

利用平面点的压缩变换绘制随机分形树叶的 Matlab 算法

梁艺宝

(北京理工大学, 北京 102488)

摘要: 分形几何学由经典几何学产生, 借助计算机数据计算和直观绘图, 解决一些经典理论无法解决的问题, 在许多领域得到广泛应用。本文以 Matlab 的实现方式对分形图形学进行浅显阐释, 对分形几何的普及有一定意义。

关键词: 分形树叶; 平面点; 压缩变换; MATLAB

中图分类号: TP391.9

文献标识码: A

文章编号: 1007-9599 (2013) 01-0029-02

许多美丽的图形由密集的点组合而成, 本文中的随机分形树叶也是如此, 通过对起始点的坐标进行相应变换得出新坐标, 描绘出新点, 再对新点进行类似的变换得到更多的新点, 当这些点同时出现在一张平面坐标系图上时就构成了美丽的图形。本文引领读者体会递归和循环等算法设计的魅力和分形几何的趣味。

1 分析与解

设置一个初始点坐标 (x_0, y_0) , 在坐标纸上描绘出来后, 进行下列步骤:

步骤一: 产生一个 $[0, 1)$ 区间内的随机数 r , 根据随机数 r 所在区间, 对点 (x_0, y_0) 进行相应变换:

$x_{k+1} = ax_k + by_k + e$ $y_{k+1} = cx_k + dy_k + f$ (其中 $K=0, 1, 2, 3, \dots$)

得到的点记为 (x_1, y_1) , 在坐标纸上描绘出来。

a, b, c, d, e, f 的取值与 r 的区间对应关系如下:

r	a	b	c	d	e	f
$[0, 0.25)$	0.5	0	0	0.16	0	0.01
$[0.25, 0.5)$	0.85	0.06	-0.06	0.85	1.2	0.85
$[0.5, 0.75)$	0.22	-0.36	0.24	0.4	1.6	0.47
$[0.75, 1)$	-0.15	0.32	0.24	0	0.64	0.17

步骤二:

将新产生的点 (x_1, y_1) 作为新的起始点, 再重复进行步骤一, 得到新点 (x_2, y_2) , 并描绘出来……如此重复, 直至一共画出 n 个点, 其中 n 是任意设定的步骤总数。

1.1 递归算法

用递归算法绘图: 编写一个 Matlab 函数 drawtrees(x,y,n,m) 其中参数 x, y, n, m 依次表示起始点的横坐标、纵坐标、递归次数、步骤最大重复次数 (即绘出的点的个数)。由题意可知, 递归次数初始值是 1, 即 $n=1$ 。Matlab 代码如下:

```
%drawtrees.m
function drawtrees(x,y,n,m)
a1=0.5;b1=0;c1=0;d1=0.16;e1=0;f1=0.01;
a2=0.85;b2=0.06;c2=-0.06;d2=0.85;e2=1.2;f2=0.85;
a3=0.22;b3=-0.36;c3=0.24;d3=0.4;e3=1.6;f3=0.47;
a4=-0.15;b4=0.32;c4=0.24;d4=0;e4=0.64;f4=0.17;
r=rand; %产生随机数
if r>=0 && r<0.25 %根据随机数取值进行参数判定
a=a1;b=b1;c=c1;d=d1;e=e1;f=f1;
elseif r>=0.25 && r<0.5
a=a2;b=b2;c=c2;d=d2;e=e2;f=f2;
elseif r>=0.5 && r<0.75
```

```
a=a3;b=b3;c=c3;d=d3;e=e3;f=f3;
elseif r>=0.75 && r<1
a=a4;b=b4;c=c4;d=d4;e=e4;f=f4;
end
if n<=m
x1=a*x+b*y+e; %压缩变换公式
y1=c*x+d*y+f;
set(0,'RecursionLimit',10000); %扩大 plot 的循环次数限制
plot(x1,y1,'g');hold on;
drawtrees(x1,y1,n+1,m); %递归画图
end
```

1.2 图形展示

令 $n=1000, 3000$ 时, 则调用函数 drawtrees(0,0,1,1000) 和 drawtrees(0,0,1,3000) 分别见图 1、图 2;

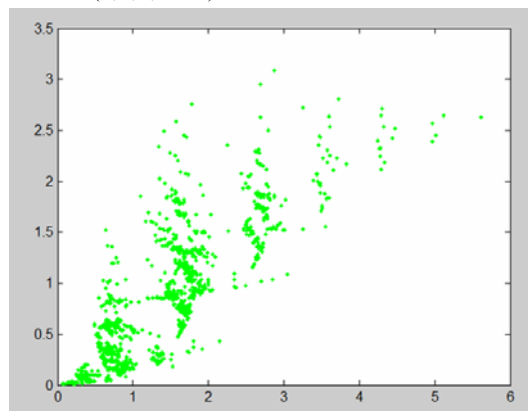


图1

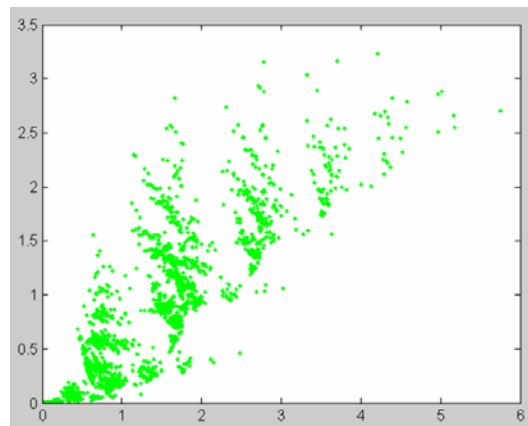


图2

当使 n 再增大时,则调用函数 $\text{drawtrees}(x,y,n,m)$ 绘图时间大大增加,且有可能导致程序崩溃。

2 算法 MATLAB 实现

2.1 循环算法

编写函数 $\text{drawtrees}(x,y,n)$,其中参数 x,y,n ,依次表示起始点的横坐标、纵坐标、步骤最大重复次数(即绘出的点的个数)。

Matlab 代码如下:

```
function drawtrees(x,y,n)
a1=0.5; b1=0; c1=0; d1=0.16; e1=0; f1=0.01;
a2=0.85; b2=0.06; c2=-0.06; d2=0.85; e2=1.2; f2=0.85;
a3=0.22; b3=-0.36; c3=0.24; d3=0.4; e3=1.6; f3=0.47;
a4=-0.15; b4=0.32; c4=0.24; d4=0; e4=0.64; f4=0.17;
for i=1:n
r=rand;
if r>=0 && r<0.25
a=a1; b=b1; c=c1; d=d1; e=e1; f=f1;
elseif r>=0.25 && r<0.5
a=a2; b=b2; c=c2; d=d2; e=e2; f=f2;
elseif r>=0.5 && r<=0.75
a=a3; b=b3; c=c3; d=d3; e=e3; f=f3;
elseif r>=0.75 && r<=1
a=a4; b=b4; c=c4; d=d4; e=e4; f=f4;
end
x=a*x+b*y+e;
y=c*x+d*y+f;
plot(x,y,'g','MarkerSize',10);
drawnow; %动态绘图,可以展示绘图的过程
end
```

2.2 图形展示

令 $n=50000$, 100000 时,则调用函数 $\text{drawtrees}(0,0,50000)$ 和 $\text{drawtrees}(0,0,100000)$,分别见图3、图4;

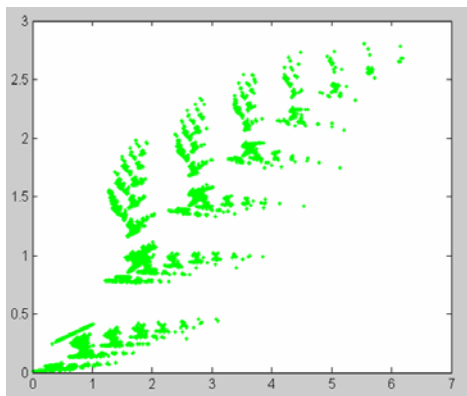


图 3

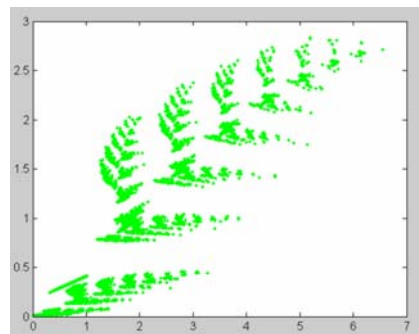


图4

令 $n=500000$, 2000000 时,则调用函数 $\text{drawtrees}(0,0,500000)$ 和 $\text{drawtrees}(0,0,2000000)$ 分别见图5、图6;

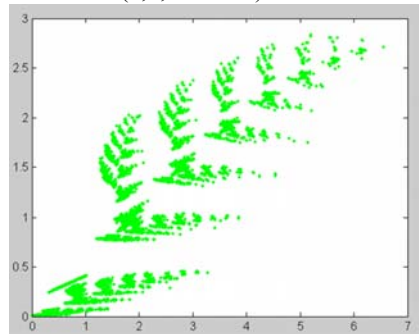


图5

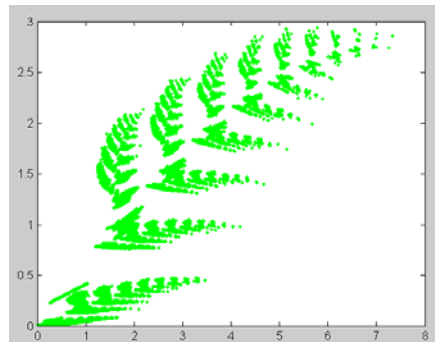


图6

3 结语

经过比对,递归算法绘图速度较慢,能绘制的点数少于循环算法,当 n 值无限增大时,循环算法绘图的时间是可观的。递归算法在某些方面是循环算法替代不了的,它的作用在其他的数据结构中体现尤为突出。本文只是个例,并不能证明循环绝对优于递归,旨在使解决问题的方法多样化。

参考文献:

[1]Stephen J.Chapman(著),郑碧波(译).MATLAB 编程.高等教育出版社.2010