

# 采用改进的牛顿迭代法的分形艺术设计<sup>①</sup>

田兴彦<sup>1,2</sup>, 邓基园<sup>2</sup>, 朱永娇<sup>3</sup>

<sup>1</sup>(琼州学院 电子信息工程学院, 三亚 572022)

<sup>2</sup>(湖南大学 软件学院, 长沙 410082)

<sup>3</sup>(长沙学院 计算机系, 长沙 410003)

**摘要:** 牛顿迭代法是为分形艺术图形实现提供素材的核心算法之一, 但直接采用牛顿迭代法生成分形艺术图形存在显示速度慢、生成图形种类少、难于控制等不足。提出了一种基于牛顿迭代法生成分形图形的改进算法 p-DQDA, 极大的提高了生成分形图的速度。同时还提出了在迭代过程中嵌入参数的构造牛顿广义迭代式的方法, 使得绘制分形艺术图形时通过简单的参数调节, 就能够生成种类繁多的具有对称美的分形艺术图形, 丰富了分形图形资源库。

**关键词:** 分形艺术; 牛顿迭代法; p-DQDA; 对称图形

## Design of Fractal Art Images with Improved Newton-Raphson Method

TIAN Xing-Yan<sup>1,2</sup>, DENG Ji-Yuan<sup>2</sup>, ZHU Yong-Jiao<sup>3</sup>

<sup>1</sup>(College of Electronics and Information Engineering, Qiongzhou University, Sanya 572022, China)

<sup>2</sup>(College of Software, Hunan University, Changsha 410082, China)

<sup>3</sup>(Department of Computer Science and Technology, University of Changsha, Changsha 410083, China)

**Abstract:** Newton-Raphson method is one of the core algorithms which provide materials in the process of the realization of fractal art images. But there are some deficiencies such as low displaying speed, little varied patterns and difficult controlling in resulting fractal images with Newton-Raphson method. Thus, based on Newton-Raphson method, a modified algorithm of generating fractal images — p-DQDA is proposed in this paper, greatly improving the speed of creating fractal images. And the general Newton iterative approach embedding parameters in the iterative process is also proposed, making it possible to generate a wide range of fractal art images with symmetric beauty by a simple parameter adjustment while drawing fractal images, thus enriching the resource library of fractal images.

**Key words:** fractal art; Newton-Raphson method; p-DQDA; symmetric images

牛顿迭代法<sup>[1-2]</sup>是分形理论中的一种重要方法。由牛顿迭代生成的分形图形具有精细结构、自相似等特点, 变幻无穷, 极具美感。随着计算机图形技术的发展, 人们对牛顿迭代法在复平面产生的混沌分形图像进行了广泛的研究<sup>[3-7]</sup>, 并在广告印刷、服装设计、纺织印染、陶瓷艺术等方面得到了成功的应用<sup>[8]</sup>。

考虑到基于牛顿迭代法生成分形图形速度慢且生成的图形种类少、难于控制等因素, 限制了其在艺术图形设计中的应用。因此本文提出了 P 分快速显示算法, 加快了牛顿法生成分形图形的速度。同时提出了

基于基本迭代函数  $F(z) = z^p - \lambda$  的迭代过程中嵌入参数构造牛顿广义迭代式的方法, 使得结合 p-DQDA 算法后, 通过简单的参数调节, 就能够生成种类繁多的具有对称性质的分形图形。

### 1 牛顿迭代法的基本原理

牛顿迭代法解方程原理<sup>[1-2]</sup>: 已知关于复数  $z$  的方程  $F(z) = 0$ ,  $F(z)$  连续可导且在解上  $F'(z) \neq 0$ , 对  $F(z)$  求导得:  $F'(z)$ , 首先猜测一个初始值  $z_0$ , 通过  $z_1 = z_0 - F(z_0) / F'(z_0)$ , 可以得到  $z_1$ ; 同样, 通过

① 基金项目:湖南省教育厅科技项目(09C124)

收稿时间:2010-02-19;收到修改稿时间:2010-03-18

$z_2 = z_1 - F(z_1)/F'(z_1)$ , 可以得到  $z_2$ ; ...; 如此反复迭代即有迭代式:

$$z_{n+1} = z_n - F(z_n)/F'(z_n), \quad (n=1,2,\dots) \quad (1)$$

一般选取  $F(z) = z^p - \lambda$ , 其中  $\{p \in \mathbb{Z} | p \geq 2\}$ ,  $\lambda = a + bi$  (默认  $\lambda = 1$ ) 为复常数。则迭代式可改写为:

$$z_{n+1} = ((p-1)z_n^p + \lambda) / pz_n^{p-1}, (n=1,2,\dots) \quad (2)$$

## 2 牛顿迭代法分形图形生成算法

根据迭代式(2)反复迭代, 得到复数序列  $\{z_k\}$ , 将会出现收敛和发散两种情况。设迭代精度  $e$ , 定义模  $\text{mod} = |z_{k+1} - z_k|$ , 在迭代过程中, 用  $\text{mod}$  来区分序列  $\{z_k\}$  的敛散性。若某点迭代到一定的次数仍然  $\text{mod} < e$ , 则序列  $\{z_k\}$  为收敛的, 此点为收敛点。若迭代到一定次数仍然成立, 则序列为发散的, 此点为发散点。为了标志不同收敛速度的点, 可根据迭代过程的收敛速度将收敛点用不同的颜色表示, 而对于发散的点则用同一种颜色表示。

假设显示器的分辨率是  $A \times B$  点, 可显示的颜色为  $K+1$  种, 分别  $0, 1, 2, \dots, K$  以表示, 则用牛顿迭代法生成分形图形的算法描述如下:

第一步: 选定迭代函数  $F(z) = z^p - \lambda$  ( $\lambda$  默认为 1)

第二步: 选定参数  $x_{\min} = y_{\min} = -1.5$ ,  $x_{\max} = y_{\max} = 1.5$ , 最大迭代步数  $MaxIter$  (如 1000), 迭代精度  $e$  (如  $10^{-4}$ ),  $dx = (x_{\max} - x_{\min}) / (A-1)$ ,  $dy = (y_{\max} - y_{\min}) / (B-1)$ ;

第三步: 令  $x_0 = x_{\min} + n_x * dx$ ,  $y_0 = y_{\min} + n_y * dy$ ,

$z_0 = x_0 + y_0 i$ , 其中  $n_x = 0, 1, \dots, A-1$ ,  $n_y = 0, 1, \dots, B-1$ ;

第四步: 对所有的点完成下面的循环

① 令  $z_0 = x_0 + y_0 i$ ,  $k = 0$ ;

② 按(2)式计算出  $z_{k+1}$ ;

③ 计算  $\text{mod} = |z_{k+1} - z_k|$ 。

如果  $\text{mod} < e$ , 则选择颜色  $K$  显示  $(n_x, n_y)$ , 跳至第五步;

如果  $\text{mod} > e$  且  $k = MaxIter$ , 则选择颜色显示  $(n_x, n_y)$ , 跳至第五步;

如果  $k < MaxIter$  且  $\text{mod} > e$ , 则  $k = k + 1$ , 跳至第②步。

第五步: 对点  $(n_x, n_y)$  显示颜色并转至下一点, 并且转至第三步。

这样就可以生成一幅彩色的分形图形, 图形上的任意一点的色彩和该点的迭代次数相对应。

## 3 P分快速显示算法(p-DQDA)

虽然牛顿迭代法的收敛速度比较快, 但由于其算法是逐点迭代逐点着色, 若屏幕显示分辨率  $300 \times 400$ , 每个屏幕点的平均迭代次数为  $M$ , 则此算法需要进行 120000 个迭代过程, 需要进行 120000M 次迭代, 尤其是当显示分辨率更高时, 迭代所需计算量则更大, 因此有必要研究图形的快速显示算法。

基于迭代函数  $F(z) = z^p - \lambda$  的牛顿迭代法生成的分形图形具有关于原点的对称性, 且存在很多连续区域。因此, 可设计改进算法, 方法如下:

Step1: 扩充屏幕: 把矩形屏幕扩充成由矩形的外接圆构成的圆形区域;

Step2: 把扩充后的圆形区域划分为均等的  $p$  个扇形子区域;

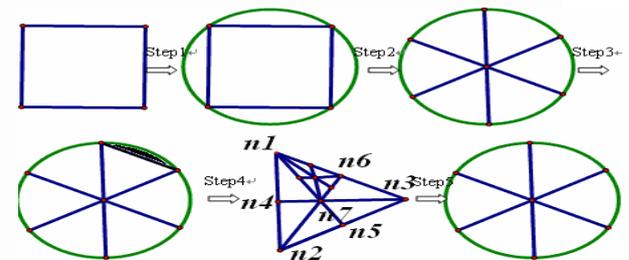


图1 p-DQDA 实现过程

Step3: 用牛顿迭代算法对黑色阴影部分的点迭代选择颜色着色;

Step4: 将扇形除阴影部分之外的三角形区域划分 6 个子区域, 然后再对六分后每个的三角形子区域做如下运算:

取子区域的顶点 1 和顶点 2 做初值迭代计算  $k_1$  和  $k_2$ ;

if ( $k_1 \neq k_2$ ) 继续六分该子区域;

else {

取子区域顶点 3 作初值迭代计算颜色  $k_3$ ;

if ( $k_1 \neq k_3$ ) 继续六分该子区域;

else {

取子区域顶点 4 作初值迭代计算颜色  $k_4$ ;

if ( $k_1 \neq k_4$ ) 继续六分该子区域;

else {

取子区域顶点 5 作初值迭代计算颜色  $k_5$ ;

if ( $k_1 \neq k_5$ ) 继续六分该子区域;

else {

```

取子区域顶点 6 作初值迭代计算颜色
k6;
if(k1 k6 !=)继续六分该子区域;
else{
    取子区域中心点 7 作初值迭代计算颜色
    k7;
    if(k1 != k7)继续六分该子区域;
    else 用颜色 k1 填充该子区域;
}
}
}
}

```

/\*六分该子区域直到不能再六分而只能逐点着色为止\*/

其中，子区域的顶点是指三角形的顶点和三角形三边的中点，子区域的中心是指三角形的内心。六分子区域是按照子区域的中心和子区域的顶点的连线划分该区域。

Step5: 把已着色的扇形子区域的颜色复制给剩余的  $p-1$  个扇形子区域;

Step6: 把矩形之外的圆的部分去除，还原屏幕。

把这个算法称之为  $P$  分快速显示算法，简称为  $p$ -DQDA( $p$ -division quick display algorithm)。此算法可分为  $p$  分屏幕和六分子区域两个步骤。由于牛顿迭代法生成分形图形算法中时间花费主要是在迭代上，因此改进算法关键是减少迭代初始点的个数。而  $p$ -DQDA 的两个步骤都能极大的减少初始迭代点的个数。

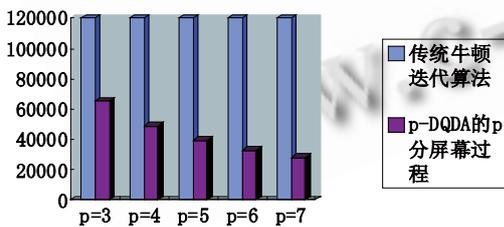


图 2 两种不同算法初始迭代点个数对比

若屏幕显示分辨率  $A \times B$ ，则在  $p$ -DQDA 的  $p$  分屏幕过程中，外接圆的半径为  $\sqrt{A^2 + B^2} / 2$ ，扇形子区域的面积为  $(A^2 + B^2)\pi / 4p$ ，迭代初始点的个数可以减少为  $(A^2 + B^2)\pi / 4p$ 。图 2 为屏幕显示分辨率为  $300 \times 400$  时传统牛顿迭代算法和  $p$ -DQDA 的  $p$  分屏幕过程中在  $p$  取不同值时初始迭代点个数的对比图。由图 2 中易见

$p$ -DQDA 的  $p$  分屏幕过程能大大的减少参与迭代过程的初始迭代点的个数，随着  $p$  值的增加，减少量越大。

对于  $p$ -DQDA 的六分子区域步骤，对于不同的图形，其初始迭代点的个数可减少为原来的  $1/2$  左右。因此，对于  $p$ -DQDA 算法，对于不同图形，其速度可提高约  $8ABp / (A^2 + B^2)\pi$  倍（当  $A=B$  时，为  $4p/\pi$  倍）。

表 1 是此算法与传统牛顿迭代法、快速显示算法<sup>[1]</sup>绘制分形图形运算时间的对比。运算时间对比是在相同的硬件条件下，采用相同的可视化技术，时间单位是毫秒，误差最大为 200 毫秒左右。

表 1 几种算法运算时间对比 (单位: ms)

阶数 算法名称	p=3	p=4	p=5	p=6	P=7
传统牛顿迭代法	11130	11352	11657	12064	12203
快速显示算法	6710	5177	5068	4825	4694
p-DQDA 算法	3113	2425	2031	1778	1569

从表 1 可以看出，两种改进算法对于传统牛顿迭代算法运算速度都有大幅度提高。对于传统牛顿迭代算法，绘制分形图的时间几乎不受阶数的影响，因为在传统牛顿迭代法中，总是对整个屏幕点进行迭代，迭代点的个数并未改变。对于快速显示算法，对不同的图形，其平均速度可提高 2-3 倍。而对于  $p$ -DQDA 算法，其平均速度可提高超过  $p$  倍，且阶数越大，改进效果越明显。



图 3  $F(z) = z^4 - 1$  时按式(2)迭代生成的分形图

由  $p$ -DQDA 的  $p$  分屏幕过程可知，此算法生成的分形图形具有  $p$  向对称性。图 3 是当  $F(z) = z^4 - 1$  时用  $p$ -DQDA 算法按式(2)生成的分形图形，具有明显的四

向对称性, 其生成速度提高了 5 倍左右。

#### 4 用 p-DQDA 绘制对称分形图形

基于迭代函数  $F(z) = z^p - \lambda$  绘制分形图形过程中, 可调节的参数十分有限, 绘制的分形图形种类比较少, 风格也比较单一。为了绘制更加丰富的分形图形, 只能通过引入广义迭代函数(如三角函数、指数函数等)。而 p-DQDA 算法是基于迭代函数  $F(z) = z^p - \lambda$  设计的, 算法的 p 分屏过程得依赖于迭代函数的次数 p。如果引入广义迭代函数, 就得提前给定一个固定的 p 值, 而且得重新设计算法。同时广义迭代函数十分复杂, 不便于算法的实现。为了避免这些问题, 我们有必要对基于基本迭代函数  $F(z) = z^p - \lambda$  的迭代式(2)进行拓展, 拓展方法如下所述。

根据迭代式(2), 利用复变函数基本理论

$$z_n^p = r_n^p (\cos p\theta_n + i \sin p\theta_n) \quad (3)$$

其中,  $r_n$  为复数  $z_n$  的模,  $\theta_n$  为复数  $z_n$  的幅角。计算时取辐角的主值, 即满足  $-\pi < \theta_n \leq \pi$ 。  $\theta_n$  可以由反正切  $\text{argtg}(y_n / x_n)$  的值确定, 相互关系为

$$\theta_n = \begin{cases} \text{arctg}(y_n / x_n) & x_n > 0 \\ 0 & x_n = 0, y_n = 0 \\ \pm\pi / 2 & x_n = 0, y_n \neq 0 \\ \text{arctg}(y_n / x_n) \pm \pi & x_n < 0, y_n \neq 0 \\ \pi & x_n < 0, y_n = 0 \end{cases} \quad (4)$$

其中  $-\pi / 2 < \text{argtg}(y_n / x_n) < \pi / 2$ , 将(3)式代入(2)式, 并按实部、虚部展开, 得到迭代形式为

$$\begin{cases} x_{n+1} = \frac{(p-1)r_n \cos \theta_n}{p} + \frac{r_n^{1-p} \cos(p-1)\theta_n}{p} \\ y_{n+1} = \frac{(p-1)r_n \sin \theta_n}{p} - \frac{r_n^{1-p} \sin(p-1)\theta_n}{p} \end{cases} \quad (5)$$

其中  $r_n = \sqrt{x_n^2 + y_n^2}$ , 复数  $z_n$  的主辐角  $\theta_n$  由(4)式求得。在迭代式(5)中嵌入若干个(一到四个)控制参数  $\alpha, \beta, \gamma, \delta$ , 若同时嵌入 4 个参数则有迭代式(6)

$$\begin{cases} x_{n+1} = \alpha \frac{(p-1)r_n \cos \theta_n}{p} + \beta \frac{r_n^{1-p} \cos(p-1)\theta_n}{p} \\ y_{n+1} = \gamma \frac{(p-1)r_n \sin \theta_n}{p} - \delta \frac{r_n^{1-p} \sin(p-1)\theta_n}{p} \end{cases} \quad (6)$$

其中  $\alpha, \beta, \gamma, \delta$  为调节参数, 当  $\alpha = \beta = \gamma = \delta = 1$  时, 式(6)简化为式(5), 即未嵌入参数时的情况。由于  $\alpha, \beta, \gamma, \delta$  任意一个参数的嵌入都改变了原来迭代函数(5)的收敛速度, 并且有可能使点的收敛性或发散性改变,

从而导致图形发生变化, 产生一些意想不到的图形, 而这些变化正是作为分形艺术图形设计所需要的。在嵌入参数的过程中我们可以选择嵌入参数的个数和位置(共 17 种选择), 这极大的增加了分形图形设计的选择空间和灵活性。同时每个参数调整都能改变分形图形, 每个参数的选择几乎可以为任意数(不能全取零), 这极大的丰富了牛顿迭代法分形图形资源库。图 4 为迭代函数  $F(z) = z^4 - 1$  按式(6)迭代嵌入参数生成的分形图。



图 4  $F(z) = z^4 - 1$  按式(6)迭代生成的分形图

#### 3 结语

本文在牛顿迭代法生成分形图形的基础上, 提出了一种改进牛顿迭代法生成分形图形的方法, 通过在迭代过程中嵌入调节参数, 使由简单的迭代函数便可生成富于变化的分形花型图案, 由设计者参与调整这种变化增加了设计的灵活性和易控制性。

针对本文思想开发了基于牛顿法分形图形生成软件, 通过参数的调整可以产生许多变化多样、造型奇特的具有对称美的分形艺术图案。生成的分形艺术图案再经过特效处理和着色算法等处理即可应用在广告印刷、服装设计、纺织印染、雕刻艺术等领域。

#### 参考文献

- 1 苏晓红, 李东, 胡铭曾. 用改进的 Newton-Raphson 方法生成对称的分形艺术图形. 计算机学报, 1999, 22(11): 1147-1151.
- 2 叶家鸣, 胡胜国. 采用牛顿迭代双侧逼近误差序列的分形艺术图形设计. 工程图学学报, 2009, 30(3): 146-153.
- 3 叶家鸣, 蒋永花. 基于牛顿迭代算法的分形艺术图形设计. 计算机技术与发展, 2006, 18(4): 86-91.

(下转第 188 页)

图2和图3分别显示的是 tburma14 和 bayg29 在运行过程求得的最优解与迭代次数的关系。

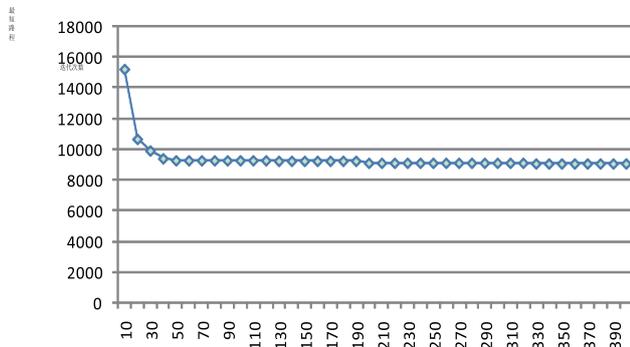


图3 运行 bayg29 实例收敛情况

从图中曲线趋势可以看出, 本文实现的算法迭代100次以内便接近问题的最优解, 由此可见利用 GALib 实现的遗传算法可以快速向最优解收敛。

## 5 结语

本文利用 GALib 实现了遗传算法, 并以解决 TSP 问题为例对相关技术进行说明, 实际开发过程表明, 基于 GALib 实现的遗传算法代码精简, 开发迅速, 可读性良好, 并且实验结果显示, 算法收敛快速, 运行结果准确。

## 参考文献

- Holland JH. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan press, 1975.
- Wall M. Galib.[2007-03]. <http://lancet.mit.edu/ga/dist/>.
- 王小平, 曹立明. *遗传算法-理论、应用与软件实现*. 西安: 西安交通大学出版社, 2002.
- Dantzig GB, Fulkerson DR, Johnson SM. Solution of a large scale traveling salesman problem. *Operation Research*, 1954, (2):393-410.
- <http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsp/>.

(上接第167页)

- Moonja J, Gi OK, Seong AK. Dynamics of Newton's method for solving some equations. *Computers & Graphics*, 2002, 26(2):271-279.
- Gilbert WJ. Generalizations of Newton's method. *Fractals*, 2001, 9(3):251-262.
- Wang XY, Liu W. The Julia set of Newton's method for

- multiple roots. *Applied Mathematics and Computation*, 2006, 171(1):101-110.
- 王兴元, 刘威. 三阶广义牛顿变换的 Julia 集. *工程图学学报*, 2005, 26(2):119-127.
- 胡多能, 王京, 张瑞秋. 分形图形与混沌图案的应用. *计算机工程与设计*, 2007, 28(4):893-895.