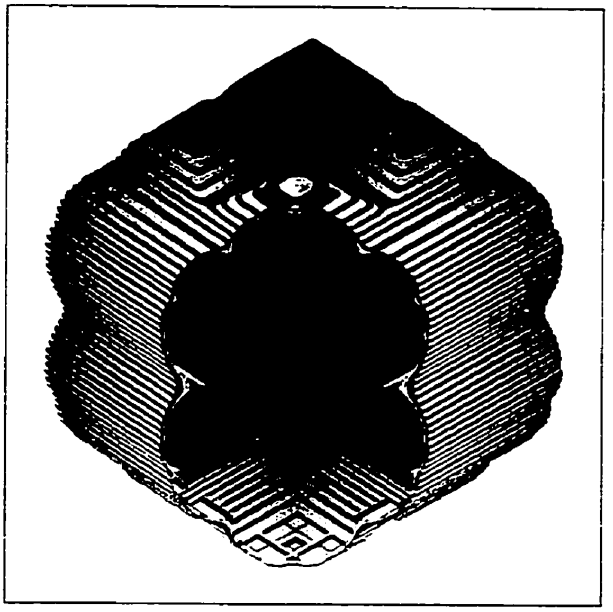

Figure 18

Figure 19

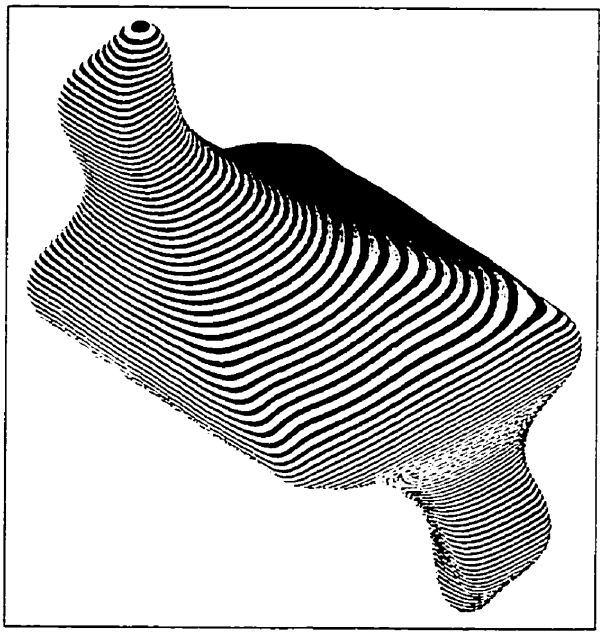

Figure 20

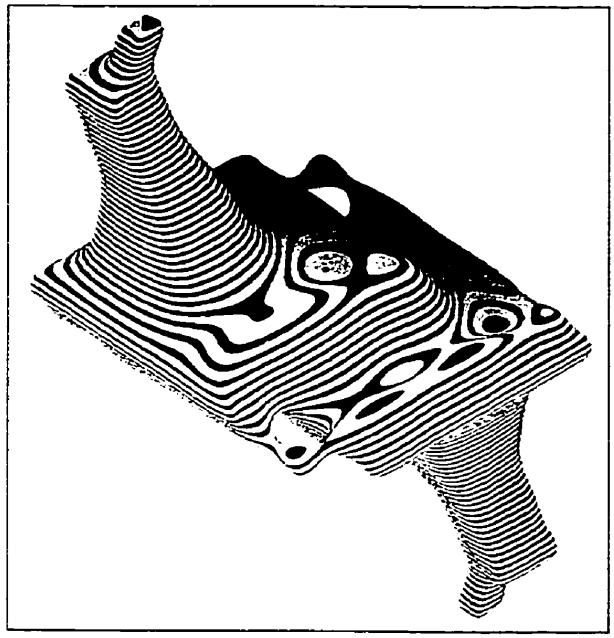

Figure 21

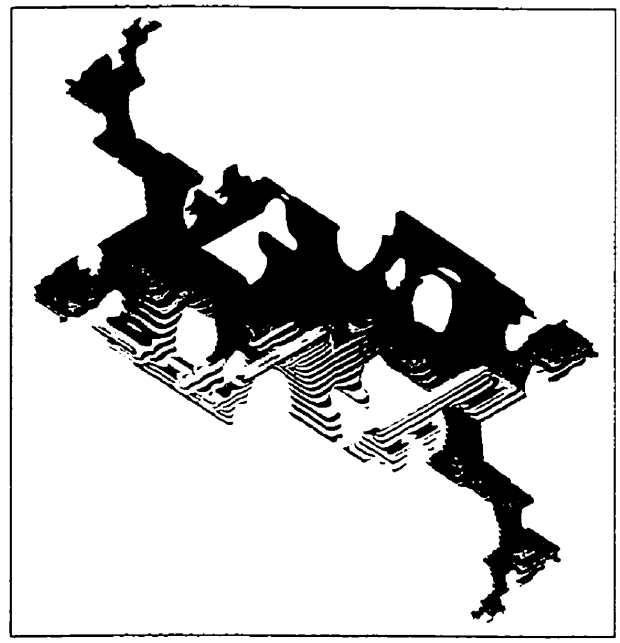

Figure 22

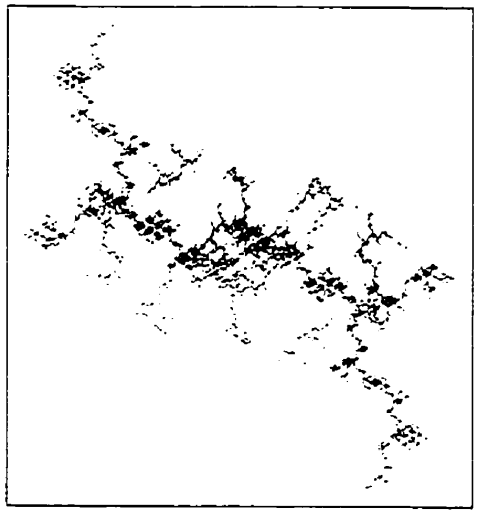

Figure 23

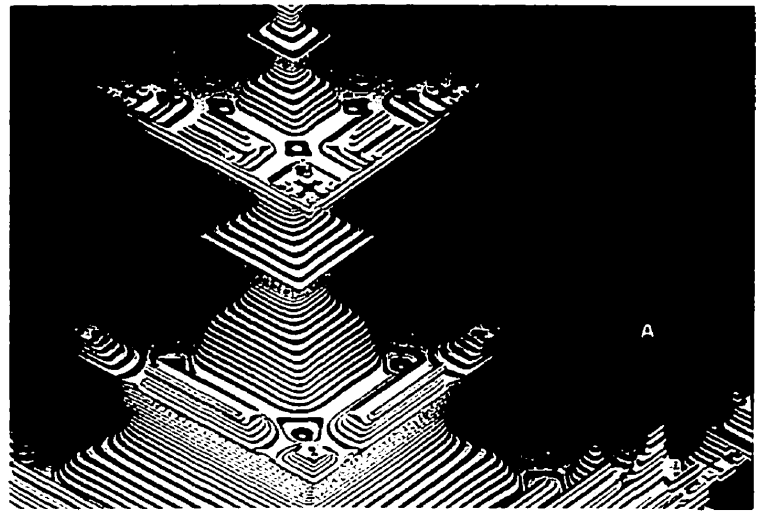

Figure 24

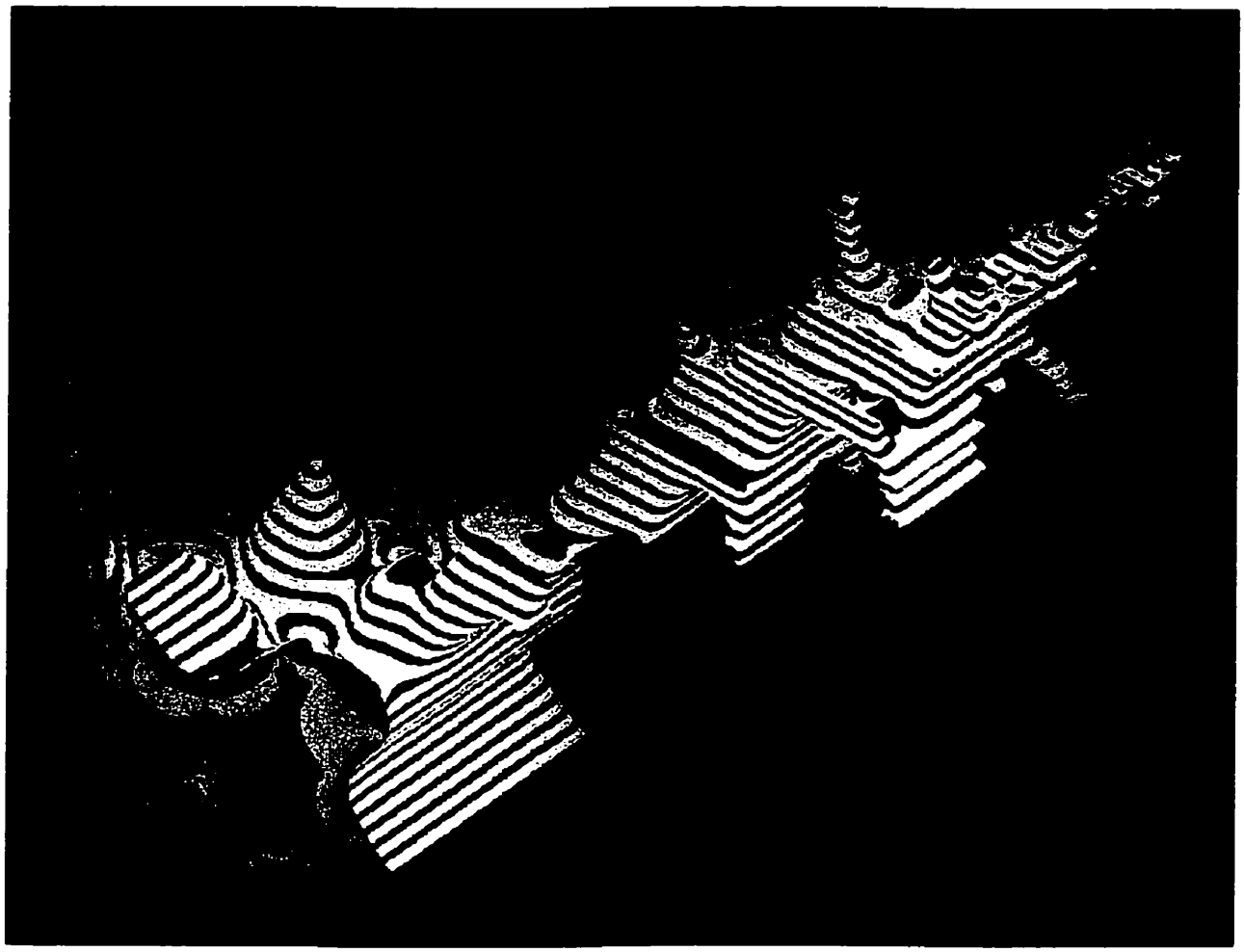

Figure 25

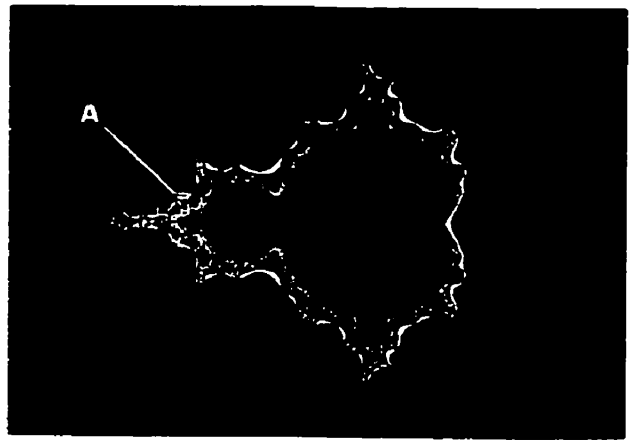

Figure 26

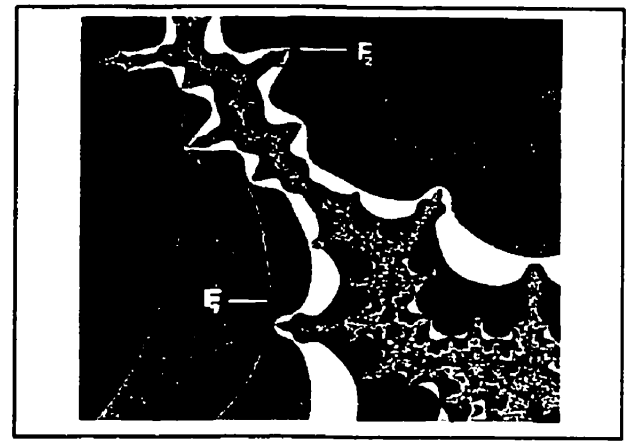

Figure 27

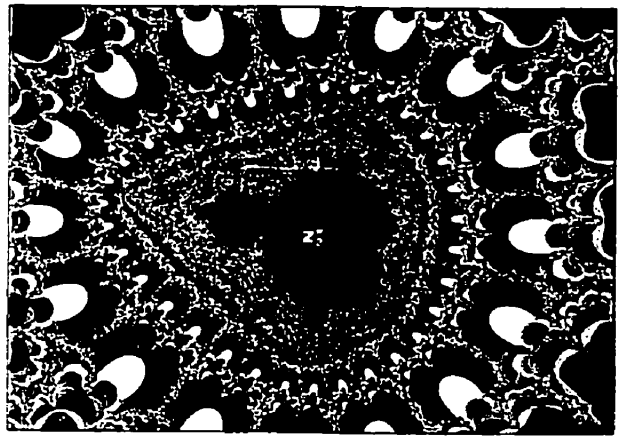

Figure 28

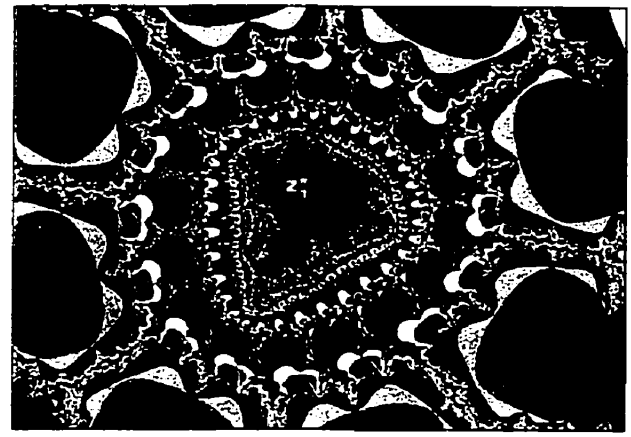

Figure 29

INTRODUCTION AU CHAPITRE 2

Depuis la confirmation de la conjecture de Bieberbach par de Branges, un des plus remarquable problème ouvert en analyse complexe est de trouver la valeur exacte de la constante de Bloch.

Soit $H(B)$ la classe de fonctions $w = f(z)$ holomorphes dans le disque unité $B = \{z \in \mathbb{C} : |z| < 1\}$. En 1925, Bloch [**4**] prouve le fameux théorème qui porte son nom:

Théorème 1.8.1 (Bloch). *Il existe une constante positive b tel que si $f \in H(B)$ et $f'(0) \neq 0$, alors f applique un certain sous-domaine de B de façon biholomorphe à un disque de rayon $b \cdot |f'(0)|$.*

Un tel disque est appelé un disque univalent pour f. La constante de Bloch peut être décrite comme:

$$\beta = \inf\{\beta_f : f \in H(B) \text{ avec } f'(0) = 1\},$$

où $\beta_f = \sup\{b : f(B)$ contient un disque univalent de rayon $b\}$. Dans ce deuxième article, nous introduisons d'autres classes d'applications et nous en trouvons une qui est exactement égale à la constante de Bloch classique β.

Les estimations supérieures et inférieures suivantes pour β ont été trouvées par Lars Ahlfors et Grunsky [**2**] et Ahlfors [**1**]:

$$0.43\cdots = \frac{\sqrt{3}}{4} \leq \beta \leq \frac{\Gamma(1/3)\Gamma(11/12)}{\Gamma(1/4)(1+\sqrt{3})^{1/2}} = 0.47\cdots.$$

La conjecture actuelle est que la valeur correcte de β est précisément cette borne supérieure. Récemment, sur la base du travail de Bonk [**6**] et du lemme de Schwarz-Pick, Chen Huaihui et P. M. Gauthier [**8**] ont amélioré la borne inférieure, pour la constante de Bloch, de la manière suivante:

$$\frac{\sqrt{3}}{4} + 2 \cdot 10^{-4} \leq \beta.$$

Passant à plusieurs variables complexes, une application holomorphe f d'un domaine de $\mathbb{C}^n$ dans $\mathbb{C}^n$ est dite non-dégénérée si $det\mathcal{J}_f$ n'est pas identiquement zéro sur le domaine. Soit B^n, la boule unité dans $\mathbb{C}^n$. Une application f non-dégénérée de B^n dans $\mathbb{C}^n$ est dite normalisée si $det\mathcal{J}_f(0) = 1$, où 0 dénote l'origine dans $\mathbb{C}^n$. Pour un tel f, nous dénotons par β_f le suprémum des valeurs b tel que l'image $f(B^n)$ contient une boule univalente de rayon b.

Si nous fixons $K > 0$ et considérons l'application holomorphe $f : \mathbb{C}^2 \longrightarrow \mathbb{C}^2$ qui est défini par:

$$f(z_1, z_2) = (z_1/\sqrt{K}, \sqrt{K}z_2).$$

Alors, f est normalisée mais $\beta_f = 1/\sqrt{K}$. Puisque K peut être choisi arbitrairement grand, nous voyons qu'il n'existe pas de théorème de Bloch pour les applications holomorphes, lorsque $n > 1$.

On pourrait argumenter que la correcte généralisation de la normalisation "$f'(0) = 1$" exemples d'applications holomorphes f, avec cette normalisation plus forte $\mathcal{J}_f(0) = I$ (I étant l'application identitée) et pour lesquels β_f est arbitrairement petit. Un des résultats de cet article est de montrer que dans le cas de $\mathbb{C}^2$, la normalisation plus forte est correcte si l'application f satisfait aux équations de Cauchy-Riemann complexifiées.

Ainsi, nous voyons que pour $n > 1$ nous avons besoin de restreindre la classe des applications à une sous-classe plus spécifique pour obtenir un théorème de Bloch. Une des sous-classes bien connue, est la classe des applications K-quasirégulières. Pour $n \geq 1$, dénotons par $|\ |$ la norme usuelle dans $\mathbb{C}^n$.

Définition 1.8.1 (Wu). *Soit B^n la boule unité ouverte de $\mathbb{C}^n$ et soit $\mathcal{F} : B^n \longrightarrow \mathbb{C}^n$ une famille d'applications holomorphes. Nous disons que $\mathcal{F}$ est K-quasirégulière si et seulement si il existe une constante K tel que, pour chaque $f = (f_1, \ldots, f_n)$ de $\mathcal{F}$,*

$$\left|\frac{\partial f(z)}{\partial z_\sigma}\right| \leq K|\det \mathcal{J}_f(z)|^{1/n}, \text{ pour } \sigma = 1, \ldots, n \text{ et } \forall z \in B^n.$$

Pour une telle classe d'applications, il est possible de trouver un théorème de Bloch dans $\mathbb{C}^n$. Wu a trouvé une magnifique preuve de ceci dans [**33**]:

Théorème 1.8.2 (Wu, 1967). *Soit $\mathcal{F} : B^n \longrightarrow \mathbb{C}^n$ une classe d'applications holomorphes et K-quasirégulières tel que $|det\mathcal{J}_f(0)|=1$ pour tout $f \in \mathcal{F}$. Alors, il existe une constante positive c tel que toutes $f \in \mathcal{F}$ possèdent une boule univalente de rayon c.*

Comme Wu le remarqua, ce résultat s'obtient aussi du travail de Bochner [**5**] en 1946. Il est aussi possible de trouver des estimations inférieures pour la constante de Bloch pour cette classe d'applications dans [**9**], [**29**] et [**32**] en adaptant correctement les définitions. Nous notons aussi que les estimations dépendent du K de la quasirégularité.

Dans cet article, nous utilisons une généralisation commutative des nombres complexes appelée nombres bicomplexes ([**24**], [**28**] et [**26**]) pour trouver une autre sous-classe d'applications qui ont une constante de Bloch dans $\mathbb{C}^2$. De plus, nous trouvons les estimations: $\frac{\beta}{\sqrt{2}} \leq \delta \leq \sqrt{2}\beta$ pour cette constante de Bloch δ,

lorsque nos applications sont sur la boule unité, et nous trouvons un domaine spécifique de $\mathbb{C}^2$ où la constante de Bloch a la même valeur que la constante de Bloch d'une variable. Finalement nous montrons que cette classe d'applications ne dépend pas de la quasirégularité.

Chapitre 2

ARTICLE 2: "A BLOCH CONSTANT FOR HYPERHOLOMORPHIC FUNCTIONS"

Article accepté pour publication dans la revue *"Complex Variables"*.[1]

2.1. INTRODUCTION

Since confirmation of the Bieberbach conjecture by de Branges, perhaps the outstanding open problem in complex analysis is that of finding the exact value of the Bloch constant.

Let $H(B)$ be the class of functions $w = f(z)$ holomorphic in the unit disc $B = \{z \in \mathbb{C} : |z| < 1\}$. In 1925, Bloch [**4**] proved the famous theorem which bears his name:

Theorem 7 (Bloch). *There exists a positive constant b such that if $f \in H(B)$ and $f'(0) \neq 0$, then f maps some subdomain of B biholomorphically onto a disc of radius $b \cdot |f'(0)|$.*

Such a disc is called a univalent disc for f. The Bloch constant may be described as:

$$\beta = \inf\{\beta_f : f \in H(B) \text{ with } f'(0) = 1\}$$

---

1. Recherche supportée par le FCAR Québec.

where $\beta_f = \sup\{b : f(B)$ contains a univalent disc of radius b$\}$. In this paper, we introduce Bloch constants for other classes of mappings and find one which is precisely equal to the classical Bloch constant β.

The following upper and lower estimates for β were found by Lars Ahlfors and Grunsky [2] and Ahlfors [1]:

$$0.43\cdots = \frac{\sqrt{3}}{4} \leq \beta \leq \frac{\Gamma(1/3)\Gamma(11/12)}{\Gamma(1/4)(1+\sqrt{3})^{1/2}} = 0.47\cdots.$$

It is conjectured that the correct value of β is precisely this upper bound. Recently, on the basis of Bonk's work [6] and the Schwarz-Pick lemma, Chen Huaihui and P. M. Gauthier [8] improved the lower bound for Bloch's constant further as follows:

$$\frac{\sqrt{3}}{4} + 2 \cdot 10^{-4} \leq \beta.$$

Passing to several complex variables, a holomorphic mapping f from a domain in $\mathbb{C}^n$ into $\mathbb{C}^n$ is said to be nondegenerate if $det\mathcal{J}_f$ is not identically zero on the domain. Let B^n denote the open unit ball in $\mathbb{C}^n$. A nondegenerate mapping f from B^n into $\mathbb{C}^n$ is said to be normalized if $det\mathcal{J}_f(0) = 1$, where 0 denotes the origin in $\mathbb{C}^n$. For such f we denote by β_f the supremum of values b such that the image $f(B^n)$ contains a univalent ball of radius b.

If we fix $K > 0$ and consider the holomorphic mapping $f : \mathbb{C}^2 \longrightarrow \mathbb{C}^2$ defined by

$$f(z_1, z_2) = (z_1/\sqrt{K}, \sqrt{K}z_2),$$

then, f is normalized but $\beta_f = 1/\sqrt{K}$. Since K can be chosen arbitrarily large, we see that there is no Bloch theorem for general holomorphic mappings, when $n > 1$.

One might argue that the correct generalization of the normalization "$f'(0) = 1$" to several variables is not "$det\mathcal{J}_f(0) = 1$". However, there are also examples

of holomorphic mappings f, with the stronger normalization $\mathcal{J}_f(0) = I$ (I is the identity mapping) and for which β_f is arbitrarily small. One of the results of this article is to show that in the case of $\mathbb{C}^2$ the stronger normalization is correct, if the mapping f satisfies the complexified Cauchy-Riemann equations.

Thus, we see that for $n > 1$ we need to restrict the class of mappings to a more specific subclass to obtain a Bloch theorem. One of the well known subclasses is the class of K-quasiregular mappings. For $n \geq 1$, let $|\ |$ denote the usual norm in $\mathbb{C}^n$.

Definition 8 (Wu). *Let B^n be the open unit ball of $\mathbb{C}^n$ and let $\mathcal{F} : B^n \longrightarrow \mathbb{C}^n$ be a family of holomorphic mappings. We say $\mathcal{F}$ is K-quasiregular iff there exists a constant K so that, for each $f = (f_1, \ldots, f_n)$ of $\mathcal{F}$, the following holds throughout B^n,*

$$\left|\frac{\partial f(z)}{\partial z_\sigma}\right| \leq K|\det \mathcal{J}_f(z)|^{1/n}, \text{ for } \sigma = 1, \ldots, n.$$

For such a class of mappings, it is possible to find a Bloch theorem in $\mathbb{C}^n$. Wu found a beautiful proof of this in **[33]**:

Theorem 8 (Wu, 1967). *Let $\mathcal{F} : B^n \longrightarrow \mathbb{C}^n$ be a K-quasiregular family of holomorphic mappings such that $|det\mathcal{J}_f(0)|=1$ for all $f \in \mathcal{F}$. Then, there is a positive constant c such that every $f \in \mathcal{F}$ possesses a univalent ball of radius c.*

As Wu pointed out, this result also follows from the work of Bochner **[5]** in 1946. It is possible to find lower estimates for the Bloch constant for this class of mappings in **[9]**, **[29]** and **[32]** provided we adapt correctly the definitions. We note also that the estimates depend on the K of quasiregularity.

In this article, we use a generalization of complex numbers called bicomplex numbers (**[24]**, **[28]**, **[26]**) to find another subclass of mappings which has a Bloch

constant in $\mathbb{C}^2$. Moreover, we find the estimates: $\frac{\beta}{\sqrt{2}} \leq \delta \leq \sqrt{2}\beta$ for this Bloch constant δ, whenever our mappings are on the unit ball, and we find a specific domain of $\mathbb{C}^2$ where the Bloch constant has the same value as the Bloch constant for one variable. Finally we show that this class of mappings does not depend on the quasiregularity.

2.2. PRELIMINARIES

Here, we introduce some of the basic results of the theory of bicomplex numbers. First, we define bicomplex numbers as follows: $\mathbb{C}_2 := \{a + bi_1 + ci_2 + dj : {i_1}^2 = {i_2}^2 = -1, j^2 = 1 \text{ and } i_2 j = j i_2 = -i_1, i_1 j = j i_1 = -i_2, i_2 i_1 = i_1 i_2 = j\}$ where $a, b, c, d \in \mathbb{R}$. The norm used on $\mathbb{C}_2$ is the Euclidean norm (also noted $|\ \ |$) of $\mathbb{R}^4$.

We remark that we can write a bicomplex number $a + bi_1 + ci_2 + dj$ as $(a + bi_1) + (c + di_1)i_2 = z_1 + z_2 i_2$ where $z_1, z_2 \in \mathbb{C}_1 := \{x + yi_1 : {i_1}^2 = -1\}$. Thus, $\mathbb{C}_2$ can be veiwed as the complexification of $\mathbb{C}_1$ and a bicomplex number can be seen as an element of $\mathbb{C}^2$. It is easy to see [**24**] that $\mathbb{C}_2$ is a commutative unitary ring with the following characterization for the noninvertible elements:

Proposition 2. *Let $w = a + bi_1 + ci_2 + dj \in \mathbb{C}_2$. Then w is noninvertible iff $(a = -d$ and $b = c)$ or $(a = d$ and $b = -c)$ iff ${z_1}^2 + {z_2}^2 = 0$.*

It is also possible to define differentiability of a function at a point of $\mathbb{C}_2$ [**24**]:

Definition 9. *Let U be an open set of $\mathbb{C}_2$ and $w_0 \in U$. Then, $f : U \subseteq \mathbb{C}_2 \longrightarrow \mathbb{C}_2$ is said to be $\mathbb{C}_2$-differentiable at w_0 with derivative equal to $f'(w_0) \in \mathbb{C}_2$ if*

$$\lim_{\substack{w \to w_0 \\ (w - w_0 \text{ inv.})}} \frac{f(w) - f(w_0)}{w - w_0} = f'(w_0).$$

We will also say that the function f is $\mathbb{C}_2$-holomorphic on an open set U iff f is $\mathbb{C}_2$-differentiable at each point of U.

As we saw, a bicomplex number can be seen as an element of $\mathbb{C}^2$, so a function $f(z_1 + z_2 i_2) = f_1(z_1, z_2) + f_2(z_1, z_2)i_2$ of $\mathbb{C}_2$ can be seen as a mapping $f(z_1, z_2) = (f_1(z_1, z_2), f_2(z_1, z_2))$ of $\mathbb{C}^2$. Here we have a characterization of such mappings:

Theorem 9. *Let U be an open set and $f : U \subseteq \mathbb{C}_2 \longrightarrow \mathbb{C}_2$ such that $f \in C^1(U)$. Let also $f(z_1 + z_2 i_2) = f_1(z_1, z_2) + f_2(z_1, z_2)i_2$. Then f is $\mathbb{C}_2$-holomorphic on U iff:*

$$f_1 \text{ and } f_2 \text{ are holomorphic in } z_1 \text{ and } z_2$$

and,

$$\frac{\partial f_1}{\partial z_1} = \frac{\partial f_2}{\partial z_2} \text{ and } \frac{\partial f_2}{\partial z_1} = -\frac{\partial f_1}{\partial z_2} \text{ on } U.$$

Moreover, $f' = \frac{\partial f_1}{\partial z_1} + \frac{\partial f_2}{\partial z_1} i_2$ and $f'(w)$ is invertible iff $det \mathcal{J}_f(w) \neq 0$.

This theorem can be obtained from results in [**24**] and [**26**]. Moreover, by the Hartogs theorem [**31**], it is possible to show that "$f \in C^1(U)$" can be dropped from the hypotheses. Now, it is natural to define for $\mathbb{C}^2$ the following class of mappings introduced in [**26**]:

Definition 10. *The class of $\mathbb{T}$-holomorphic mappings on a open set $U \subseteq \mathbb{C}^2$ is defined as follows:*

$$TH(U) : = \{f : U \subseteq \mathbb{C}^2 \longrightarrow \mathbb{C}^2 \mid f \in H(U) \text{ and } \frac{\partial f_1}{\partial z_1} = \frac{\partial f_2}{\partial z_2}, \frac{\partial f_2}{\partial z_1} = -\frac{\partial f_1}{\partial z_2} \text{ on } U\}.$$

It is the subclass of holomorphic mappings of $\mathbb{C}^2$ satisfying the complexified Cauchy-Riemann equations.

In [**26**], bicomplex numbers were called tetranumbers and $\mathbb{C}_2$ was denoted by $\mathbb{T}$. In this article, we will use this notation when a definition can be written independently of the theory of bicomplex numbers .

We remark that $f \in TH(U)$ iff f is $\mathbb{C}_2$-holomorphic on U. It is also important to know that every bicomplex number $z_1 + z_2 i_2$ has the following unique idempotent representation:

$$z_1 + z_2 i_2 = (z_1 - z_2 i_1)e_1 + (z_1 + z_2 i_1)e_2$$

where $e_1 = \frac{1+j}{2}$ and $e_2 = \frac{1-j}{2}$.

This representation is very useful because: addition, multiplication and division can be done term-by-term. Also, an element will be noninvertible iff $z_1 - z_2 i_1 = 0$ or $z_1 + z_2 i_1 = 0$.

The notion of holomorphicity can also be seen with this kind of notation. For this we need to define the functions $h_1, h_2 : \mathbb{C}_2 \longrightarrow \mathbb{C}_1$ as $h_1(z_1 + z_2 i_2) = z_1 - z_2 i_1$ and $h_2(z_1 + z_2 i_2) = z_1 + z_2 i_1$. Also, we need the following definition:

Definition 11. *We say that $X \subseteq \mathbb{C}_2$ is a $\mathbb{C}_2$-cartesian set determined by X_1 and X_2 if $X = X_1 \times_e X_2 : = \{z_1 + z_2 i_2 \in \mathbb{C}_2 : z_1 + z_2 i_2 = w_1 e_1 + w_2 e_2, (w_1, w_2) \in X_1 \times X_2\}$.*

In [**24**] it is shown that if X_1 and X_2 are domains of $\mathbb{C}_1$ then $X_1 \times_e X_2$ is also a domain of $\mathbb{C}_2$. Now, it is possible to state the following striking theorems [**24**]:

Theorem 10. *If $f_{e1} : X_1 \longrightarrow \mathbb{C}_1$ and $f_{e2} : X_1 \longrightarrow \mathbb{C}_1$ are holomorphic functions of $\mathbb{C}_1$ on the domains X_1 and X_2 respectively, then the function $f : X_1 \times_e X_2 \longrightarrow \mathbb{C}_2$ defined as*

$$f(z_1 + z_2 i_2) = f_{e1}(z_1 - z_2 i_1)e_1 + f_{e2}(z_1 + z_2 i_1)e_2, \ \forall \ z_1 + z_2 i_2 \in X_1 \times_e X_2$$

is $\mathbb{C}_2$-holomorphic on the domain $X_1 \times_e X_2$.

Theorem 11. *Let X be a domain in $\mathbb{C}_2$, and let $f : X \longrightarrow \mathbb{C}_2$ be a $\mathbb{C}_2$-holomorphic function on X. Then there exist holomorphic functions $f_{e1} : X_1 \longrightarrow \mathbb{C}_1$ and $f_{e2} : X_2 \longrightarrow \mathbb{C}_1$ with $X_1 = h_1(X)$ and $X_2 = h_2(X)$, such that:*

$$f(z_1 + z_2 i_2) = f_{e1}(z_1 - z_2 i_1)e_1 + f_{e2}(z_1 + z_2 i_1)e_2, \ \forall \ z_1 + z_2 i_2 \in X.$$

We note here that X_1 and X_2 will also be domains of $\mathbb{C}_1$.

Finally we give some concrete simple examples of entire $\mathbb{T}$-holomorphic mappings which come from the bicomplex theory. First, the exponential mapping defined as:

$$e^{z_1+z_2 i_2} : = e^{z_1}(cos(z_2) + i_2 sin(z_2)) = e^{z_1 - z_2 i_1}e_1 + e^{z_1+z_2 i_2}e_2,$$

with $e^{w_1+w_2} = e^{w_1} \cdot e^{w_2}$, $\forall \ w_1, w_2 \in \mathbb{C}_2$ and $(e^w)' = e^w$, $\forall \ w \in \mathbb{C}_2$. Secondly, the polynomials of degree n, with bicomplex multiplication, defined as:

$$a_1 \cdot w^n + a_2 \cdot w^{n-1} + ... + a_n,$$

with $a_i \in \mathbb{C}_2$ for $i = 1, ..., n$ and $w \in \mathbb{C}_2$. We note that the derivative will be what we expect for polynomials.

2.3. Bloch constant for $\mathbb{T}$-holomorphic mappings

Now we are ready to prove that there is a Bloch constant for the class of $\mathbb{T}$-holomorphic mapping of $\mathbb{C}^2$ on the unit Ball. The proof requires the natural "disc" of $\mathbb{C}_2$ called the $\mathbb{C}_2$-disc and defined as follows: $D(a_1 + a_2 i_2, r_1, r_2) : = \{z_1 + z_2 i_2 : z_1 + z_2 i_2 = w_1 e_1 + w_2 e_2, |w_1 - (a_1 - a_2 i_1)| < r_1, |w_2 - (a_1 + a_2 i_1)| < r_2\}$. Also, we call $D(a_1 + a_2 i_2, r) : = D(a_1 + a_2 i_2, r, r)$ the $\mathbb{C}_2$-disc of radius r. This

kind of disc is in fact the $\mathbb{C}_2$-cartesian product of two discs of $\mathbb{C}_1$. Also, we need to remark that: $f'(0) = 1$ iff $\mathcal{J}_f(0) = I$ (the identity matrix).

Theorem 12. *There is a positive constant d such that if $f \in TH(B^2)$ with $f'(0) \in \mathbb{C}_1 \backslash \{0\}$, then f maps some subdomain of B^2 biholomorphically onto a ball of radius $d \cdot |f'(0)|$. In particular, if $\mathcal{J}_f(0) = I$, the radius is d.*

Proof. We note first that $D := = D(0,1) \subseteq B^2(0,1)$ [**24**]. In particular $f \in TH(D)$ and thus by Theorem 11, we can write $f = f_{e1}e_1 + f_{e2}e_2$ on D with f_{ei} holomorphic on D_i where D_i:=h_i(D)=$\{w_i \in \mathbb{C}_1 : |w_i| < 1\}$ for $i = 1, 2$. Also, $f'(0)$=$f'_{e1}(0)e_1 + f'_{e2}(0)e_2$ [**24**, Theorem 24.3] and then $f'(0)$ invertible implies $f'_{e1}(0) \neq 0$ and $f'_{e2}(0) \neq 0$. Then, by the Bloch theorem in one variable, there exists a positive constant b such that f_{ei} maps some subdomain G_i of D_i biholomorphically onto a disc $B_i(c_i, b \cdot |f'_{ei}(0)|)$.

Now, define

$$G := \{z_1 + z_2 i_2 \in \mathbb{C}_2 : z_1 + z_2 i_2 = w_1 e_1 + w_2 e_2, (w_1, w_2) \in G_1 \times G_2\}.$$

In fact, $G = G_1 \times_e G_2$ is a domain of $\mathbb{C}_2$ and $G \subseteq D(0,1) \subseteq B^2(0,1)$. Then, $f = f_{e1}e_1 + f_{e2}e_2$, with $f_{ei} : G_i \longrightarrow B_i(c_i, b \cdot |f'_{ei}(0)|)$ biholomorphic for $i = 1, 2$.

Let $c := c_1 e_1 + c_2 e_2$, then

$$f : G \longrightarrow D(c, b \cdot |f'_{e1}(0)|, b \cdot |f'_{e2}(0)|)$$

is $\mathbb{T}$-biholomorphic. By a result in [**24**], $B^2(c, min(\frac{r_1}{\sqrt{2}}, \frac{r_2}{\sqrt{2}})) \subseteq D(c, r_1, r_2)$ and so $B^2(c, min(b \cdot \frac{|f'_{e1}(0)|}{\sqrt{2}}, b \cdot \frac{|f'_{e2}(0)|}{\sqrt{2}})) \subseteq D(c, b \cdot |f'_{e1}(0)|, b \cdot |f'_{e2}(0)|)$. Thus, for the domain

$$G' := = f^{-1}(B^2(c, min(b \cdot \frac{|f'_{e1}(0)|}{\sqrt{2}}, b \cdot \frac{|f'_{e2}(0)|}{\sqrt{2}}) \subseteq G \subseteq D(0,1) \subseteq B^2(0,1),$$

the function f is $\mathbb{T}$-biholomorphic from $G' \subseteq B^2(0,1)$ to a ball of center c and of radius equal to $b \cdot \frac{min(|f'_{e1}(0)|, |f'_{e2}(0)|)}{\sqrt{2}}$.

Finally, we remark that $f'(0) \in \mathbb{C}_1 \backslash \{0\}$ implies $f'_{e1}(0) = f'_{e2}(0)$. Then, $d \cdot |f'(0)| = b \cdot \frac{min(|f'_{e1}(0)|, |f'_{e2}(0)|)}{\sqrt{2}}$ if and only if $d = \frac{b}{\sqrt{2}}$, because $|f'(0)| = \left[\frac{|f'_{e1}(0)|^2 + |f'_{e2}(0)|^2}{2}\right]^{1/2}$. □

We have just seen that there exists a Bloch constant for the class of $\mathbb{T}$-holomorphic mappings with $\mathcal{J}_f(0) = I$ on the unit ball and now we wish to estimate this Bloch constant. For this, we need first to work on the natural unit disc: $D(0,1)$, of bicomplex numbers.

Theorem 13. *Let $f \in TH(D)$ with $f'(0) \in \mathbb{C}_1 \backslash \{0\}$. Then there is a positive constant a such that f maps some subdomain of D biholomorphically onto a $\mathbb{C}_2$-disc of radius $a \cdot |f'(0)|$. In particular, if $\mathcal{J}_f(0) = I$, the radius is a.*

Proof. The proof is contained in the proof of Theorem 12.□

On this special domain of $\mathbb{C}_2$, it is possible to find the exact value of the Bloch constant. For this, we need to prove the following lemma, which is itself of interest:

Lemma 4. *Let $f : U \longrightarrow \mathbb{C}^2$ be a $\mathbb{T}$-holomorphic injection with U open, then f is a $\mathbb{T}$-biholomorphic mapping from U to $f(U)$.*

Proof. Because f is a holomorphic injection, we know **[21]** that $f(U)$ is open in $\mathbb{C}^2$, that f is a biholomorphic mapping from U to $f(U)$, and $det\mathcal{J}_f(z) \neq 0$, $\forall z \in U$. Thus $f'(z)$ will be an invertible number $\forall z \in U$.

Now, we want to prove that:

$$\lim_{\substack{w \to w_0 \\ (w - w_0 \text{ inv.})}} \frac{f^{-1}(w) - f^{-1}(w_0)}{w - w_0} \text{ exists, } \forall w_0 \in f(U).$$

Let $z = f^{-1}(w)$ and $z_0 = f^{-1}(w_0)$. Then for $w - w_0$ invertible we obtain: $\frac{f^{-1}(w) - f^{-1}(w_0)}{w - w_0} = \frac{z - z_0}{f(z) - f(z_0)}$. Also, because $f \in TH(U)$ and $f'(z_0)$ invertible we

have: $(f(z)-f(z_0))/(z-z_0)$ is invertible for z near z_0 and

$$\lim_{\substack{z\to z_0 \\ (z-z_0 \text{ inv.})}} \frac{1}{\left(\frac{f(z)-f(z_0)}{z-z_0}\right)} = \frac{1}{f'(z_0)} \text{ exists, } \forall z_0 \in U. \tag{2.3.2.3.1}$$

Thus, $\forall \epsilon > 0$, there exists $\delta_1 > 0$ such that $\left|\frac{z-z_0}{f(z)-f(z_0)} - \frac{1}{f'(z_0)}\right| < \frac{\epsilon}{2}$, whenever $|z-z_0| < \delta_1$ and $z-z_0$ is invertible. Choose $\delta > 0$ such that $|w-w_0| < \delta$ implies $|z-z_0| < \delta_1$. Then,

$$\left|\frac{z-z_0}{f(z)-f(z_0)} - \frac{1}{f'(z_0)}\right| < \frac{\epsilon}{2}, \tag{2.3.2.3.2}$$

whenever $|w-w_0| < \delta$ and both $w-w_0$ and $z-z_0$ are invertible. If $|w-w_0| < \delta$ and $w-w_0$ invertible but $z-z_0$ is not, there always exists $\epsilon' > 0$ such that $|(z+\epsilon')-z_0| < \delta_1$, both $(z+\epsilon')-z_0$ and $f(z+\epsilon')-f(z_0)$ are invertible and $\left|\frac{z-z_0}{f(z)-f(z_0)} - \frac{1}{f'(z_0)}\right| \leq \left|\frac{z-z_0}{f(z)-f(z_0)} - \frac{(z+\epsilon')-z_0}{f(z+\epsilon')-f(z_0)}\right| + \left|\frac{(z+\epsilon')-z_0}{f(z+\epsilon')-f(z_0)} - \frac{1}{f'(z_0)}\right| < \frac{\epsilon}{2} + \frac{\epsilon}{2}$. Thus, $|w-w_0| < \delta$ and $w-w_0$ invertible implies (2.3.2.3.2). This concludes the proof.□

It is now possible to describe the exact value of the Bloch constant for the class of $\mathbb{T}$-holomorphic mappings with $\mathcal{J}_f(0) = I$ on the unit $\mathbb{C}_2$-disc. In fact, it turns out, that it coincides precisely with the classical Bloch constant.

Notation 1. $\alpha := inf\{\alpha_f : f \in TH(D(0,1)) \text{ with } \mathcal{J}_f(0) = I\}$,

$\alpha_f := sup\{a : f(D(0,1)) \text{ contains a univalent } \mathbb{C}_2\text{-disc of radius } a\}$,

Theorem 14.

$$\alpha = \beta,$$

where β is the Bloch constant of one variable.

Proof. Again, let $f := f_{e1}e_1 + f_{e2}e_2$ on D with f_{ei} holomorphic on D_i where D_i:=h_i(D)=$\{w_i \in \mathbb{C}_1 : |w_i| < 1\}$ for i=1, 2. Moreover, suppose $f'_{ei}(0) = 1$ for i=1, 2; then, by the definition of the Bloch constant for one variable, $\forall \epsilon > 0$ there

exists a univalent disc of radius c_ϵ for f_{e1} and f_{e2} such that $c_\epsilon \geq \beta - \epsilon$. Hence, $\forall f \in TH(\mathcal{D})$ with $\mathcal{J}_f(0) = I$, $\alpha_f \geq \beta - \epsilon \ \forall \epsilon > 0$ and thus $\alpha \geq \beta$.

Also, we know by **[30]** that there exists $g \in H(B)$ such that:

$$g'(0) = 1 \text{ and } \beta_g = \beta.$$

Let us now define:

$$f(z_1 + z_2 i_2) : = g(z_1 - z_2 i_1)e_1 + g(z_1 + z_2 i_1)e_2.$$

Then, $f_1(z_1, z_2) = \frac{g(z_1 - z_2 i_1) + g(z_1 + z_2 i_1)}{2}$ and $f_2(z_1, z_2) = \frac{g(z_1 - z_2 i_1) - g(z_1 + z_2 i_1)}{2} i_1$ so by Theorem 10 and the remark after Theorem 9, $f \in TH(D)$ with $f'(0) = 1$. We want to show that for this f, $\alpha_f \leq \beta$. If not, i.e. $\alpha_f > \beta$, then $f(D(0,1))$ contains a univalent $\mathbb{C}_2$-disc of radius c' such that $c' > \beta$. Thus f maps a subdomain $G \subseteq D(0,1)$ biholomorphically onto a $\mathbb{C}_2$-disc of radius c'. But $f \in TH(D(0,1))$, so by Lemma 4:

$$f : G \to D(0, c') \text{ is } \mathbb{T}\text{-biholomorphic.}$$

This is a contradiction because Theorem 11 applied to f and f^{-1} forces g to map the subdomain $h_i(G) \in B$ biholomorphically onto a disc of radius c' for $i = 1, 2$.□

The followings definitions are used to prove the main result of this article:

Definition 12. *We say that f has a $\mathbb{T}$-univalent ball if f has a $\mathbb{T}$-biholomorphic univalent ball.*

Notation 2. $\delta : = inf\{\delta_f : f \in TH(B^2(0,1)) \text{ with } \mathcal{J}_f(0) = I\}$,

$\delta_f : = sup\{d : f(B^2(0,1)) \text{ contains a univalent ball of radius } d\}$,

$\delta' : = inf\{\delta'_f : f \in TH(B^2(0,1)) \text{ with } \mathcal{J}_f(0) = I\}$.

$\delta'_f : = sup\{d : f(B^2(0,1)) \text{ contains a } \mathbb{T}\text{-univalent ball of radius } d\}$,

It is now possible to find the following estimates for our Bloch constant on the unit ball:

Theorem 15.

$$\frac{\beta}{\sqrt{2}} \leq \delta \leq \sqrt{2}\beta,$$

where β is the Bloch constant of one variable.

Proof. First we prove that $\frac{\beta}{\sqrt{2}} \leq \delta' \leq \delta$. Suppose $f \in TH(B^2(0,1))$ with $\mathcal{J}_f(0) = I$. By the proof of Theorem 12, for every $b < \beta$, $f(B^2(0,1))$ contains, in fact, a $\mathbb{T}$-univalent ball of radius $d = \frac{b}{\sqrt{2}}$. In fact, $\forall \epsilon > 0$ there is a b_ϵ such that $b_\epsilon \geq \beta - \epsilon$, so setting $d_\epsilon = \frac{b_\epsilon}{\sqrt{2}}$, we have $d_\epsilon \geq \frac{\beta-\epsilon}{\sqrt{2}}$ which implies $\delta'_f \geq \frac{\beta-\epsilon}{\sqrt{2}}$, $\forall \epsilon > 0$ and thus

$$\delta'_f \geq \frac{\beta}{\sqrt{2}}.$$

Hence $\frac{\beta}{\sqrt{2}} \leq \inf \delta'_f : = \delta'$. Finally, because a $\mathbb{T}$-univalent ball for f is a univalent ball, we have $\delta'_f \leq \delta_f$ and then $\delta' \leq \delta$. In fact, by Lemma 4 we have $\delta = \delta'$.

The second part of the proof is to prove that $\delta \leq \sqrt{2}\beta$. We will prove this by contradiction. Suppose that there exists $\epsilon > 0$ such that $\delta > \sqrt{2}(\beta + \epsilon)$. Then

$$\delta_f > \sqrt{2}(\beta + \epsilon), \quad \forall f \in TH_N(B^2(0,1)),$$

where $TH_N(B^2(0,1)) : = \{f \in TH(B^2(0,1))$ such that $\mathcal{J}_f(0) = I\}$. Hence, $\forall f \in TH_N(B^2(0,1))$ there exists, c_f such that $c_f > \sqrt{2}(\beta + \epsilon)$ and f maps a subdomain M of $B^2(0,1)$ biholomorphically onto $B^2(w_f, c_f)$, a ball centered at w_f of radius c_f.

However, by Theorem 14, we know that $\forall \epsilon > 0$ there exists $g \in TH_N(D(0,1))$ such that $g(D(0,1))$ cannot contain a univalent $\mathbb{C}_2$-disc of radius $r > \beta + \epsilon$. Also $B^2(0, \frac{1}{\sqrt{2}}) \subseteq D(0,1)$, so $g \in TH_N(B^2(0, \frac{1}{\sqrt{2}}))$. Let us define:

$$g^*(w) = \sqrt{2}g(\frac{w}{\sqrt{2}}) \text{ on } B^2(0,1).$$

Then $g^*(w) \in TH_N(B^2(0,1))$. Hence, there exists c_{g^*} such that $c_{g^*} > \sqrt{2}(\beta+\epsilon)$ and g^* maps a subdomain W of $B^2(0,1)$ biholomorphically onto $B^2(w_{g^*}, c_{g^*})$.

Then, g maps the subdomain $\frac{W}{\sqrt{2}}$ biholomorphically onto $B^2(\frac{w_{g^*}}{\sqrt{2}}, \frac{c_{g^*}}{\sqrt{2}})$. But $D(\frac{w_{g^*}}{\sqrt{2}}, \frac{c_{g^*}}{\sqrt{2}}) \subseteq B^2(\frac{w_{g^*}}{\sqrt{2}}, \frac{c_{g^*}}{\sqrt{2}})$, so g maps, biholomorphically

$$g^{-1}(D(\frac{w_{g^*}}{\sqrt{2}}, \frac{c_{g^*}}{\sqrt{2}})) \text{ onto } D(\frac{w_{g^*}}{\sqrt{2}}, \frac{c_{g^*}}{\sqrt{2}}).$$

This contradicts the way in which g was chosen because $g^{-1}(D(\frac{w_{g^*}}{\sqrt{2}}, \frac{c_{g^*}}{\sqrt{2}})) \subseteq \frac{W}{\sqrt{2}} \subseteq D(0,1)$ with $\frac{c_{g^*}}{\sqrt{2}} > \beta + \epsilon$.□

2.4. $\mathbb{T}$-HOLOMORPHY AND QUASIREGULARITY

As we saw in the introduction, there is a Bloch theorem for K-quasiregular mappings. Then, to justify our Bloch theorem on the unit ball, we need to prove that the new class of mappings is not totally included in the class of K-quasiregular mappings.

First, with Definition 8 it is easy to show the following characterization:

Remark 1. *If $f \in TH(B^2)$ then f is K-quasiregular iff*

$$\left|\frac{\partial f_1}{\partial z_1}\right|^2 + \left|\frac{\partial f_2}{\partial z_1}\right|^2 \leq K^2 \left|\left(\frac{\partial f_1}{\partial z_1}\right)^2 + \left(\frac{\partial f_2}{\partial z_1}\right)^2\right| \text{ on } B^2.$$

The following examples will clearly show that a $\mathbb{T}$-holomorphic mapping is not necessarily quasiregular.

Example 1. *Let $f(w) = u \cdot w$, where u and w are in $\mathbb{C}_2$. If $u \not\equiv 0$ is noninvertible then, f is not quasiregular. If u is invertible, then f is K-quasiregular for*

$$K = \left(\frac{|u_1|^2 + |u_2|^2}{|u_1{}^2 + u_2{}^2|}\right)^{1/2},$$

where $u = u_1 + u_2 i_2$. In particular, if $u \in \mathbb{C}_1$, then f is conformal. However, we note in this example, that $f : \mathbb{C}^2 \longrightarrow \mathbb{C}^2$ is a linear transformation which is nondegenerate if and only if u is invertible.

In fact, any injective holomorphic mapping of the closed ball $\bar{B}^2$ into $\mathbb{C}^2$ is K-quasiregular for some K, but it is interesting to estimate K. In the last example, it is important to specify that f is clearly an entire $\mathbb{T}$-holomorphic mapping and that the multiplication is the multiplication between bicomplex numbers. The next examples will show that there exist some nontrivial mappings which are simultaneously quasiregular and $\mathbb{T}$-holomorphic with $\mathcal{J}_f(0) = I$.

Example 2. *If $f(w) = e^w$ then f is K-quasiregular on B^2 iff $K \geq \sqrt{cosh(2)}$. Moreover, because $(e^w)'|_{w=0} = e^0 = 1$, we have $\mathcal{J}_{e^w}(0) = I$.*

Proof. Let f_{e1} and f_{e2} be holomorphic functions on $\mathbb{C}_1$. Then we know that

$$f(z_1 + z_2 i_2) = f_{e1}(z_1 - z_2 i_1)e_1 + f_{e2}(z_1 + z_2 i_1)e_2$$

is $\mathbb{T}$-holomorphic with $f'(z_1 + z_2 i_2) = f'_{e1}(z_1 - z_2 i_1)e_1 + f'_{e2}(z_1 + z_2 i_1)e_2$. First, we seek conditions on f'_{e1} and f'_{e2} such that f is K-quasiregular. By Remark 1 it is easy to show that f is K-quasiregular on B^2 iff

$$|f'_{e1}(z_1 - z_2 i_1) + f'_{e2}(z_1 + z_2 i_1)|^2 + |f'_{e1}(z_1 - z_2 i_1) - f'_{e2}(z_1 + z_2 i_1)|^2 \leq$$

$$4K^2 |f'_{e1}(z_1 - z_2 i_1) f'_{e2}(z_1 + z_2 i_1)|$$

on B^2, that is

$$|f'_{e1}(z_1 - z_2 i_1)|^2 + |f'_{e2}(z_1 + z_2 i_1)|^2 \leq 2K^2 |f'_{e1}(z_1 - z_2 i_1) f'_{e2}(z_1 + z_2 i_1)|. \tag{2.4.2.4.1}$$

Now, $e^{z_1+z_2i_2} = e^{z_1-z_2i_1}e_1 + e^{z_1+z_2i_1}e_2$, so in this case $f'_{e1}(z_1 - z_2i_1) = e^{z_1-z_2i_1}$ and $f'_{e2}(z_1 - z_2i_1) = e^{z_1+z_2i_1}$. Hence (2.4.2.4.1) becomes

$$|e^{-2z_2i_1}| + |e^{2z_2i_1}| \leq 2K^2.$$

Let $z_2 = x + yi_1$. Then, because $z_1 + z_2i_2 \in B^2$, we have that $|y| < 1$. Moreover, $\sup_{\{|y|<1\}} \{e^{2y} + e^{-2y}\} = e^2 + e^{-2}$ and then $K \geq \sqrt{\frac{e^2+e^{-2}}{2}}$, i.e. $K \geq \sqrt{cosh(2)}$.□

In Example 2, because $f'(0) = 1$, we know by Theorem 12 that $\delta_f \geq d$. Since the mapping f is K-quasiregular, we already know that $\delta_f \geq c_K$ by Theorem 8. However, Theorems 8 and 12 merely assert the existence of the constants c_K and d without giving any estimates for these constants. From Theorem 15, on the other hand, we have an interesting lower estimate $\delta_f \geq \delta \geq \beta/\sqrt{2}$, by invoking lower estimates on the classical Bloch constant β [**8**]. One can also give lower estimates for the Bloch constant for K-quasiregular holomorphic mappings [**9**].

The next example is a mapping f for which $\delta_f \geq d$ by Theorem 12, but for which it is impossible to invoke Theorem 8.

Example 3. *If $f(w) = w + \frac{w^2}{2}$, then f is an entire $\mathbb{T}$-holomorphic (normalized) mapping, but for all K is not K-quasiregular.*

Proof. The function f is normalized because $f'(w) = 1+w$ and then $f'(0) = 1$. Also, $w_0 = -1/2+1/2j$ is in B^2 with $f'(w_0) = 1/2+1/2j$ which is noninvertible. Hence f cannot satisfy the criteria of Remark 1 at w_0 and then for all K, f cannot be K-quasiregular. Actually, quasiregular holomorphic mappings in $\mathbb{C}^2$ are necessarily locally injective hence locally quasiconformal, but we wished to avoid invoking this rather deep theorem (see [**12**]).□

Now, we show that our class of mappings includes some K-quasiregular mappings for arbitrary values of K. Then, Theorems 12 and 15 give geometric information about K-quasiregular mappings for a subclass including different values of K.

Example 4. *If* $f(z_1 + z_2 i_2) = \left(\frac{e^{n(z_1 - z_2 i_1)}}{n}\right) e_1 + (z_1 + z_2 i_1) e_2$, *then* f *is an entire* $\mathbb{T}$*-holomorphic (normalized) mapping which is* K*-quasiregular in* B^2 *with* K *becoming necessarily bigger as* n *increases.*

Proof. First, we see that f is normalized because $f'_{e1}(0) = e^{n0} = 1$ and $f'_{e2}(z_1 + z_2 i_1) = 1$, i.e. $f'(0) = 1e_1 + 1e_2 = 1$. Also, by (2.4.2.4.1) f is K-quasiregular on B^2 iff $|e^{n(z_1 - z_2 i_1)}|^2 + 1 \leq 2K^2 |e^{n(z_1 - z_2 i_1)}|$ on B^2. Then, we must have

$$K \geq \sqrt{\frac{\frac{1}{|e^{n(z_1 - z_2 i_1)}|} + |e^{n(z_1 - z_2 i_1)}|}{2}} \text{ on } B^2.$$

However, $\sup\limits_{\{w \in B^2\}} \sqrt{\frac{\frac{1}{|e^{n(z_1 - z_2 i_1)}|} + |e^{n(z_1 - z_2 i_1)}|}{2}} = \max\limits_{\{a^2 + d^2 \leq 1\}} \sqrt{\frac{\frac{1}{e^{n(a+d)}} + e^{n(a+d)}}{2}} = \sqrt{\frac{\frac{1}{e^{n(a'+d')}} + e^{n(a'+d')}}{2}}$, where $a' + d'$ can be taken to be positive. Finally we see that the last expression goes to infinity as $n \to \infty$.□

Finally, we give an example of a mapping which is $\mathbb{T}$-holomorphic and biholomorphic without being quasiregular on the unit ball of $\mathbb{C}^2$. In fact, we show that the class of $\mathbb{T}$-biholomorphic mappings cannot be totaly included in the class of quasiconformal mappings on the unit ball.

Example 5. *If* $f(z_1 + z_2 i_2) = \left(-\frac{(z_1 - z_2 i_1)^2}{2\sqrt{2}} + (z_1 - z_2 i_1)\right) e_1 + (z_1 + z_2 i_1) e_2$ *then* f *is an entire* $\mathbb{T}$*-holomorphic (normalized) mapping which is biholomorphic but not quasiregular on the unit ball of* $\mathbb{C}^2$.

Proof. Because f'_{ei} is nonzero on $h_i(D(0,\sqrt{2}))$ for $i = 1,2$ and $B^2(0,1) \subseteq D(0,\sqrt{2})$, then f is $\mathbb{T}$-biholomorphic on B^2. However, f cannot be K-quasiregular on B^2 because the relationship (2.4.2.4.1) will fail for all K as $z_1 - z_2 i_1 \to \sqrt{2}$ with $z_1 - z_2 i_1 \in B^2$.□

2.5. FINAL REMARKS

It is then interesting to ask whether the same is possible in the case of $\mathbb{T}$-holomorphic mappings in $\mathbb{C}^2$. However, here we can directly find a Picard theorem without invoking our Bloch theorem.

Theorem 16 (Picard). *Let $f \in TH(\mathbb{C}^2)$. If there are two bicomplex numbers α, β such that $\alpha - \beta$ is invertible and for which the set*

$$\{w \in \mathbb{C}_2 : \ w - \alpha \text{ is noninvertible}\} \cup \{w \in \mathbb{C}_2 : \ w - \beta \text{ is noninvertible}\}$$

is not in the range of f, then f is constant.

Proof. We have just to apply the so-called "little Picard theorem" [**27**] to f_{e1} and f_{e2}. The fact that $\alpha - \beta$ is invertible will insure us that $\alpha - \beta$ will be equal to a bicomplex number $s_1 e_1 + s_2 e_2$ with s_1 and s_2 nonzero. Let $\alpha = \alpha_1 e_1 + \alpha_2 e_2$, $\beta = \beta_1 e_1 + \beta_2 e_2$. Suppose f_{e1} takes the value α_1 at a_1. There exist $z_1, z_2 \in \mathbb{C}_1$ such that $z_1 - z_2 i_1 = a_1$. Thus,

$$f(z_1 + z_2 i_2) \in \{w \in \mathbb{C}_2 : \ w - \alpha \text{ is noninvertible}\}.$$

Contradiction. Hence, f_{e1} omits α_1. Similarly, f_{e2} omits α_2, f_{e1} omits β_1 and f_{e2} omits β_2. Since $\alpha_1 - \beta_1 = s_1 \neq 0$, α_1 and β_1 are distinct. Similarly $\alpha_2 \neq \beta_2$.□

In the same way, it is possible to find also a Casorati-Weierstrass theorem:

Theorem 17 (Casorati-Weierstrass). *Let $f \in TH(\mathbb{C}^2)$ with $f'(w)$ not identically noninvertible. Then, $f(\mathbb{C}^2)$ is dense in $\mathbb{C}^2$.*

Proof. The hypotheses imply that we can write $f(z_1+z_2i_2) = f_{e1}(z_1-z_2i_1)e_1+ f_{e2}(z_1+z_2i_1)e_2$ with $f_{e1}, f_{e2} \in H(\mathbb{C}_1)$ and nonconstant. Then we can apply the Casorati-Weierstrass theorem for $\mathbb{C}_1$ to f_{e1} and f_{e2} in order to prove that $f(\mathbb{C}_2)$ is dense in $\mathbb{C}_2$.□

A famous example of Fatou and Bieberbach (see [**20**]) shows that the usual formulation of the Picard theorem in $\mathbb{C}$ does not extend to holomorphic mappings in $\mathbb{C}^2$.

In this connection, we have some interesting consequences of Theorem 17 which can be interpreted as an other kind of little Picard theorem for bicomplex numbers:

Corollary 1. *There is no nondegenerate* $\mathbb{T}$*-holomorphic mapping*

$$f : \mathbb{C}^2 \longrightarrow \mathbb{C}^2$$

such that $\mathbb{C}^2 \setminus f(\mathbb{C}^2)$ *contains a ball.*

Corollary 2. *Fatou-Bieberbach examples cannot be* $\mathbb{T}$*-holomorphic mappings, i.e. they connot satisfy the complexified Cauchy-Riemann equations.*

For a beautiful formulation of Picard's theorem which holds in higher dimensions, see [**14**]. Also, for a version of Picard's theorem for quasiregular mappings see Rickman [**22**].

BIBLIOGRAPHIE

[1] L. V. Ahlfors, An extension of Schwarz's lemma, *J. Anal. Math.* **43**, 359-364 (1938).

[2] L. V. Ahlfors and H. Grunsky, Über die Blochsche Konstante, *Math. Z.* **42**, 671-673 (1937).

[3] S. Bedding and Briggs K., Iteration of quaternion maps, *Internat. J. Bifur. Chaos Appl. Sci. Engrg.* **5**, 877-881 (1995).

[4] A. Bloch, Les théorèmes de M.Valiron sur les fonctions entières et la théorie de l'uniformisation, *Ann. Fac. Sci. Univ. Toulouse(3)* **17**, 1-22 (1925).

[5] S. Bochner, Bloch's theorem for real variables, *Bull. Amer. Math. Soc.* **52**, 715-719 (1946).

[6] M. Bonk, On Bloch's constant, *Proc. Amer. Math. Soc.* **110**, 889-894 (1990).

[7] L. Carleson and T. W. Gamelin, *Complex Dynamics*, Springer-Verlag, New York, 1993.

[8] H. Chen and P. M. Gauthier, On Bloch's Constant, *J. Analyse* **69**, 275-291 (1996).

[9] H. Chen and P. M. Gauthier, Bloch constants in several variables, *Trans. Amer. Math. Soc.*, (to appear).

[10] A. Douady and J. H. Hubbard, Iteration des polynômes quadratiques complexes, *C.R. Acad. Sc. Paris* **294**, 123-126 (1982).

[11] B. Fauser, Clifford Algebraic Remark on the Mandelbrot Set of Two-Component Number Systems, *Advances in Applied Clifford Algebras* **6**, 1-26 (1996).

[12] P. M. Gauthier, Covering properties of holomorphic mappings, *Contemporary Mathematics* **222**, 211-218 (1999).

[13] J. Gomatam, J. Doyle, B. Steves and I. McFarlane, Generalization of the Mandelbrot set: Quaternionic Quadratic Maps, *Chaos, Solitons & Fractals* **5**, 971-985 (1995).

[14] M. L. Green, Holomorphic maps into complex projective space omiting hyperplanes, *Trans. Amer. Math. Soc.* **169**, 89-103 (1972).

[15] C. J. Griffin and G. C. Joshi, Associators in Generalized Octonionic Maps, *Chaos, Solitons & Fractals* **3**, 307-319 (1993).

[16] C. J. Griffin and G. C. Joshi, Transition Points in Octonionic Julia Sets, *Chaos, Soliton & Fractals* **3**, 67-88 (1993).

[17] R. Heidrich and G. Jank, On the Iteration of Quaternionic Moebius Transformations, *Complex Variables* **29**, 313-318 (1996).

[18] J. A. R. Holbrook, Quaternionic Fatou-Julia Sets, *Ann. sc. math. Québec* **11**, 79-94 (1987).

[19] I. L. Kantor, *Hypercomplex numbers*, Springer-Verlag, New York, 1989.

[20] W. Kaplan, Functions of Several Complex Variables, Ann Arbor Publishers, Michigan, 1964.

[21] L. Kaup and B. Kaup, Holomorphic Functions of Several Variables, Walter de Gruyter & Co., Berlin, 1983.

[22] I. Laine and S. Rickman, Value Distribution Theory, Lecture Notes in Mathematics no. 981, Springer-Verlag, 1983.

[23] A. Norton, Generation and Display of Geometric Fractals in 3-D, *Computer Graphics* **16**, 61-67 (1982).

[24] G. B. Price, *An Introduction to Multicomplex Spaces and Functions*, Marcel Dekker Inc., New York, 1991.

[25] D. Rochon, A Bloch Constant for Hyperholomorphic Functions, *Complex Variables*, (to appear).

[26] D. Rochon, *Sur une généralisation des nombres complexes: les tétranombres*, M. Sc. Université de Montréal, 1997.

[27] W. Rudin, Real and Complex Analysis, 3^e ed. McGraw-Hill, 1987.

[28] J. Ryan, Complexified Clifford Analysis, *Complex Variables* **1**, 119-149 (1982).

[29] K. Sakaguchi, On Bloch's theorem for several complex variables, *Sci. Rep. Tokyo Kyoiku Daigaku. Sect. A.* **5**, 149-154 (1956).

[30] J. L. Schiff, Normal families, Springer-Verlag, 1993.

[31] B. V. Shabat, Introduction to Complex Analysis part II: Functions of Several Variables, American Mathematical Society, 1992.

[32] S. Takahashi, Univalent mappings in several complex variables, *Ann. Math* **53**, 464-471 (1951).

[33] H. Wu, Normal families of holomorphic mappings, *Acta Math.* **119**, 193-233 (1967).

CONCLUSION

Dans le cas de notre théorème de Bloch sur la boule unité, il serait intéressant de tenter d'obtenir la véritable valeur de la constante. Il est à noter que nos estimations actuelles ne contredisent pas la possibilité que la valeur soit la même que la constante de Bloch d'une variable complexe.

Le dernier théorème dans l'article sur la dynamique bicomplexe est une bonne indication que le tétrabrot est non-connexe car les hypothèses sur l'ensemble de Mandelbrot peuvent être confirmées par ordinateur avec un haut degré de précision. Pour confirmer que la conjecture est vraie, nous avons deux choix, soit de démontrer théoriquement les hypothèses sur la géométrie de l'ensemble de Mandelbrot ou de prouver plus directement que le tétrabrot est non-connexe. Si la conjecture est prouvée comme étant vraie, une nouvelle question pourrait être de savoir la cardinalité de la famille des composantes connexes. Aussi, il pourrait être intéressant de savoir si les ensembles de "Julia-rempli" associés à des points sur des morceaux non-connexes du tétrabrot ont des propriétés spécifiques comme celles d'être non-connexes. Finalement, une autre question pertinente serait de savoir la dimension fractale locale de la frontière de tétrabrot.

REMERCIEMENTS

En tout premier lieu, je dois mes plus sincères remerciements à mon directeur de recherche et ami Paul Gauthier. Tout au long de mes dernières années de recherche, de ma 3^{ième} année universitaire jusqu'à maintenant, Paul a su créer un climat propice à l'élaboration de nouvelles idées. Ce climat de confiance et de compréhension m'a permis de mener à terme et de dépasser mes objectifs de départ. De plus, Paul a su m'introduire, par le biais de nombreux voyages et de précieux conseils, à la dynamique de la recherche mathémathique actuelle. En conséquence, mes idées ont pu prendre la forme formelle nécessaire à leur diffusion. Merci aussi pour les merveilleuses discussions d'ordre philosophique et spirituelle qui ont contribuées à rendre mes années de recherche des plus agréables.

Je voudrais aussi remercier mon père Alain et ma mère Lisette pour toute leur compréhension et leur aide durant mes années d'études. Merci pour votre support moral et votre affection sans limites.

Finalement, je voudrais remercier mon grand ami d'enfance Alec Bourque sans qui le "tétrabrot" serait encore une tâche plus ou moins floue sur mon ordinateur.

Annexe A

PROGRAMMES POUR ENGENDRER LES FIGURES

FIGURE 2 (Visual Basic 1024x768 24 bpp)

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 1 To (2 * (700))
      For x = 1 To 800
         For y = 1 To 720

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.16
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

            If ck >= 5 Then
               PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 6
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 6
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 3 (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 1 To (2 * (454))
      For x = 1 To 800
         For y = 1 To 720

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.16
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

            If z = (2 * 454) Then
               If ck >= 5 And ck < 7 Then col = RGB(0, 0, 128)
               If ck >= 7 And ck < 9 Then col = RGB(255, 120, 0)
               If ck >= 9 And ck < 11 Then col = RGB(255, 0, 0)
               If ck >= 11 And ck < 13 Then col = RGB(300, 600, 50)
               If ck >= 13 And ck < 17 Then col = RGB(150, 0, 300)
               If ck >= 17 And ck < 22 Then col = RGB(150, 200, 50)
               If ck >= 22 Then col = RGB(150, 0, 0)
            End If

            If ck >= 5 Then
               PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: v = 0: cd = 0
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 4 (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 1 To (2 * (454))
      For x = 1 To 800
         For y = 1 To 720

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.16
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

            If z = (2 * 454) Then
               If ck >= 7 And ck < 9 Then col = RGB(255, 120, 0)
               If ck >= 9 And ck < 11 Then col = RGB(255, 0, 0)
               If ck >= 11 And ck < 13 Then col = RGB(300, 600, 50)
               If ck >= 13 And ck < 17 Then col = RGB(150, 0, 300)
               If ck >= 17 And ck < 22 Then col = RGB(150, 200, 50)
               If ck >= 22 Then col = RGB(150, 0, 0)
            End If

            If ck >= 7 Then
               PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 5 (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 1 To (2 * (445))
    For x = 1 To 800
      For y = 1 To 720

        n = 300
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

        If z = (2 * 445) Then
          If ck >= 8 And ck < 9 Then col = RGB(255, 120, 0)
          If ck >= 9 And ck < 11 Then col = RGB(255, 0, 0)
          If ck >= 11 And ck < 13 Then col = RGB(300, 600, 50)
          If ck >= 13 And ck < 17 Then col = RGB(150, 0, 300)
          If ck >= 17 And ck < 22 Then col = RGB(150, 200, 50)
          If ck >= 22 Then col = RGB(150, 0, 0)
        End If

        If ck >= 8 Then
          PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
        End If

      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 130
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 6

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 1 To (2 * (445))
      For x = 1 To 800
         For y = 1 To 700

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

            If z = (2 * 445) Then

               If ck >= 10 And ck < 11 Then col = RGB(255, 0, 0)
               If ck >= 11 And ck < 13 Then col = RGB(300, 600, 50)
               If ck >= 13 And ck < 17 Then col = RGB(150, 0, 300)
               If ck >= 17 And ck < 22 Then col = RGB(150, 200, 50)
               If ck >= 22 Then col = RGB(150, 0, 0)
            End If

            If ck >= 10 Then
               PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command3_Click()
Unload Me
End Sub
```

'FIGURE 7 [Le Tétrabrot] (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 1 To (2 * (445))
    For x = 1 To 800
      For y = 1 To 700

        n = 300
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

        If z = (2 * 445) Then
          If ck >= 14 And ck < 17 Then col = RGB(300, 600, 0)
          If ck >= 17 And ck < 22 Then col = RGB(150, 0, 300)
          If ck >= 22 Then col = RGB(150, 0, 0)
        End If

        If ck >= 14 Then
          PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
        End If

      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 130
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 7 [Le tétrabrot en utilisant seulement l'algorithme de Mandelbrot] (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 1 To (2 *(445))
    For x = 1 To 800
      For y = 1 To 700

        n = 300
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

        If z = (2 *445) Then
          If ck >= 14 And ck < 17 Then col = RGB(300, 600, 0)
          If ck >= 17 And ck < 22 Then col = RGB(150, 0, 300)
          If ck >= 22 Then col = RGB(150, 0, 0)
        End If

        If ck >= 14 Then
          PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
        End If
      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zxx, zy, zyy, zx1, zy1, zc1, zd1, zc, zcc, zd, zdd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 130
  zx1 = mandel1(zx, zy) + cx + cd
  zy1 = mandel2(zx, zy) + cy - cc
  zc1 = mandel1(zc, zd) + cx - cd
  zd1 = mandel2(zc, zd) + cy + cc
  zx = zx1: zy = zy1: zc = zc1: zd = zd1
  zxx = (zx + zc) / 2: zyy = (zy + zd) / 2: zcc = (zd - zy) / 2: zdd = (zx - zc) / 2

  If ((zxx * zxx) + (zyy * zyy) + (zcc * zcc) + (zdd * zdd)) > 4 Then check = i: i = 130
  i = i + 1
Loop
End Function

Function mandel1(wx As Double, wy As Double) As Double
mandel1 = (wx * wx) - (wy * wy)
End Function

Function mandel2(wx As Double, wy As Double) As Double
mandel2 = 2 * wx * wy
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 8 [Le tétrabrot avec d=-0.3] (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 2 * 470 To (2 * (470))
      For x = 1 To 800
         For y = 1 To 700

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

            If z = (2 * 470) Then
               If ck >= 14 And ck < 17 Then col = RGB(300, 600, 0)
               If ck >= 17 And ck < 22 Then col = RGB(150, 0, 300)
               If ck >= 22 Then col = RGB(150, 0, 0)
            End If

            If ck >= 14 Then
               PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = -0.3
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 9 [Le tétrabrot avec d=-0.7] (Visual Basic)

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 2 * 470 To (2 * (470))
      For x = 1 To 800
         For y = 1 To 700

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

            If z = (2 * 470) Then
               If ck >= 14 And ck < 17 Then col = RGB(300, 600, 0)
               If ck >= 17 And ck < 22 Then col = RGB(150, 0, 300)
               If ck >= 22 Then col = RGB(150, 0, 0)
            End If

            If ck >= 14 Then
               PSet (x + Fix((z / 2) / 1.5) - 230, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = -0.7
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 10

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 1 To (2 * (800))
    For x = 1 To 800
      For y = 1 To 900

        n = 5500
        x1 = (x / n) - 1.817: y1 = (y / n) - 0.1: z1 = ((z / 2) / 5500) - 0.060363636
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB((z / 2) / 2, (z / 2) / 2, z)

        If ck >= 16 Then
          PSet (y + Fix((z / 2) / 8) - 210, x + Fix((z / 2) / 1.5) - 220), col
            Else
        End If

      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 50
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 50
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 11

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = (4 * 560) To (4 * 780)
      For x = 1 To 800
         For y = 2500 To 4000

            n = 32000
            x1 = (x / n) - 1.817: y1 = (y / n) - 0.1: z1 = (((z / 4) / 2) / 5500) - 0.060363636
            ck = check((x1) - 0.056, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB(((z - 4 * 550) / 2) / 2, ((z - 4 * 550) / 2) / 2, (z - 4 * 550))

            If ck >= 18 Then
               PSet (y + Fix(((z / 4) / 2) / 6) - 2800, x + Fix(((z / 4) / 2) / 0.2) - 1670), col

            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 70
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 70
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 12

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = (12 * 623) To (12 * 705)
    For x = 1 To 800
      For y = 9500 To 10500

        n = 100000
        x1 = (x / n) - 1.817: y1 = (y / n) - 0.1: z1 = (((z / 3 / 4) / 2) / 5500) - 0.060363636
        ck = check((x1) + 0.003, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB(((z - 4 * 3 * 623) / 2) / 2, ((z - 4 * 3 * 623) / 2) / 2, (z - 4 * 3 * 623))

        If ck >= 18 Then
          PSet (y + Fix(((z / 3 / 4) / 2)) - 9800, x + Fix(((z / 3 / 4) / 2) / 0.08) - 4150), col

        End If

      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 70
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 70
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 13 (Visual Basic 1280x1024, 24 bpp)

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 400 To 4000
    For x = 4400 To 5350
      For y = 5500 To 7500

        n = 5500
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.5: z1 = ((z / 2) / n) - 0.184
        ck = check(x1, y1, z1)


        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 100)
        Else col = RGB(((z) / 7) / 2, ((z) / 7) / 2, ((z) / 2))

        If ck >= 15 Then
          PSet (y + Fix((z / 6) / 4) - 5900, x + Fix((z / 6) / 0.4) - 5080), col

        End If

      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 30
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 30
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 14

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 230 To 1650
    For x = 1 To 800
      For y = 1 To 1400

        n = 20000
        x1 = (x / n) - 1.783: y1 = (y / n) - 0.075: z1 = (z / n) - 0.0417
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB((z / 2) / 3, (z / 2) / 3, z)

        If ck >= 16 Then
          PSet (y + Fix((z / 2) / 6) - 240, x + Fix((z / 2) / 0.8) - 520), col
        End If

      Next y
    Next x
  Next z
End Sub

Function color(number As Double) As Long
Dim r As Long, g As Long, b As Long
  'color=rgb(number/2,number/2,number)
  color = RGB((number / 2) / 2, (number / 2) / 2, number) 'bleu
  'color = RGB((number / 3), (number / 3), 0) 'jaune
End Function

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 50
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 50
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
 ScaleMode = vbPixels
 For x = 0 To 1600
     Line (Fix(x), y)-(Fix(x), y + 800), color(x)
 Next x
End Sub

Private Sub Command3_Click()
Unload Me
End Sub
```

FIGURE 15

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = (2 * (248)) To (2 * (430))
      For x = 1 To 1100
         For y = 1 To 700

            n = 250
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.356
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB(((z - (300)) / 2) / 2, ((z - (300)) / 2) / 2, (z - (300)))

            If ck >= 12 Then
               PSet (x + Fix((z / 2) / 1.5) - 240, y + Fix((z / 2) / 18) + 30), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1:  cx = -1.754878: cy = 0: cc = 0: cd = 0
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 16

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = (2 * (225)) To (2 * (478))
      For x = 1 To 950
         For y = 1 To 700

            n = 250
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.356
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB(((z - (2 * 150)) / 2) / 2, ((z - (2 * 150)) / 2) / 2, (z - (2 * 150)))

            If ck >= 7 Then
               PSet (x + Fix((z / 2) / 1.5) - 240, y + Fix((z / 2) / 18) + 30), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1: cx = -1.16: cy = 0.25: cc = 0: cd = 0
Do While i < 130
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 17

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = (2 * (70)) To (2 * (608))
      For x = 1 To 900
         For y = 1 To 700

            n = 250
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.356
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB(((z - (2 * 70)) / 2) / 2, ((z - (2 * 70)) / 2) / 2, (z - (2 * 70)))

            If ck >= 7 Then
               PSet (x + Fix((z / 2) / 1.5) - 240, y + Fix((z / 2) / 18) + 30), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1: cx = 0.25: cy = 0: cc = 0: cd = 0
Do While i < 50
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 50
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 18

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  n
  Dim col As Long
  For z = (2 * (70)) To (2 * (470))
    For x = 1 To 900
      For y = 1 To 700

        n = 250
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.356
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB(((z - (2 * 70)) / 2) / 2, ((z - (2 * 70)) / 2) / 2, (z - (2 * 70)))

        If z = (2 * 470) Then
          If ck >= 7 And ck < 8 Then col = RGB(150, 0, 300)
          If ck >= 8 And ck < 11 Then col = RGB(150, 200, 50)
          If ck >= 11 Then col = RGB(150, 0, 0)
        End If

        If ck >= 7 Then
          PSet (x + Fix((z / 2) / 1.5) - 240, y + Fix((z / 2) / 18) + 30), col
        End If

      Next y
    Next x
  Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1: cx = 0.25: cy = 0: cc = 0: cd = 0
Do While i < 130
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 20

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = (2 * (4)) To (2 * (687))
    For x = 1 To 1100
      For y = 1 To 720

        n = 300
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB(((z - (0)) / 2) / 2, ((z - (0)) / 2) / 2, (z - (0)))

        If ck >= 4 Then
          PSet (x + Fix((z / 2) / 1.5) - 350, y + Fix((z / 2) / 18) - 10), col
        End If

      Next y
    Next x
  Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1: cx = 0: cy = 1: cc = 0: cd = 0
Do While i < 10
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 10
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 21

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = (2 * (21)) To (2 * (670))
      For x = 1 To 1100
         For y = 1 To 720

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB(((z - (0)) / 2) / 2, ((z - (0)) / 2) / 2, (z - (0)))

            If ck >= 5 Then
               PSet (x + Fix((z / 2) / 1.5) - 350, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1:  cx = 0: cy = 1: cc = 0: cd = 0
Do While i < 10
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 10
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 22

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col As Long
   For z = 2 * (103) To (2 * (577))
      For x = 1 To 1100
         For y = 1 To 700

            n = 300
            x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
            ck = check(x1, y1, z1)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
            Else col = RGB(((z - (2 * 100)) / 2) / 2, ((z - (2 * 100)) / 2) / 2, (z - (2 * 100)))

            If ck >= 8 Then
               PSet (x + Fix((z / 2) / 1.5) - 350, y + Fix((z / 2) / 18) - 10), col
            End If

         Next y
      Next x
   Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1:  cx = 0: cy = 1: cc = 0: cd = 0
Do While i < 15
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 15
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 23

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col As Long
  For z = 2 * (115) To (2 * (565))
    For x = 1 To 1100
      For y = 1 To 700

        n = 300
        x1 = (x / n) - 2.1: y1 = (y / n) - 1.2: z1 = ((z / 2) / n) - 1.13
        ck = check(x1, y1, z1)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)
        Else col = RGB(((z - (2 * 105)) / 2) / 2, ((z - (2 * 105)) / 2) / 2, (z - (2 * 105)))

        If ck >= 15 Then
          PSet (x + Fix((z / 2) / 1.5) - 350, y + Fix((z / 2) / 18) - 10), col
        End If

      Next y
    Next x
  Next z
End Sub

Function check(zx As Double, zy As Double, zc As Double) As Integer
Dim zx1, zy1, zc1, zd1, zd, cx, cy, cc, cd, i
zd = 0: i = 1: cx = 0: cy = 1: cc = 0: cd = 0
Do While i < 130
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 130
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 24

```
DefDbl A-Z

Private Sub Command1_Click()
   ScaleMode = vbPixels
   Dim x, y, z, c, ck, x1, y1, z1, n
   Dim col, colb As Long
   For z = 100 To ((2 * 800) + 1100)
      For x = 1600 To 2400
         For y = 1 To 1400

            n = 5500
            x1 = (x / n) - 1.817: y1 = (y / n) - 0.1: z1 = (((z - 1100) / 2) / 5500) - 0.060363636
            ck = check(x1, y1, z1)
            If (z > 1200) Then col = RGB(((z - 1100) / 2) / 2, ((z - 1100) / 2) / 2, (z - 1100))
            Else col = RGB(25, 25, 100)

            If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)

            If ck >= 16 Then
               PSet (y + Fix(((z - 1100) / 2) / 8) - 210, x + Fix(((z - 1100) / 2) / 1.5) - 1900), col

            End If

         Next y
      Next x
   Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 100
   zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
   zy1 = 2 * zx * zy - 2 * zc * zd + cy
   zc1 = 2 * zx * zc - 2 * zy * zd + cc
   zd1 = 2 * zx * zd + 2 * zy * zc + cd
   zx = zx1: zy = zy1: zc = zc1: zd = zd1

   If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 100
   i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```

FIGURE 25

```
DefDbl A-Z

Private Sub Command1_Click()
  ScaleMode = vbPixels
  Dim x, y, z, c, ck, x1, y1, z1, n
  Dim col, colb As Long
  For z = 1450 To 2000
    For x = 2150 To 2500
      For y = 860 To 1450

        n = 5500
        x1 = (x / n) - 1.817: y1 = (y / n) - 0.1: z1 = (((z - 1100) / 2) / 5500) - 0.060363636
        ck = check(x1, y1, z1)
        If (z > 1200) Then col = RGB(((z - 1100) / 2) / 2, ((z - 1100) / 2) / 2, (z - 1100))
        Else col = RGB(25, 25, 100)

        If (x Mod 8 = 0) Or (x Mod 8 = 1) Or (x Mod 8 = 2) Then col = RGB(0, 0, 150)

        If ck >= 16 And ck < 17 Then colb = RGB(100, 0, 100)
        If ck >= 17 And ck < 19 Then colb = RGB(150, 150, 0)
        If ck >= 19 And ck < 23 Then colb = RGB(0, 150, 100)
        If ck >= 23 And ck < 30 Then colb = RGB(100, 50, 50)
        If ck >= 30 And ck < 40 Then colb = RGB(150, 100, 200)
        If ck >= 40 Then colb = RGB(200, 0, 0)

        If (z = 2000) Or (y = 860) Then col = colb

        If ck >= 16 Then
          PSet (y + Fix(((z - 1100) / 2) / 8) - 600, x + Fix(((z - 1100) / 2) / 1.5) - 2200), col

        End If

      Next y
    Next x
  Next z
End Sub

Function check(cx As Double, cy As Double, cc As Double) As Integer
Dim zx, zy, zx1, zy1, zc1, zd1, zc, zd, cd, i
zx = 0: zy = 0: zc = 0: zd = 0: i = 1: cd = 0
Do While i < 100
  zx1 = (zx * zx) - (zy * zy) + (zd * zd) - (zc * zc) + cx
  zy1 = 2 * zx * zy - 2 * zc * zd + cy
  zc1 = 2 * zx * zc - 2 * zy * zd + cc
  zd1 = 2 * zx * zd + 2 * zy * zc + cd
  zx = zx1: zy = zy1: zc = zc1: zd = zd1

  If ((zx * zx) + (zy * zy) + (zc * zc) + (zd * zd)) > 4 Then check = i: i = 100
  i = i + 1
Loop
End Function

Private Sub Command2_Click()
Unload Me
End Sub
```