سجادات كسورية جديدة وخوارزمياتها

خولة علي مصطفى ، امل عباس فلامرز

قسم الرياضيات ، كلية العلوم ، جامعة كركوك ، كركوك ، العراق

(تاريخ الاستلام: 2012/10/18 ---- تاريخ القبول: 2013/4/10)

الملخص

ان الهدف من هذا البحث هو توليد اشكال جديدة لسجادة سيربنسكي (Sierpinski carpet) حيث ان هذه الاشكال الجديدة لها ايضا ابعادا كسورية كما لسجادة سيربنسكي . في بداية البحث تم طرح فكرة الكسوريات وسجادة سيربنسكي ثم قمنا بتقديم الاشكال الجديدة الناتجة للتكرارات العشر الاولى مع اعطاء تسمية لكل منها و تقديم الخوارزميات الخاصة لعدد منها والمبرمجة بلغة C^{++} .

المقدمة

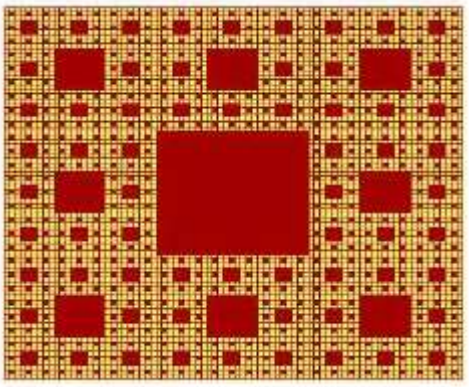
قدم الرياضي البولندي واكلو سيربنسكي (1882–1969) شكلا هندسيا جديدا اسماه سجادة سيربنسكي (Serpenski Carpet) في عام 1916 وكانت واحد من ابدع الاشكال في الكسوريات فقد حظيت باهنمام كبير في مجال التبولوجيا ولدراسة أوفى واشمل لسجادة سيربنسكي من الناحية التبولوجية يمكن الرجوع الى المصادر [7,8,9,10] .

تعتبر سجادة سيربنسكي (Serpenski Carpet) اساسا لكافة الاشكال الكسورية ذات بعد واحد في المستوي الديكارتي . هذا الشكل المثير، والذي يمكن أن يخفي فيه كل ما يمكن من الأشكال ذات البعد الواحد في المستوي بالمعنى التبولوجي . وهذا يعني انه ، في المستوي ، أي جسم ذو بعد واحد قد يظهر في سجادة سيربنسكي (ليس من الضروري تماما) ربما واحدة من الأشكال التي تعادلها تبولوجيا" .[1]

في التبولوجي ، يمكن النظر الى الأشكال وكانها مصنوعة من المطاط، والتي يمكن ضبط شكلها لتناسب ذلك [1و2] . على سبيل المثال ، الخط المستقيم ومنحنى كوخ (Koch Curve) متعادلان تبولوجيا وكلاهما تتواجدان في سجادة سيربنسكي .

الفكرة الاساسية لانشاء سجادة سيربنسكي كان الحصول على شكل واحد يمكنه ان يحوي جميع الاشكال الاخرى ذات البعد الواحد، وكانت نتيجة عمل سيربنسكي وبشكل دقيق : هو سجادة سيربنسكي المعروفة بانها شكل واحد ندمج فيه جميع الاشكال الممكنة في المستوي بعد ضغطها لتتناسب مع محتوى السجادة [1] .

ان السجادة تبدو كشكل مألوف ومنتظم ولكنها ابعد ما يكون عن ذلك [1] . فمثلا في سجادة سيربنسكي لاكننا نجد الاشكال المربعة فقط والتي هي مكافئة تبولوجيا للدائرة [7] . وبصورة عامة لبناء سجادة سيربنسكي نأخذ مربع كبداية ومن ثم نقسمه الى تسع مربعات متساوية ونحذف المربع الوسطي ومع كل مربع من المربعات الثمانية المتبقية نكرر نفس العمل وبعد عدد كافي من التكرارات (iterations) نحصل على سجادة سيربنسكي (الشكل رقم1) .



شكل رقم (1)

شكل رقم (2)

للوهلة الاولى تبدو سجادة سيربنسكي ككبوسة سيربنسكي (Serpenski Gasket) (الشكل رقم 2) في حين انهما مخلفتان تماماً عن بعض [1,2] . حيث انهما متشابهان في طريقة الانشاء لكن مختلفان في بعدهما الكسوري (Fractal Dimension) والذي يحسب:

$$n = \left(\frac{L}{S}\right)^{D} \Rightarrow \quad D = \frac{\ln(n)}{\ln(\frac{L}{S})}$$

حيث L يمثل طول القطعة و S عدد القطع فيكون البعد الكسوري لسجادة سيربنسكي D هو $D = \ln 8/\ln 3 \approx 1.89$ وان البعد الكسوري لكبوسة سيربنسكي هو $D = \ln 3/\ln 2 \approx 1.58$ [1،2] .

الغرض الرئيسي لهذا البحث هو عرض بعض السجادات الجديدة والتي تتولد بنفس الطريقة التي قام بها سيربنسكي عند انشاء سجادته،

مبتدئين بمربع ولكن بتقسيمات مختلفة عن سجادة سيربنسكي بالاضافة الى حساب البعد الكسوري لكل سجادة جديدة وكتابة الخوارزمية لكل منها بلغة C^{++}.

الاشكال الكسورية والخواص التبولوجية

لغرض تعريف تبولوجي مكافئ لسجادة سيربنسكي سنأخذ المجموعة المعدودة (Countable Set) المتكونة من اربعة عناصر $S=\{0,1,2,3\}$.

ونعرف عليها التبولوجي المفرق (Discrete Topology) ومن ثم ناخذ حاصل الضرب الديكارتي $S^N=\{0,1,2,3\}^N$ مع تبولوجي حاصل الضرب وليس بالضرورة ان تكون مفرق وتكون عناصر المجموعة S^N بشكل متتابعات لا نهائية من العناصر الاربعة أي إنه إذا كانت $x\in S^N$ فانها تكون بالشكل الاتي :

$x=x_0x_1x_2...x_n...$ ولذلك فان التبولوجي الذي $\ni\ x_i\in S$ يعرف سجادة سيربنسكي يكون مرصوص (compact) كون التبولوجي المعرف على S والتبولوجي على S^N هو حاضل ضرب ديكارتي لتبولوجي مرصوص وبما التبولوجي المعرف على S هو تبولوجي مفرق فانها فضاء غير متصل كليا(Totally Disconnected) لكن ليس بالضرورة ان تكون التبولوجي على S^N غير متصل كليا ، كما ان عدد التبولوجيات المغمورة في المستوي لسجادة سيربنسكي هو تبولوجي واحد وتحتوي عل مجموعات كثيفة للتبولوجي المكافي الى النقاط [9] ، وانه اذا كانت X مستمر الاتصال محليا (locally connected continuum) في المستوي فأنه لكل $\varepsilon>0$ يوجد عدد منتهي من المجالات المكملة من X بقطر اكبر من ε [9] ، ولانشاء سجادات سيربنسكي الجديدة يمكن استخدام طريقة مان التكرارية (Mannterative Procedure) والتي هي الطريقة الأمثل في التحليل الحسابي اللاخطي لحل المعادلات [2,4] ويمكننا تلخيص خطوات انشاء اي شكل كسوري بالخوارزمية ادناه :

1- البداية .
2- اختر المجموعة الابتدائية S .
3- اختر مجموعة من الدوال المقلصة $f_1, f_2, f_3,..., f_n$ لتكون نظام الدوال المكررة الخاصة بك .
4- اوجد تأثير نظام الدوال المكررة على المجموعة الابتدائية S .
5- اجعل الناتج يساوي S وعد الى الرقم 4 .
6-النهاية .

3 سجادات سيربنسكي الجديدة

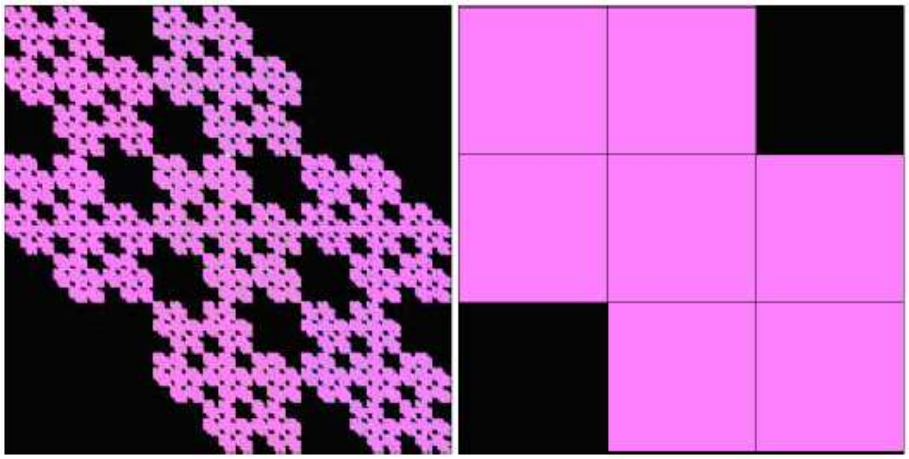
سننشأ نوعين من السجادات والتي هي جديدة وغير موجودة في المصادر. النوع الاول مشابه في جوهره لطريقة انشاء سجادة سيربنسكي بينما النوع الثاني تنشا تبعا لطريقة مان التكرارية (Mann iterative procedure) [3] وتتلخص طريقة مان التكرارية بالتالي : [2,4]

لتكن f دالة معرفة على الفضاء الخطي X ولتكن $x_0\in X$ ولتكن x_1 بالشكل $x_1=\lambda f(x_0)+(1-\lambda)x_0$ حيث $0<\lambda<1$ وعن طريق الاستقراء فان

$x_n=\lambda f(x_{n-1})+(1-\lambda)x_{n-1}$ حيث n=1,2,3,… .

سجادة رقم (1)

هذه السجادة هي من النوع الاول وطريقة انشاء هذه السجادة شبيه نوعا ما لطريقة انشاء سجادة سيربنسكي حيث نقوم بتقسيم المربع الاساس الى تسع مربعات متساوية ولكن عند الحذف قمنا بحذف اكثر من مربع ، حيث حذفنا المربع الذي يقع الى الاعلى من جهة اليمين والمربع الذي يقع الى الاسفل من جهة اليسار كما في الشكل (3 أ) و كررنا هذا العمل على المربعات السبع المتبقية ونستمر بهذ العمل للتكرارات العشر الاولى نحصل على السجادة كما في شكل رقم (3 ب) وان بعدها الكسوري هو $D=\frac{\ln 7}{\ln 3}\approx 1.77$

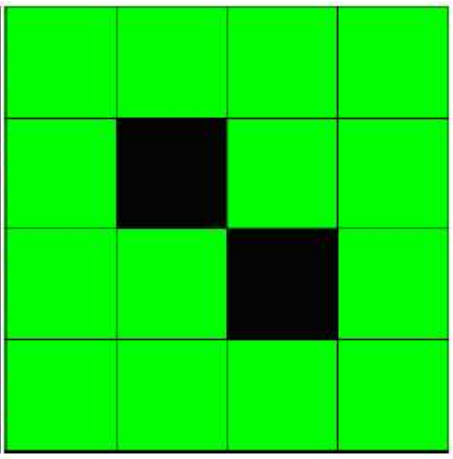


شكل رقم (3 أ)

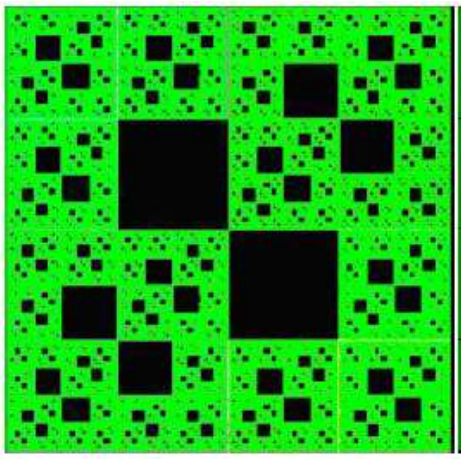
شكل رقم (3 ب)

سجادة رقم (2)

لأنشاء هذه السجادة والتي تعتبر من النوع الاول ايضا نقوم بتقسيم المربع الاساس الى ست عشر مربعا متساويا ومن ثم نقوم بحذف المربعين باللون الاسود كما في الشكل (4 أ) وكررنا العملية لعدد من المرات على المربعات الاربع عشر المتبقية ونستمر بهذ العمل نحصل على السجادة كما في شكل رقم (4 ب) ، وان بعدها الكسوري هو $D=\frac{\ln 14}{\ln 4}\approx 1.9$



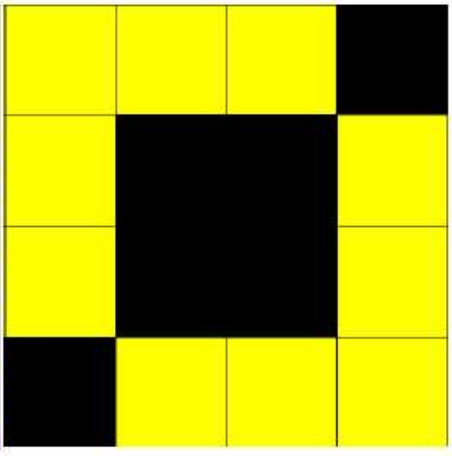
شكل رقم (4 أ)



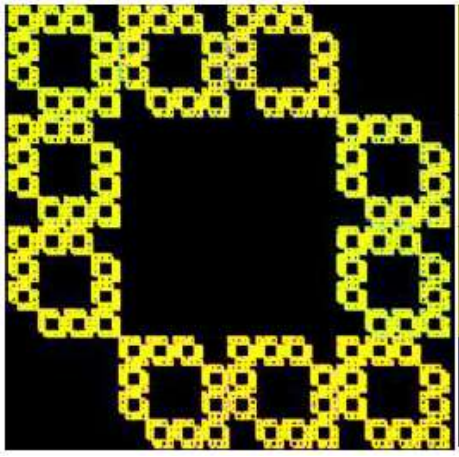
شكل رقم (4 ب)

السجادة رقم (3)

هذه السجادة من النوع الثاني والمنشأة باستخدام طريقة مان التكرارية حيث قسمنا المربع الاساس الى ثلاث عشر مربعا بحيث يقع المربع الاكبر في الوسط والذي يكون مساحته $\frac{1}{4}$ مساحة المربع الاساس والمربعات الاثني عشر الباقية لكل منها مساحة تعادل $\frac{1}{16}$ مساحة المربع الاساس . ثم قمنا بأزالة كل من المربع الكبير في الوسط والمربعين في اعلى اليمن والذي في اسفل اليسار كما في الشكل (5 أ) وبتكرار العملية لعدة مرات نحصل على السجادة كما في الشكل (5 ب) حيث البعد الكسوري لها $D=\frac{\ln 10}{\ln 4}\approx 1.66$



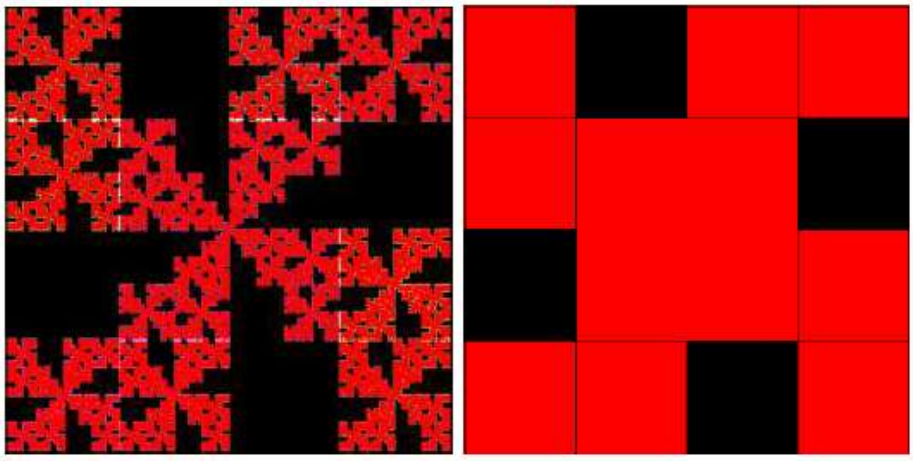
شكل رقم (5 أ)



شكل رقم (5 ب)

السجادة رقم (4)

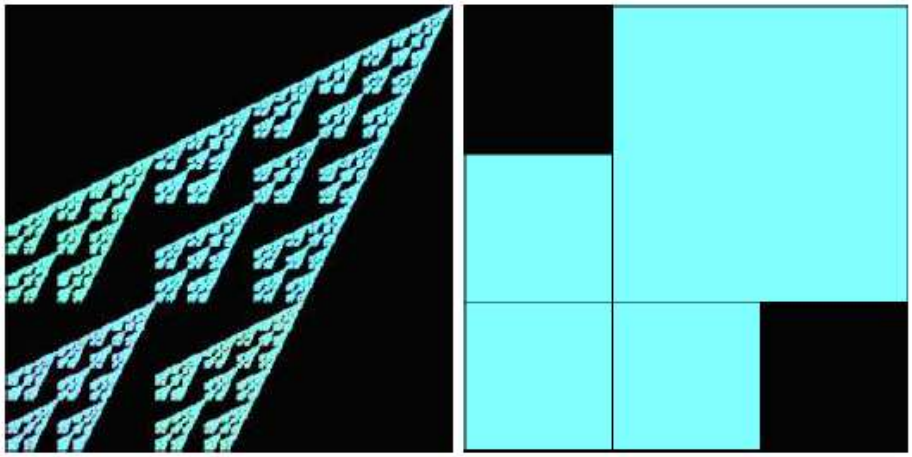
هذه السجادة من النوع الثاني ايضا حيث قسمنا المربع الاساس الى ثلاث عشر مربعا بحيث يقع المربع الاكبر في الوسط والذي يكون مساحته $\mathbf{\frac{1}{4}}$ مساحة المربع الاساس والمربعات الاثني عشر الباقية لكل منها مساحة تعادل $\mathbf{\frac{1}{16}}$ مساحة المربع الاساس . ثم قمنا بأزالة كل من المربعات ذات اللون الاسود كما في الشكل (6 أ) وبتكرار العملية لعدة مرات نحصل على السجادة كما في الشكل (6 ب) حيث البعد الكسوري لها $\mathbf{D=\frac{\ln 12}{\ln 4}\approx 1.79}$



شكل رقم (6 أ) شكل رقم (6 ب)

السجادة رقم (5)

كذلك هذه السجادة من النوع الثاني حيث قسمنا المربع الاساس الى ست مربعات بحيث يقع المربع الاكبر الى الاعلى من جهة اليمين والذي يكون مساحته $\frac{4}{9}$ مساحة المربع الاساس والمربعات الخمسة الباقية لكل منها مساحة تعادل $\frac{1}{9}$ مساحة المربع الاساس . ثم قمنا بأزالة كل من المربعات ذات اللون الاسود كما في الشكل (7 أ) وبتكرار العملية لعدة مرات نحصل على السجادة كما في الشكل (7 ب) حيث البعد الكسوري لها $D = \frac{\ln 7}{\ln 3} \approx 1.77$



شكل رقم (7 أ) شكل رقم (7 ب)

Algorithm 1
This is the algorithm to generate the number 1 carpet.
*x*1 = 50
*y*1 = 50
*x*2 = 450
*y*2 = 450
// Main module
// draw base of carpet
Plotrectangle (*x*1, *y*1, *x*2, *y*2)
// fill color in base
Setcolor (LIGHTPURPLE)
fillrectangle (*x*1, *y*1, *x*2, *y*2)
// draw carpet
setcolor(BLACK)
rect(*x*1, *y*1, *x*2, *y*2)
// function to draw carpet by dividing whole square into nine equal parts and
// dropping left bottom and right top squares and so on .
Rect (int *x*1, int *y*1, int *x*2, int *y*2)
Begin
if (*x*2 – *x*1 > 10) and (*y*2 – *y*1 > 10) then
*x*3 = *x*1 + (*x*2 – *x*1)/3
*y*3 = *y*1 + (*y*2 – *y*1)/3
*x*4 = *x*1 + 2*(*x*2 – *x*1)/3
*y*4 = *y*1 + 2*(*y*2 – *y*1)/3
// drop left bottom square
Plotrectangle (*x*1, *y*4, *x*3, *y*2)
Fillrectangle (*x*1, *y*4, *x*3, *y*2)
// drop right top square
Plotrectangle (*x*4, *y*1, *x*2, *y*3)
Fillrectangle (*x*4, *y*1, *x*2, *y*3)
// left top square

a = *x*1, *b* = *y*1, *c* = *x*3, *d* = *y*3
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// left square
a = *x*1, *b* = *y*3, *c* = *x*3, *d* = *y*4
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// bottom square
a = *x*3, *b* = *y*4, *c* = *x*4, *d* = *y*2
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// right bottom square
a = *x*4, *b* = *y*4, *c* = *x*2, *d* = *y*2
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// right square
a = *x*4, *b* = *y*3, *c* = *x*2, *d* = *y*4
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// top square
a = *x*3, *b* = *y*1, *c* = *x*4, *d* = *y*3
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// middle square
a = *x*3, *b* = *y*3, *c* = *x*4, *d* = *y*4
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
End of if
End

Algorithm 2

This is the algorithm to generate the number 3 carpet

*x*1 = 50
*y*1 = 50
*x*2 = 450
*y*2 = 450
// Main module
// draw base of carpet
plotrectangle (*x*1, *y*1, *x*2, *y*2)
// fill color in base
Setcolor (LIGHTYELLOW)
Fillrectangle (*x*1, *y*1, *x*2, *y*2)
// draw carpet
Setcolor (BLACK)
Rect (*x*1, *y*1, *x*2, *y*2)
// function to draw carpet by dividing whole square into 12 equal and one large
// parts and dropping left bottom, right top and large squares and so on.
rect(int *x*1, int *y*1, int *x*2, int *y*2)
Begin
if (*x*2 – *x*1 > 10) and (*y*2 – *y*1 > 10) then
*x*3 = *x*1 + (*x*2 – *x*1)/4
*y*3 = *y*1 + (*y*2 – *y*1)/4
*x*4 = *x*1 + 2*(*x*2 – *x*1)/4
*y*4 = *y*1 + 2*(*y*2 – *y*1)/4
*x*5 = *x*1 + 3*(*x*2 – *x*1)/4
*y*5 = *y*1 + 3*(*y*2 – *y*1)/4
// drop left bottom square
Plotrectangle (*x*1, *y*5, *x*3, *y*2)
Fillrectangle (*x*1, *y*5, *x*3, *y*2)
// drop right top square
Plotrectangle (*x*5, *y*1, *x*2, *y*3)
Fillrectangle (*x*5, *y*1, *x*2, *y*3)
//drop large one which is in the center
Plotrectangle (*x*3, *y*3, *x*5, *y*5)
Fillrectangle (*x*3, *y*3, *x*5, *y*5)
// left top square
a = *x*1, *b* = *y*1, *c* = *x*3, *d* = *y*3
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
//right left top square
a = *x*3, *b* = *y*1, *c* = *x*4, *d* = *y*3
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// right bottom square
a = *x*5, *b* = *y*5, *c* = *x*2, *d* = *y*2
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
//left right top square
a = *x*5, *b* = *y*1, *c* = *x*2, *d* = *y*3
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// left right bottom square
a = *x*4, *b* = *y*5, *c* = *x*5, *d* = *y*2
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// right left bottom square
a = *x*3, *b* = *y*5, *c* = *x*4, *d* = *y*2
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// below left top square
a = *x*1, *b* = *y*3, *c* = *x*3, *d* = *y*4
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// above left bottom square
a = *x*1, *b* = *y*4, *c* = *x*3, *d* = *y*5
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// below right top square
a = *x*5, *b* = *y*3, *c* = *x*2, *d* = *y*4
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// above right bottom square
a = *x*5, *b* = *y*4, *c* = *x*2, *d* = *y*5
rect(*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
End of if
End

Algorithm 3

This is the algorithm to generate the number 5 carpet .

*x*1 = 50
*y*1 = 50
*x*2 = 450
*y*2 = 450
// Main module
// draw base of carpet
Plotrectangle (*x*1, *y*1, *x*2, *y*2)
// fill color in base
Setcolor (LIGHTBLUE)

fillrectangle (*x*1, *y*1, *x*2, *y*2)
// draw carpet
Setcolor (BLACK)
Rect (*x*1, *y*1, *x*2, *y*2)
// function to draw carpet by dividing whole square into five equal and
// one large on the right top parts dropping left top and right bottom squares and
// so on .
Rect (int *x*1, int *y*1, int *x*2, int *y*2)
Begin
if (*x*2 – *x*1 > 10) and (*y*2 – *y*1 > 10) then
*x*3 = *x*1 + (*x*2 – *x*1)/3
*y*3 = *y*1 + (*y*2 – *y*1)/3
*x*4 = *x*1 + 2*(*x*2 – *x*1)/3
*y*4 = *y*1 + 2*(*y*2 – *y*1)/3
// drop left top square
Plotrectangle (*x*1, *y*1, *x*3, *y*3)
Fillrectangle (*x*1, *y*1, *x*3, *y*3)
// drop right bottom square
Plotrectangle (*x*4, *y*4, *x*2, *y*2)
Fillrectangle (*x*4, *y*4, *x*2, *y*2)
// left bottom square
a = *x*1, *b* = *y*4, *c* = *x*3, *d* = *y*2
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// left square
a = *x*1, *b* = *y*3, *c* = *x*3, *d* = *y*4
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// bottom square
a = *x*3, *b* = *y*4, *c* = *x*4, *d* = *y*2
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
// large square
a = *x*3, *b* = *y*1, *c* = *x*2, *d* = *y*4
rect (*a*, *b*, *c*, *d*)
plotrectangle (*a*, *b*, *c*, *d*)
End of if
End

الخوارزمية لكل سجادة كسورية والتي قدمت في هذا البحث تكون مختلفة عن الاخرى .ومع ذلك، فهي تسير في نفس الاتجاه حيث الخوارزمية رقم (1) تخص السجادات المنشأة بنفس طريقة سيربنسكي والخوارزميات رقم 2 و 3 تخص السجادات المنشأة حسب طريقة مان التكرارية .

الاستنتاجات

ان السجادات ذوات الاشكال الكسورية المختلفة هي ليست مجرد رسومات جميلة منشأة باستخدام الحاسوب لكن لها تطبيقات حقيقية في صناعة السجاد ومن السهل رؤية ان قياس المنطقة المتروكة من الحذف لكل سجادة كسورية هي مساوية الى الصفر . ويجدر بنا ملاحظة ان بعد السجادة الكسورية يزداد عندما ينقص معامل التصغير λ ، اي انه كلما زاد عدد المربعات المحذوفة فان البعد الكسوري يقل .

المصادر

[1] H. Peitgen, H. Jürgens, and D. Saupe, *Fractals for the Classroom, Part One: Introduction to Fractals and Chaos*. New York : Springer-Verlag, 1992 .
[2] H. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals*. New York: Springer-Verlag, 1992 .
[3] W.R. Mann, "Mean Value Methods in Iteration", *Proc. Amer. Math. Soc*., **4** (1953), pp. 506–510 .
[4] B.E. Rhoades, "Comments on Two Fixed Point Iteration Methods", *J. Math. Anal. Appl.,* **56** (1976), pp. 741–750 .
[5] Z. Huang, "A New Convergence Result for Fixed Point Iterations in Bounded Interval of *Rn*", *Computers Math. Applic.*,**34**(**12**) (1997), pp. 33–36 .
[6] Eberhard Zeidler, *Nonlinear Functional Analysis and Its Applications Part I.* New York: Springer-Verlag, 1985 .
[7] M.Eugenia Montiel, Alberto S. Aguado and Ed Zaluska, Topology in Fractals,S .
[8]M.F.Barnsley, Fractals Every Where, second ed. ,Morgan Kaufmann, 1993.
[9] Benjamin Vejnar, A Topological Characterization of The Sierpinski Triangle, Topology and Its Applications 159(2012) 1404-1408 .

new fractal carpets and their algorithms

khawlah Ali Mustafa , amal Abass Flamers
Dep Math , College of Science , Kirkuk university , Kirkuk , Iraq

(Received: 18/10/ 2012 ---- Accepted: 10/4/ 2013)

Abstract

The aim of this paper is to generate a new shapes of Serpenski Carpet where this new shapes have fractal dimensions just like Serpenski Carpet . At the beginning of our paper the idea of fractals and Serpenski Carpet are introduced and then we made the new shapes for the first 10 iterations with given a name for each one of them and then provide algorithms for a number of them which programmed in C + + .