Índex:

Introducció, estructura i metodologia del treball

El tema del treball, els fractals, és un tema complexe i sobretot molt extens. Quan vaig decidir fer el treball sobre els fractals encara no sabia com d'ampli era el món on em posava.
Vaig escollir aquest tema perquè em va atraure que la geometria fractal hagués estat inventada com a mitjà per a mesurar i entendre algunes estructures naturals tant aleatòries i diverses.
Alhora, vaig adonar-me de la poca divulgació que tenen i de la poca gent que n'ha sentit parlar fora dels cercles especialitzats en matemàtiques.
Els fractals, sobretot els fractals al pla complex, tenen una vesant estètica molt espectacular que ha portat a la aparició de multitud de webs on els artistes aficionats a la creació de gràfics fractals i el seu retoc mitjançant recursos informàtics hi exposen les seves obres.
El tema em va apassionar des del primer moment però degut a la complexitat de les matemàtiques que els fractals comporten, i la gran extensió del tema, vaig haver d'acotar la meva recerca i va ser difícil trobar una pràctica adequada a les meves possibilitats.
Aquest problema es va resoldre un cop vaig descobrir el programa NetLogo i vaig començar a experimentar i a formular instruccions amb ell.
Ja que el treball incloïa un apartat dedicat a la dimensió fractal, em va semblar interessant calcular la dimensió d'algun fractal natural ja que són els que ens toquen de més aprop.
Prenent com a exemple el treball de Mandelbrot de la costa de Gran Bretanya he mesurat la dimensió fractal de la costa de l'illa de Mallorca.
Els fractals són, després de tot, un tema apassionant i una aplicació de les matemàtiques més complexes enfocada cap a la natura i cap a les formes a les que estem avesats.

Va ser sorprenent veure com des d'unes matemàtiques cada cop més enrevessades i complexes, de sobte, pots treure el cap i adonar-te de les impressionants estructures de gran estètica que es poden crear, de la proximitat que hi ha entre aquestes, el nostre món natural i aquelles complexes matemàtiques de les que hem partit.

El treball es presenta encadenat en quatre grans blocs independents entre ells però alhora relacionats ja que cada bloc necessita de conceptes introduïts anteriorment. Tot plegat ens permetrà apreciar millor la pràctica final.

El primer bloc està destinat a la divulgació del concepte de fractal en els seus trets més generals i bàsics, la seva història i context.

El segon està dedicat a donar a conèixer alguns del fractals geomètrics més comuns i importants, i en particular a la explicació del concepte de la dimensió fractal. Conté una part teòrica on es defineix la dimensió fractal i s'explica la fórmula per a calcular-la, i una altra on s'aplica la fórmula per a calcular la dimensió dels els fractals geomètrics ja presentats. Finalment, s'inclou el càlcul de la dimensió de la línia de la costa de l'illa de Mallorca.

El tercer apartat d'aquest treball és el més dens i engloba pràcticament tota la part matemàtica més avançada. Està dedicat a l'explicació dels fractals construïts al "pla complex". Aquest tema exigeix àmplies explicacions de conceptes matemàtics i acaba tancant-se amb l'exposició de les creacions de dos dels grans matemàtics d'aquest camp. Aquest capítol potser és el més independent de tots, però em va semblar necessari divulgar els fractals al pla complex per arribar a comprendre la complexitat i l'amplitud del concepte fractal.

En el quart bloc s'exposen les pràctiques fetes amb el programa informàtic NetLogo amb una breu introducció al mètode seguit (sistema-L) i al llenguatge utilitzat (llenguatge Logo). S'inclouen les pràctiques de reproducció de quatre fractals geomètrics, ja exposats anteriorment, fetes amb el programa NetLogo i dos pràctiques que intenten reproduir dues estructures naturals: l'estructura d'un arbre i la d'una falguera.

El primer pas que vaig fer al començar la recerca, va ser buscar informació sobre el tema per tal introduir-me en el món dels fractals i poder entendre el que venia després. Posteriorment vaig acotar la recerca i vaig buscar i trobar una part pràctica al meu abast. Un cop donat un enfoc al treball i triades les pràctiques a fer, vaig estructurar-lo i vaig decidir les parts, que han anat variant els seus continguts en funció de les noves necessitats que han anat sorgint.

Després de la realització de les pràctiques, en el procés de redactat del treball, vaig trobar-me davant la dificultat, ja intuïda, de la complexitat de l'explicació, i la necessitat per altre banda d'exposar els conceptes clau amb prou claredat per tal de poder exposar seguidament altres conceptes més complexos. Aquesta ha estat la part difícil del treball, que espero haver superat.

Per fer arribar millor el treball, he decidit crear una pàgina web per poder exposar-hi tota la part interactiva de les meves pràctiques.

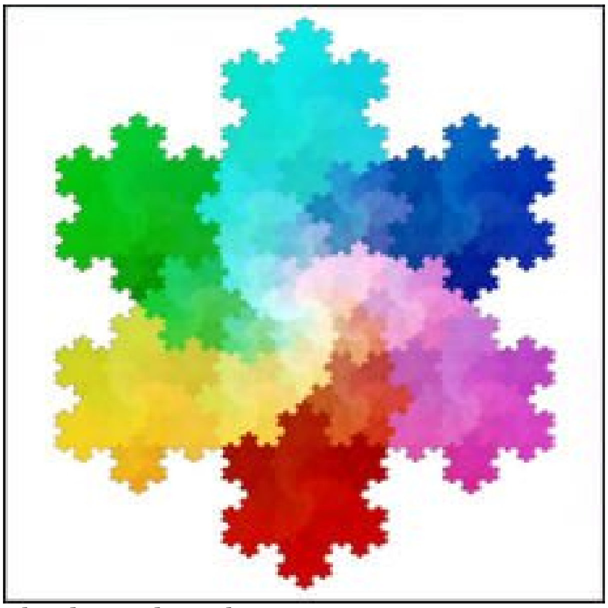
Així, al CD adjunt, podeu trobar-hi els arxius que conformen aquesta web i on a més a més de l'adaptació al format d'hipertext de la versió impresa del treball hi ha els resultats interactius i gràfics de les simulacions fractals que en la versió impresa que teniu a les mans sols es poden veure de forma estàtica.

Finalment, m'he trobat que degut a interactivitat de les pràctiques, i la seva condició d'imatges, el quart bloc on s'exposen, té un volum molt considerable, no degut a la quantitat de contingut sinó a la quantitat d'imatges que requereix el text, necessàries per a mostrar el resultat de les meves pràctiques.

1. Introducció al món dels fractals

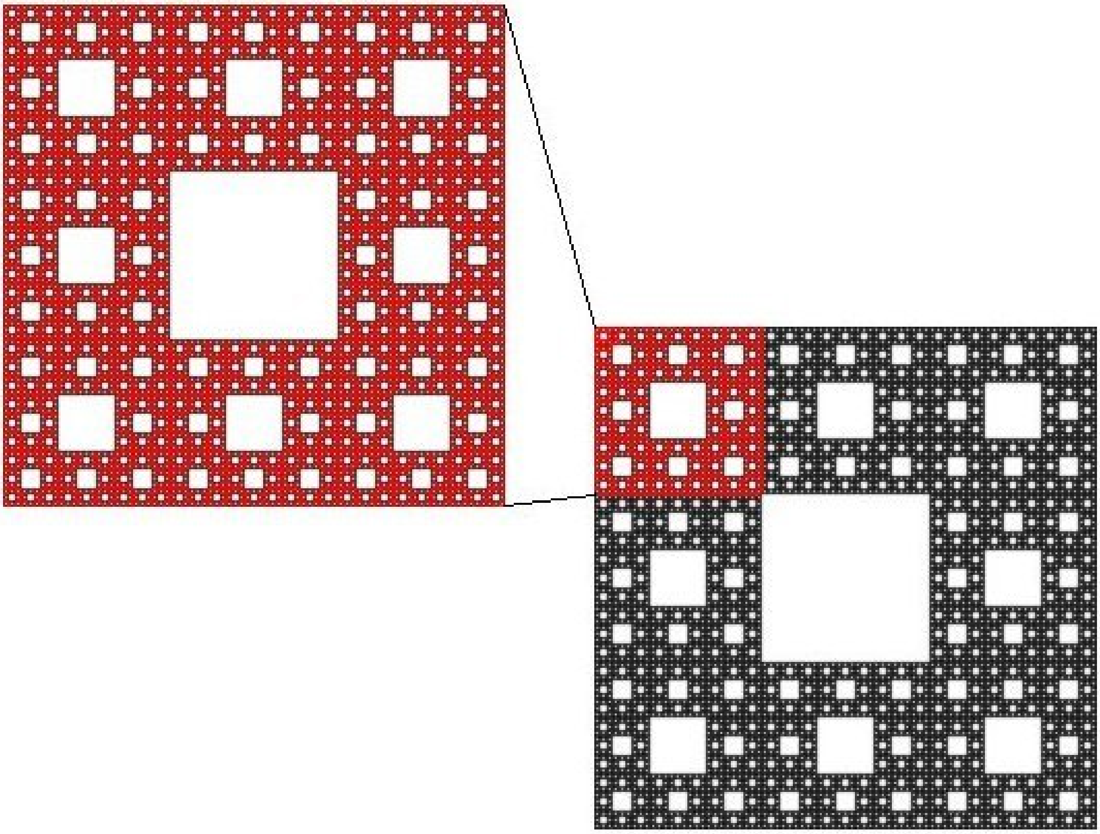
En el nostre llenguatge quotidià, no utilitzem usualment el terme fractal ni el concepte és popularment conegut. Tot i així, els fractals ens envolten i el nostre cos mateix forma estructures fractals.
Hi ha moltes maneres de definir un fractal, però el que fa que ens interessin en aquest treball és que els fractals són fonamentalment un pont entre la natura i les matemàtiques, són una sorprenent explicació d'algunes estructures i regularitats naturals.
Els fractals van ser estudiats llargament per Benoît Mandelbrot (matemàtic del Centre d’Investigació Thomas J. Watson, que la empresa IBM té a Yorktown Heights, Nueva York) i el terme fractal va ser pràcticament implantat per ell gràcies al seu llibre de l’any 1977 “Els objectes fractals” . Anteriorment a Mandelbrot, matemàtics com Kocht, Cantor i Peano entre altres ja treballaven amb objectes que fins a Mandelbrot no varen ser catalogats com a fractals.
Mandelbrot va inventar el terme fractal per designar un seguit d’objectes geomètrics d’estructura irregular que li van cridar l’atenció en el seu intent de trobar una geometria més apropiada que la clàssica per descriure les formes de la natura. A ulls de Mandelbrot, aquesta sèrie de figures abans considerades curiositats dins les matemàtiques sense major interès, van resultar tenir similituds i aspectes en comú prou interessants. Aquests objectes són els fractals. Els que va començar a estudiar Mandelbrot van ser els fractals geomètrics més simples com ara; el conjunt de Cantor, el triangle de Sierpiński, la corba de Peano, el floc de neu de Koch i molts altres.

La principal similitud que tenen aquests objectes entre ells i que va cridar l’atenció de Mandelbrot és l’auto semblança i un seguit de propietats sorprenents com per exemple: longitud infinita, àrea nul·la, etc.



Floc de neu de Koch

Quan parlem d'**auto semblança**, ens referim a que a diferents escales de detall, aquestes figures geomètriques, presenten formes o estructures similars: Una mateixa figura es pot subdividir en peces cada una de les quals és una còpia a diferent escala de la figura grossa.



Auto similitud del Quadrat de Sierpinski

Aquesta és una característica molt important dels fractals. I en conseqüència, trobem que dins un fractal hi ha milers i milers de fractals iguals o similars a ell, cada vegada més petits.

Podem dir que els fractals tenen una estructura geomètrica recursiva, és a dir, són auto semblants, estan compostos d'elements també geomètrics de mida i orientació variable, però d'aspecte similar a l'estructura general.

Una altra molt important característica dels fractals i la més sorprenent és la de tenir dimensió fraccionària. Tots sabem que un punt té dimensió 0, una línia té dimensió 1, un pla té dimensió 2, i un volum té dimensió 3.
Doncs els objectes fractals tenen dimensions que es poden escriure en forma de fracció. Per exemple, una dimensió 5/3 correspon a un cos que es troba entre una línia i un pla ja

que té una dimensió més gran que 1 (línia) i alhora més petita que 2 (pla). El terme fractal provén del llatí fractus que significa “trencat”, “fracturat” o “irregular”.

Ha passat molt poc temps des de que Mandelbrot va formular la definició de fractal però és sorprenent la rapidesa amb que els científics han elaborat models per descriure i comprendre com la naturalesa genera les seves formes i com el creixement a la naturalesa està vinculat a models fractals. I és que sembla que la naturalesa tingui predilecció per l’estètica fractal, es poden trobar fractals en abundància a la natura i al nostre voltant. Si li ho expliquem bé, fins i tot un nen podria trobar formes fractals en múltiples estructures vegetals: fulles, troncs, branques, arrels, en el perfil de les muntanyes, roques i pedres, en la costa d’una platja, en els flocs de neu...

Si posem per exemple el perfil d’una muntanya, trobarem que aquest és constantment modificat per esllavissades, vent o altres fenòmens. La muntanya, es modifica tota ella a gran o a petita escala, de la mateixa manera. Així, trobem en el perfil resultant similituds a petita i a gran escala. Similituds entre el perfil d'una muntanya i el d'una petita pedra de la muntanya.

Muntanyes fractals

Moltes formes de la naturalesa són fractals i molts processos naturals es guien per comportaments fractals.

S’han trobat estructures fractals en la distribució de les estrelles en les galàxies o en la de les galàxies a l'univers, en el perfil de les costes marítimes, en les crestes i les formes de les muntanyes, en els perfils rocosos, en els perfils dels boscos, de les les fronteres, etc. També en el nostre organisme: l'estructura del sistema circulatori, la ramificació de venes, artèries i nervis, la ramificació dels bronquis en els pulmons, les dendrites de les neurones, els alvèols...

Tenen estructura fractal coses sorprenents com per exemple, un imant, un vidre o les ones cerebrals produïdes per un cervell sa.
També es poden trobar fractals en els núvols, en els llampecs i llamps, en els arbres... La nostra vida està plena a vessar de fractals i els fractals són matemàtiques.

Els fractals no permeten explicar ni donen models per a descriure totes les formes naturals però tot i així ens trobem, per primera vegada davant un plantejament que permet descriure i donar resposta a formes geomètriques naturals tant diferents com les que hem vist abans.
Bàsicament, ens trobem davant un mètode per representar i descriure mitjançant unes instruccions senzilles (algoritmes) una gran varietat de formes naturals i alhora fàcil d'utilitzar i manipular amb els ordinadors.

Quan diem que la línia d'una zona de la costa és un fractal, evidentment no volem dir que hi hagi una corba i una fórmula matemàtica que s'ajustin de forma precisa al perfil del litoral. Significa que és possible definir un model matemàtic fractal que s'ajusti amb un marge determinable d'error al perfil de la costa.

Hem dit que a la naturalesa hi ha incomptables fractals però també que els fractals tenen una forma recursiva infinita (auto semblança), es a dir que dins del fractal gros hi ha infinits fractals més petits i de diferents i múltiples mides. Això fa que ens plantegem una qüestió, hi ha realment fractals naturals si la precisió de la natura no arriba als nivells de la definició de fractal? Si responem que no, llavors no n'hi ha de fractals a la natura, però això seria el mateix que dir que en la naturalesa no existeixen esferes ja que la terra o altres planetes no s'ajusten al model del que seria una esfera perfecte. Els fractals no s'ajusten completament al que és la natura i òbviament no existeixen en el món natural com tampoc no existeixen rectes ni esferes, simplement serveixen per a descriure formes naturals fins ara indescriptibles ja que els fractals, com la natura, són massa irregulars per ser descrits fàcilment en el llenguatge geomètric tradicional.

2. Fractals geomètrics més importants. La dimensió fractal

2.1. Presentació dels fractals geomètrics més importants

En alguns conjunts de figures, Mandelbrot va descobrir una discordança entre la seva mesura real i la seva configuració espacial com a conjunt de punts (**corbes de longitud infinita, figures amb una àrea nul·la...**). Aquests conjunts li van semblar prou interessants com per estudiar-los en la seva recerca d'una geometria que descrigués les estructures naturals.

Quan Mandelbrot va "inventar" el mot fractal, el va utilitzar per designar aquesta serie d'objectes geomètrics d'estructura irregular. Mandelbrot no va donar una definició precisa de fractal però va caracteritzar les noves figures amb les tres propietats següents:

1. Figures auto similars. És a dir figures que tenen una estructura tal que dins ella mateixa, l'estructura, es repeteix infinites vegades a escales més petites.
2. Figures amb dimensió fraccionària (no entera). Figures massa complexes per tractar-se només d'un punt, una línia, un pla o un volum, figures que sobrepassen els nivells de la seva dimensió física i per tant tenen una dimensió superior a ella.
3. Conjunts que apareixen després de processos **iteratius** infinits. (Figures que apareixen al **repetir** en elles mateixes un mateix procés per tal d'aconseguir l'auto similitud.)

El conjunt de Cantor

El conjunt de Cantor, introduït pel matemàtic alemany Georg Cantor, és un conjunt fractal que es construeix dividint un segment en tres parts i suprimint el segment central. Tot seguit es procedeix a fer el mateix als segments laterals restants i així indefinidament. De manera que queda un seguit de punts tots situats sobre l'antic segment.

Els punts són abstractes i per tant irrepresentables però podem realitzer un gràfic del

procés iteratiu que conforma el conjunt de Cantor fins a la sisena repetició (iteració):
El segment inicial.

A sota podem veure la primera iteració (el resultat d'aplicar el procés al segment inicial).

Seguidament la segona iteració en la que s'ha aplicat el procés sobre cada un dels segments restants a la primera iteració.

En la tercera iteració veiem com s'ha aplicat un altre cop el procés sobre cada un dels quatre segments de la imatge anterior.

La quarta iteració:

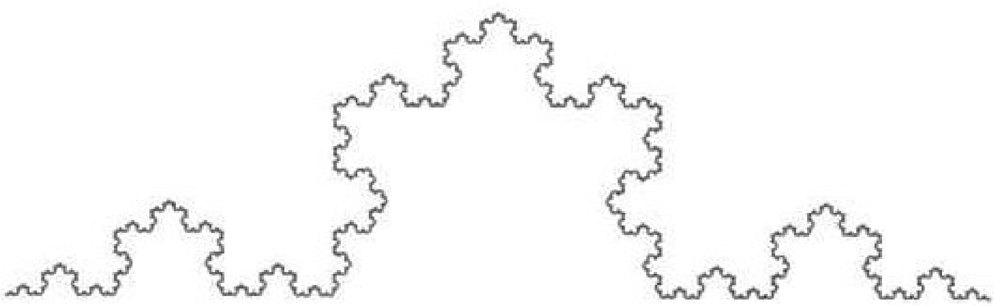
La cinquena iteració:

La sisena iteració gairebé inapreciable:

Aquest és el model de l'inici procediment de fabricació d'un fractal que segueix aquesta repetitiva acció (funció) indefinidament.

La corba de Koch

Una corba de Koch és un objecte fractal que té la propietat de tenir longitud infinita, ser alhora contínua i de no ser derivable en cap dels seus punts. Va ser construïda per Helge von Koch el 1904 i va ser una de les primeres corbes fractals que es van divulgar.

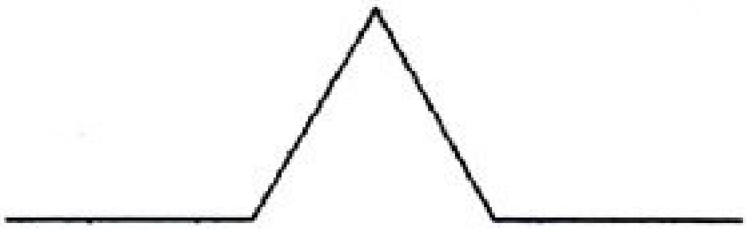


Per construir la corba de Koch s’ha de seguir un algoritme simple partint d’un segment al que s’ha de substituir per quatre segments de mida 1/3 de la mida del segment inicial. Aquests segments es col·loquen de manera que el terç central del segment mare sigui ocupat per dos dels segments petits formant un triangle equilàter sense la base inferior. Seguidament es repeteix el mateix procés en cada un dels segments restants. La corba de Koch és el resultat de seguir el procediment anterior infinitament.

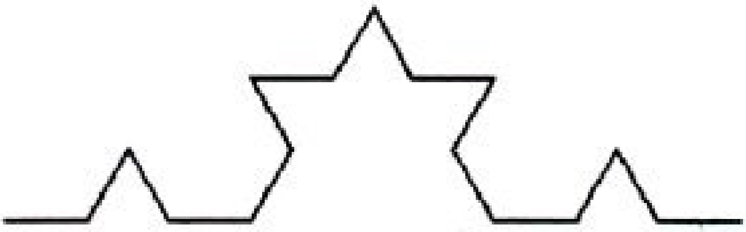
La corba de Koch també pot ser representada pel sistema L-system com comprovarem.

El segment inicial:

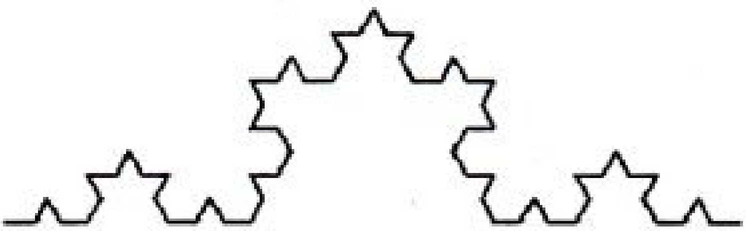
La primera iteració:



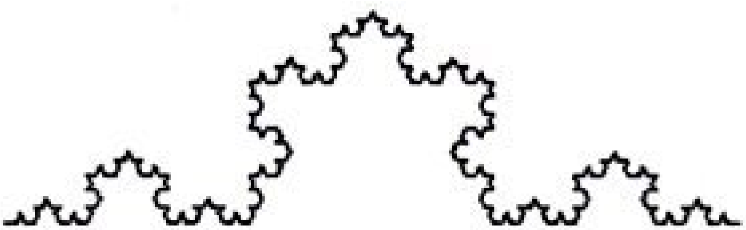
La segona iteració:



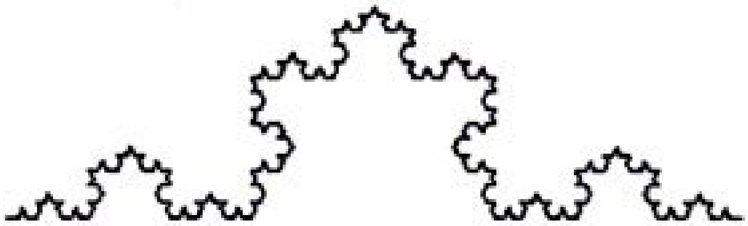
La tercera iteració:



La quarta iteració:

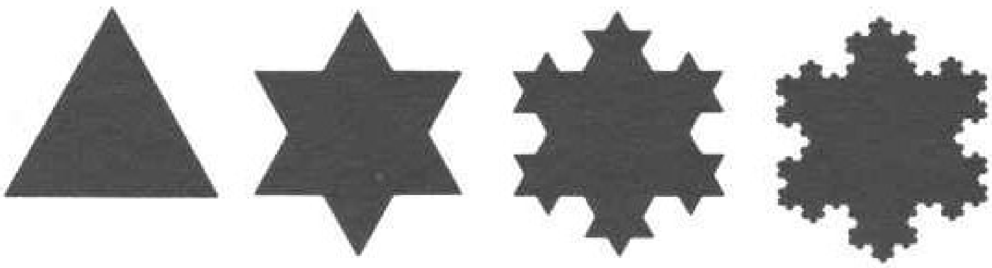


La cinquena iteració:



El floc de neu de Koch

El floc de neu de Koch o estrella de Koch és un conjunt geomètric fractal publicat al mateix any i pel mateix autor que la corba de Koch. Aquesta figura està formada per un triangle equilàter cada un dels costats del qual està format per una corba de Koch. L'àrea del floc de neu de Koch és 8/5 de l’àrea del triangle inicial, per tant, un perímetre infinit format per les tres corbes de Koch, obtingudes iterant la funció als tres costats del triangle equilàter, engloba una àrea finita.



Primeres iteracions del Floc de neu de Koch

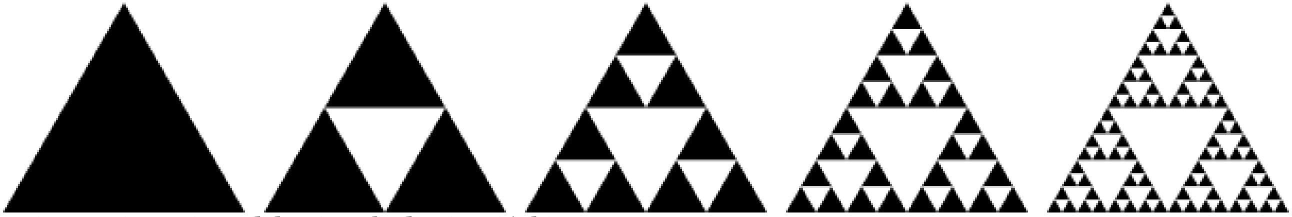
Triangle i tetraedre de Sierpiński

El triangle de Sierpiński és un objecte fractal, que va ser introduït el 1915 pel matemàtic polonès Wacław Sierpiński. Es pot construir a partir de qualsevol triangle encara que inicialment es va construir a partir d’un triangle equilàter.

Per a la construcció del triangle de Sierpiński s’ha de seguir un algorisme molt simple:

1. Un triangle inicial, es divideix en quatre triangles unint els punts mitjans dels seus costats
2. S'elimina el triangle interior
3. En cada un dels tres triangles restants es procedeix a fer el pas 1

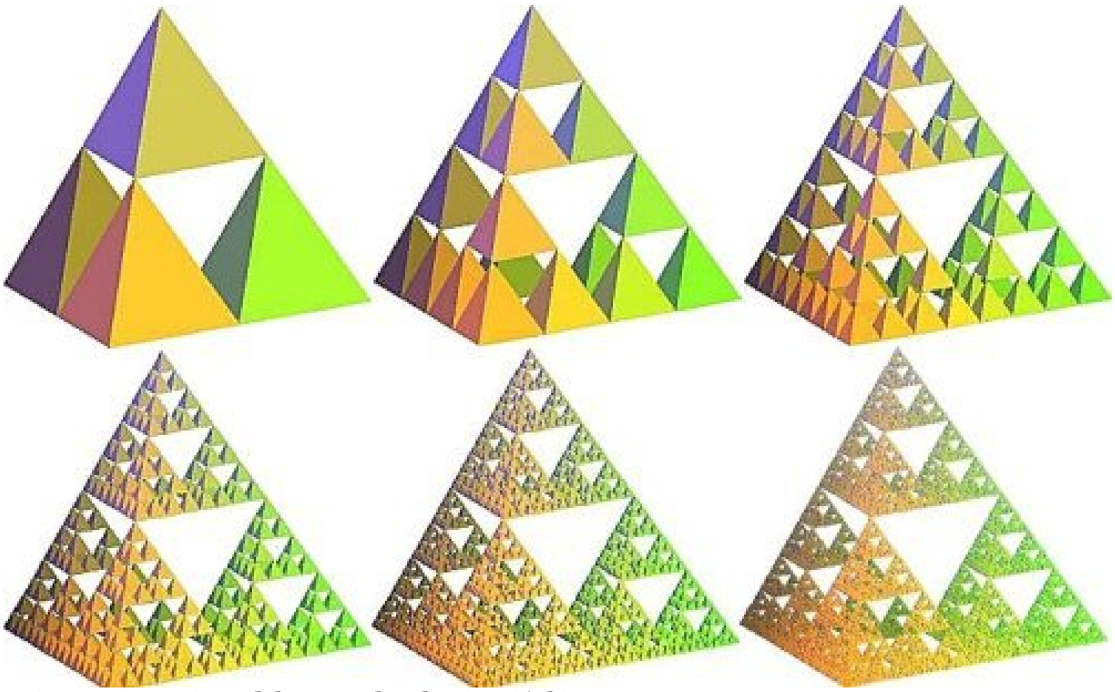
El triangle de Sierpiński és el resultat de fer el procediment anterior indefinidament.



Primeres iteracions del Triangle de Sierpiński

Una altra manera d'originar el triangle de Sierpiński és a través d'un sistema d'iteració de funcions basat en la substitució, anomenat L-system, que explicarem més endavant.
La àrea d'aquest triangle iterat infinites vegades és zero, ja que cada cop que iterem l'àrea es redueix a 3/4 de l'anterior i per tant iterat moltes vegades, l'àrea tendeix a zero.

El Tetraedre de Sierpiński és la versió tridimensional del triangle de Sierpiński. Ja que es parteix d'un tetraedre regular i es divideix en tetraedres de costat meitat i s'eliminen els que no són a les puntes. Seguidament s'itera el procediment en els tetraedres restants.

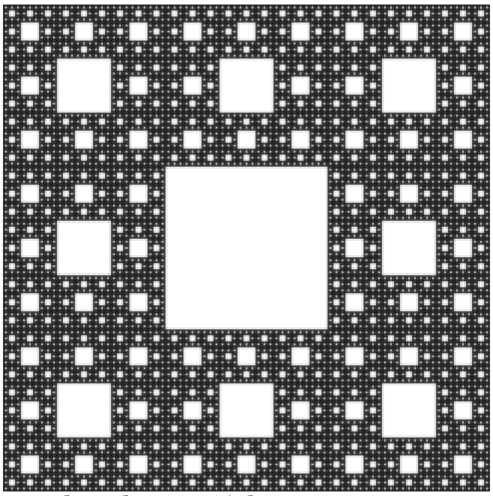


Primeres iteracions del Tetraedre de Sierpiński

El quadrat de Sierpiński

El quadrat de Sierpiński és un objecte pla fractal descrit per Wacław Sierpiński el 1916. El

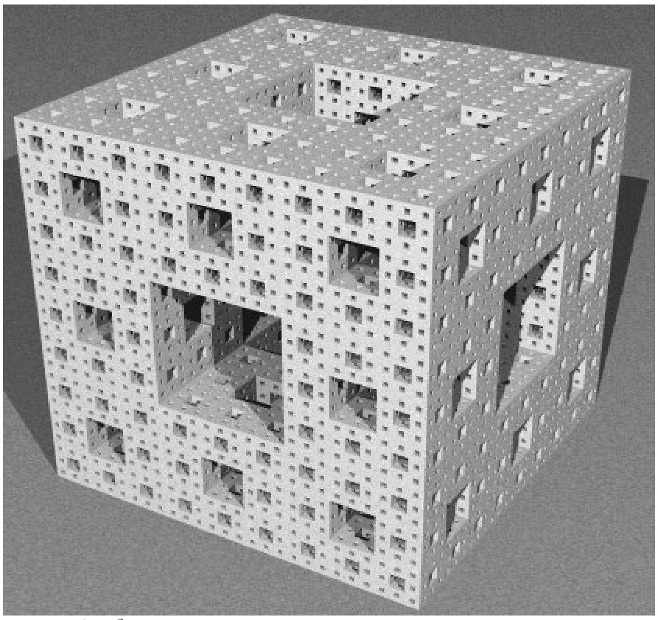
quadrat és una generalització del Conjunt de Cantor en dos dimensions.
Per construir-lo, s'ha de començar amb un quadrat que és dividit en nou quadrats iguals, tres per banda i un al centre el qual és eliminat. Seguidament s'itera l'operació en els vuit quadrats restants.



Quadrat de Sierpiński

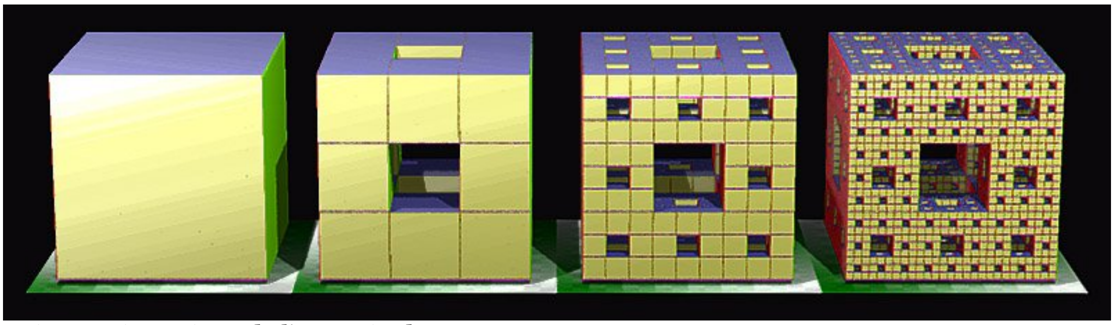
L'esponja de Menger

L'esponja de Menger és una versió tridimensional del Conjunt de Cantor o del Quadrat de Sierpiński. Va ser descrit per un matemàtic Austríac anomenat Parl Menger el 1926.



Esponja de Menger

Per construir-lo s'ha de començar des d'un cub, i dividir-lo en 27 petits cubs (9 per costat i un al centre). Com es feia al construir el quadrat de Sierpiński, s'elimina el cub central de cada cara i el del centre, així, queden **20** cubs iguals en els quals es repeteix l'operació anterior infinitament.



Primeres iteracions de l'Esponja de Menger

El nombre de cubs de la formació en cada noment es pot trobar amb: 20^n. On n és el nombre d'iteracions (vegades que s'ha repetit el procés explicat):

0 iteracions --> 1 cub

1 iteracions --> 20 cubs

2 iteracions --> 400 cubs

3 iteracions --> 8.000 cubs

4 iteracions --> 160.000 cubs

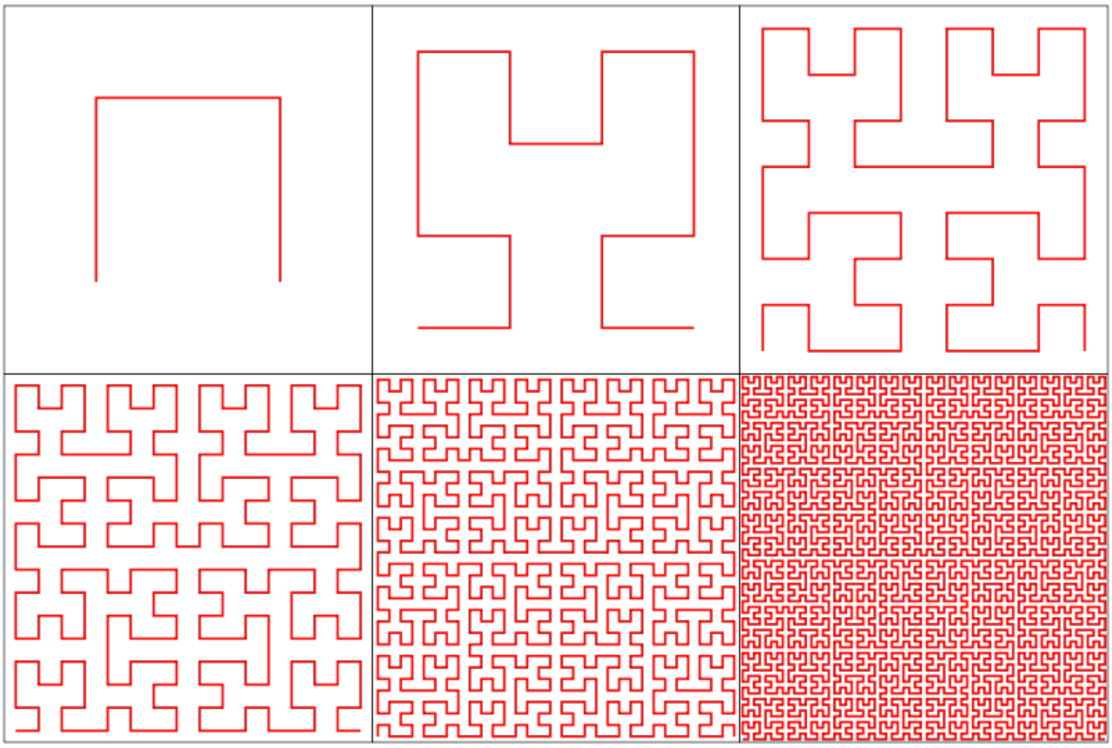
5 iteracions --> 3.200.000 cubs

6 iteracions --> 64.000.000 cubs

Les corbes de Peano i Hilbert

El 1890, Peano va construir un corba contínua que passa per tots els punts del quadrat unitat. Era el primer exemple d'una corba que omple un espai. Anys més tard Hilbert va construir-ne una de la mateixa naturalesa més simple de descriure.

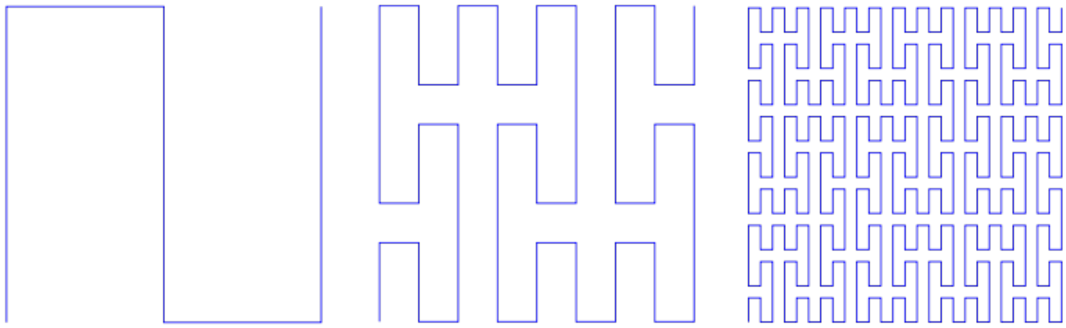
La corba de Hilbert es construeix dividint el un quadrat del que es parteix en quatre quadrats més petits i unint els centres de cada quadrat amb una línia contínua que comença al centre del quadrat inferior esquerre i acaba al quadrat inferior dret. Tot seguit es repeteix l'operació a cada quadrat de mida 1/2 de l'inicial substituint el tros de corba ja existent per la corresponent a l'operació.
Es continua indefinidament amb aquest procés unint els centres dels quadrats que apareixen a cada etapa.


Primeres iteracions de la corba de Hilbert

La corba resultat d'iterar infinitament aquesta formula omple el quadrat unitat i rep el nom de corba de Hilbert.

La corba de Peano, es construeix de manera molt similar a la corba de Hilbert.
Un quadrat es divideix en 9 quadrats més petits i iguals i en cada un d'ells es dibuixa una

corba com la representada en la primera casella. Al iterar dos vegades aquesta funció s'obté una corba com la de la tercera casella.



Primeres iteracions de la corba de Peano

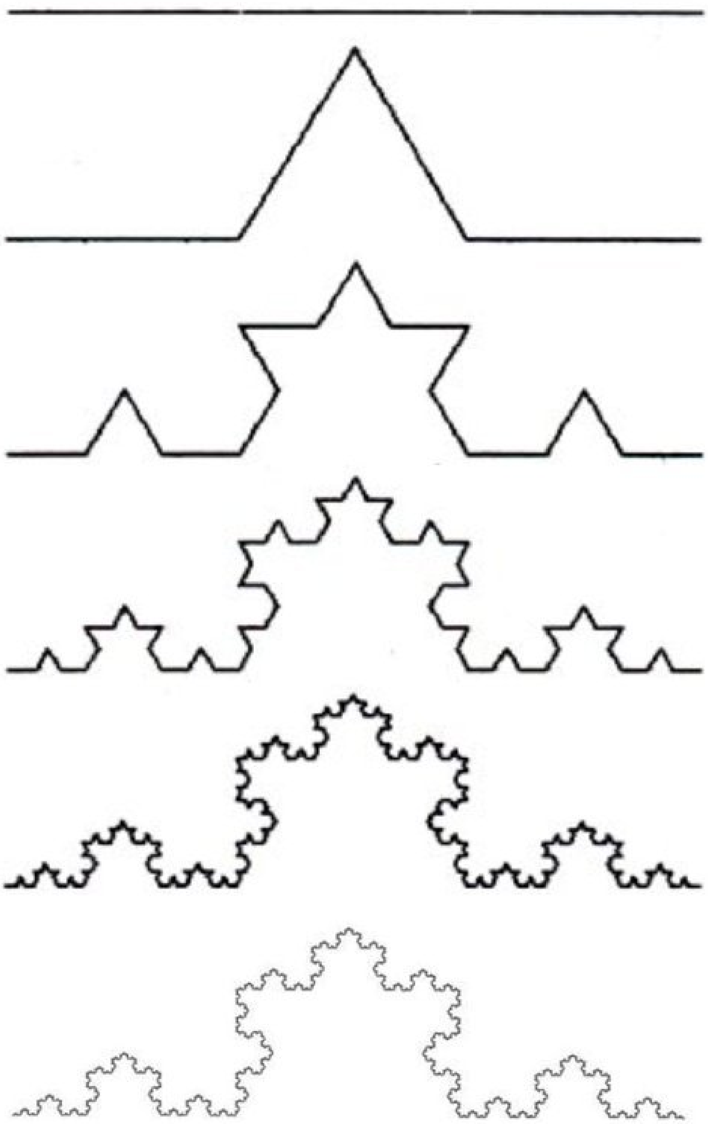
Com hem dit abans, si iterem indefinidament aquests dos processos, obtindrem simplement un pla, una figura de dimensió 2.

2.2. Dimensió fraccionària, el seu càlcul

L'aparició d'unes noves estructures fractals fins ara considerades sense importància va obligar a inventar una nova geometria, unes noves regles per tal de mesurar-les ja que els fractals no es poden mesurar dins els límits de la geometria clàssica o euclidiana.
La principal diferència entre la geometria fractal i la geometria clàssica és que aquesta última, presenta contorns diferenciables i mesurables. En la geometria fractal, en canvi, apareixen contorns indefinits, irregulars, difícils de mesurar.
Si intentem determinar la longitud del contorn d'una illa el resultat dependrà de la resolució del mapa, quanta més resolució més longitud. Per això, compararem els fractals entre sí determinant el seu grau d'irregularitat, utilitzant com a mesura la dimensió i no les habituals magnituds de la geometria clàssica.

El concepte de longitud dins el món dels fractals, no es pot utilitzar com en la geometria clàssica.

Ho podem veure molt bé amb la corba de Koch.



Primeres iteracions de la Corba de Koch

Cada vegada que iterem el procés, ho sigui que repetim l'operació de substitució del segment per quatre segments més petits, ens trobem que la longitud de la línia augmenta un terç de l'anterior: és a dir que la corba que ocupa l'espai a l'inici del procés, va augmentant pas a pas la seva longitud de manera que la longitud d'aquesta corba és infinita, no es pot mesurar. Cada corba es 4/3 de l'anterior.

I encara que volguéssim mesurar la seva longitud, qualsevol instrument o unitat amb la que ho intentem, es quedarà grossa ja que l'apreciació a la que arriba l'instrument sempre té un límit i en canvi la corba de Koch no té límits de longitud.

Així doncs, va sorgir una nova necessitat per a mesurar una nova classe d'objectes que no es podien mesurar com es feia fins llavors.
Això va obligar a crear un nou concepte: La dimensió fractal.
Abans d'introduir a les matemàtiques el concepte de la dimensió fractal, hi havia un concepte de dimensió definit per Poincaré: la dimensió euclidiana o topològica que es podia definir fàcilment amb dos condicions:

1. Un conjunt buit té dimensió –1
2. Un objecte té una dimensió un grau superior a la dels punts que l'envolten o les parts que el formen.

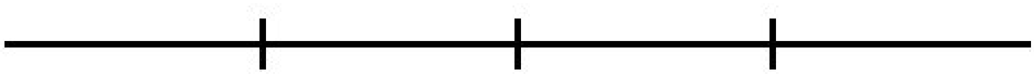
Així doncs trobem que:

Un conjunt buit té dimensió: D = -1
Un punt té dimensió: D = 0 (està envoltat de buit)
Un segment té dimensió: D = 1 (està format per punts)
Un quadrat té dimensió: D = 2 (està format per segments)
Un cub té dimensió: D = 3 (està format per quadrats)

Cap sorpresa de moment ja que aquests resultats coincideixen amb el concepte popular de dimensió. Així doncs, tots tenim un concepte de dimensió que normalment s'até al concepte de la dimensió euclidiana.
La dimensió fractal, com veurem, és una generalització de la dimensió euclidiana i va ser definida per Felix Hausdorff el 1919 abans de que Mandelbrot definís els fractals com a tals.

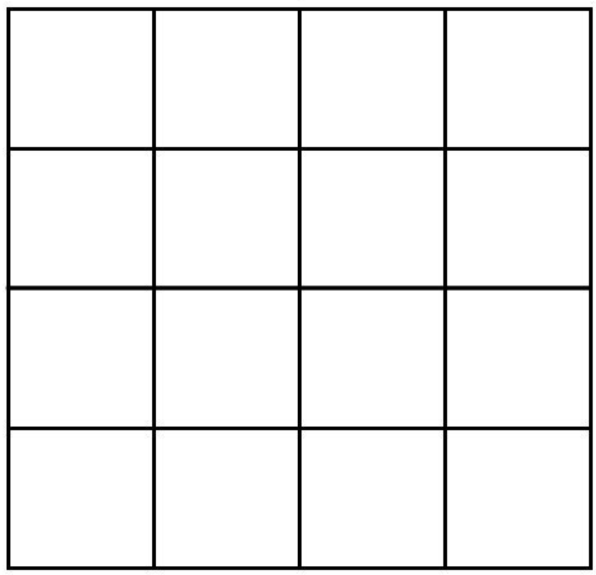
Hausdorff va idear un concepte de dimensió aplicable a aquests tipus de corbes i superfícies estranyes que apareixien en els objectes en que ell es basava: Conjunt de Cantor, Corba de Peano, etc.
Anys després, Mandelbrot va definir els fractals com: els objectes la dimensió de Hausdorff dels quals, no fos entera.
La dimensió de Hausdorff es calcula mitjançant una fórmula:
Partint d'un segment (també auto similar) podem dividir-lo en quatre trossos iguals entre

ells. Cada tros és ¼ de la mida del segment original i per tant es necessiten $4 = 4^1$ segments petits per cobrir el segment original.



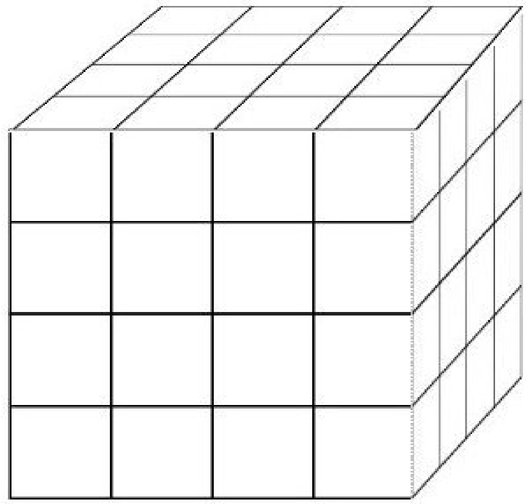
Segment de dimensió 1 dividit en quatre trossos

Si ara considerem un quadrat i el subdividim en quadrats més petits cada un amb longitud lateral igual a ¼ de la longitud lateral del quadrat inicial, necessitarem $16 = 4^2$ per reconstruir el quadrat original.



Quadrat de dimensió 2 dividit entre 16 trossos

Passem al cub. Si dividim el cub original en cubs més petits tots amb aresta de mesura ¼ de la mesura de l'aresta del quadrat original necessitarem, després, $64 = 4^3$ cubs petits per tal de reconstruir el gros.



Cub de dimensió 3 dividit entre 64 trossos

En els casos anteriors, l'exponent té molta importància ja que ens dóna la dimensió.

Així veiem que:

1. En una línia (dimensió 1) necessitem $4 = 4^1$ trossos per a cobrir el segment original.
2. En un quadrat (dimensió 2) necessitem $16 = 4^2$ trossos per a completar el quadrat original.
3. En un cub (dimensió 3) necessitem $64 = 4^3$ trossos per a completar el cub original.

Si generalitzem podrem extreure'n la formula. Posem a :

D = Dimensió

(1,2,3 en aquests casos)

N = Al número de peces petites necessàries per a cobrir la figura inicial.

(4, 16, 64 en els exemples anteriors)

S = Al factor escala. És a dir el quocient entre una mida presa a la figura gran i la mateixa mida presa a la figura petita en una mateixa unitat.

(4 = 4(unitats)/1(unitats) en els casos anteriors)

Així doncs:

$\mathbf{N = S^D}$

Apliquem les propietats logarítmiques:

$\log S^D = \log N$

$D \log S = \log N$

$\mathbf{D = \log N / \log S}$

Ja tenim una fórmula que segons Hausdorff ens permet calcular la dimensió d'una figura sigui fractal o no.
Hem de pensar que aquesta és una definició de dimensió aplicable fàcilment dins el camp dels fractals en alguns casos però no en d'altres.
Hi ha fractals que no tenen auto similitud estricte com per exemple els fractals del pla complex. Els fractals del pla complex tenen auto similitud estadística (l'estructura repetida no és exactament igual a la grossa i no es presenten les rèpliques sempre en la mateixa posició ni en la mateixa escala que les seves germanes) i per tant, això fa molt més complicat aplicar la fórmula. Tot i així amb l'ajut dels ordinadors i de les noves tecnologies s'ha pogut calcular la dimensió fractal de molts d'ells.
Hi ha altres fractals que tenen auto similitud estadística però que ens permeten calcular la dimensió amb la mateixa fórmula. Com veurem, un exemple són els perfils naturals ja que no són infinitament irregulars, tenen un límit com tots els fractals naturals, i això ens permet calcular-ne la dimensió.

2.3. Càlcul de la dimensió fraccionaria d'alguns fractals geomètrics

En aquest capítol, estudiarem matemàticament **els fractals geomètrics** presentats anteriorment calculant-ne la dimensió fractal aplicant la fórmula de Hausdorff.

Per aconseguir-ho, cal que apliquem la formula abans explicada:

D = log N / log S

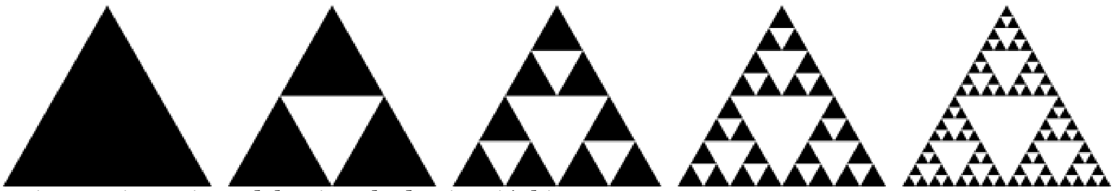
On: **D** = Dimensió

N = El nombre de peces petites necessàries per a cobrir la figura inicial.

S = El factor escala (el quocient entre una mida presa a la figura gran i la mateixa mida presa a la figura petita en una mateixa unitat.)

La dimensió del Triangle de Sierpiński

Com a primer exemple observem al Triangle de Sierpiński iniciat amb un triangle al qual dividim entre quatre triangles més petits dels quals només a tres iterarem la operació. (Hi haurà tres rèpliques de mida 1/2 del costat de la figura gran dins seu).



Primeres iteracions del Triangle de Sierpiński

Ja no ens cal res més.

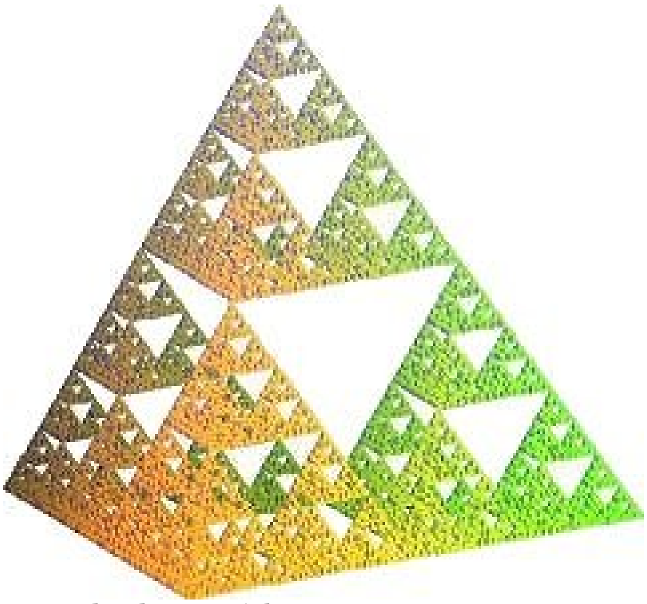
N = 3 (hi ha tres rèpliques petites dins el gros)

S = 2/1 = 2 (Costat de la figura gran dividit entre el costat de la figura petita)

D = log 3 / log 2 ≈ 1,585

Ens ha sortit una dimensió més gran que 1 (segment) però més petita que 2 (pla) així doncs, és massa complicat per a ser una línia però massa buit per a ser un pla. Té dimensió fractal.

La dimensió del Tetraedre de Sierpiński



Tetraedre de Sierpiński

N = 4 (3 tetraedres a la base i un sobre seu)

S = 2/1 = 2 (L'aresta dels petits és 1/2 de l'aresta de la gran)

D = log 4 / log 2 = 2

Així encara que és una figura situada a l'espai té dimensió 2.

La dimensió del Conjunt de Cantor

Comencem amb un segment, el dividim en tres trossos, i només iterem l'operació als segments laterals de longitud 1/3 cada un de la longitud del segment inicial.

Primeres iteracions del Conjunt de Cantor

Així ja ho tenim tot:

N = 2 (hi ha dos rèpliques dins el total)

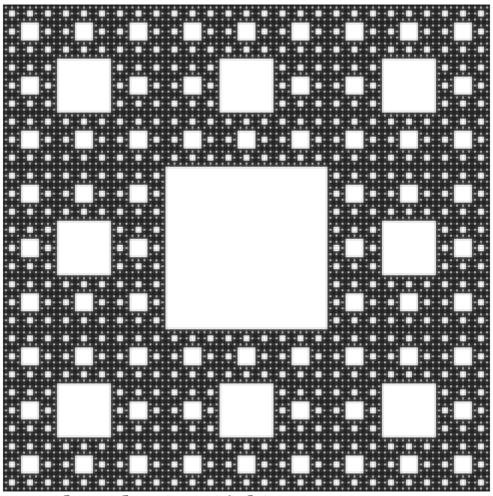
S = 3/1 = 3 (Llargària del segment inicial dividida entre la llargària del segment menor.)

D = log 2 / log 3 ≈ 0' 631

Com ens esperàvem, no arriba a la dimensió 1 però tampoc no és un punt de dimensió 0, és un conjunt de punts infinitament petits inapreciables i irrepresentables degut a l'abstracció que comporta. Podeu comprendre per què aquest conjunt també s'anomena pols de Cantor.

La dimensió del Quadrat de Sierpiński

Comencem amb un quadrat, el dividim en nou quadrats més petits, iguals entre ells i només iterem l'operació en vuit d'ells.



Quadrat de Sierpiński

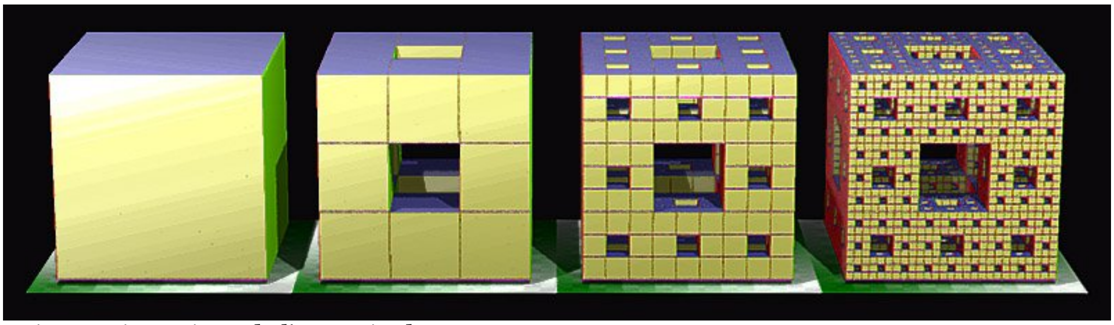
N = 8

S = 3/1 = 3

D = log 8 / log 3 ≈ 1'893

Gairebé un pla però no encara.

La dimensió de l'Esponja de Menger



Primeres iteracions de l'Esponja de Menger

Molt semblant a l'anterior. Tenim 20 cubs de mida 1/3 del gros en els quals iterarem la funció.

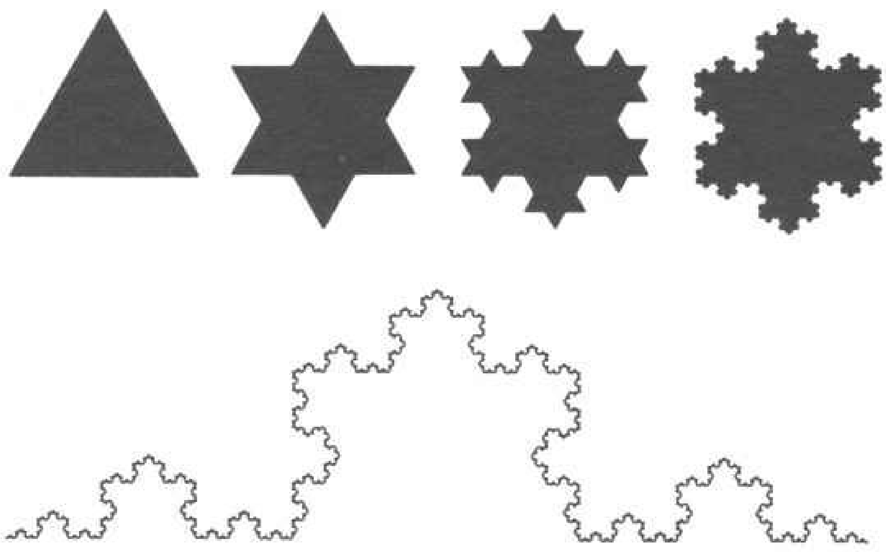
N = 20

S = 3/1 = 3

D = log 20 / log 3 ≈ 2'727

No arriba a ser un volum.

La dimensió de la Corba i el Floc de neu de Koch



Primeres iteracions del Floc de neu de Koch i la Corba de Koch

Tenim quatre segments que cobreixen el segment total sobre els quals iterarem la funció i cada un d'ells és 1/3 del total.

N = 4

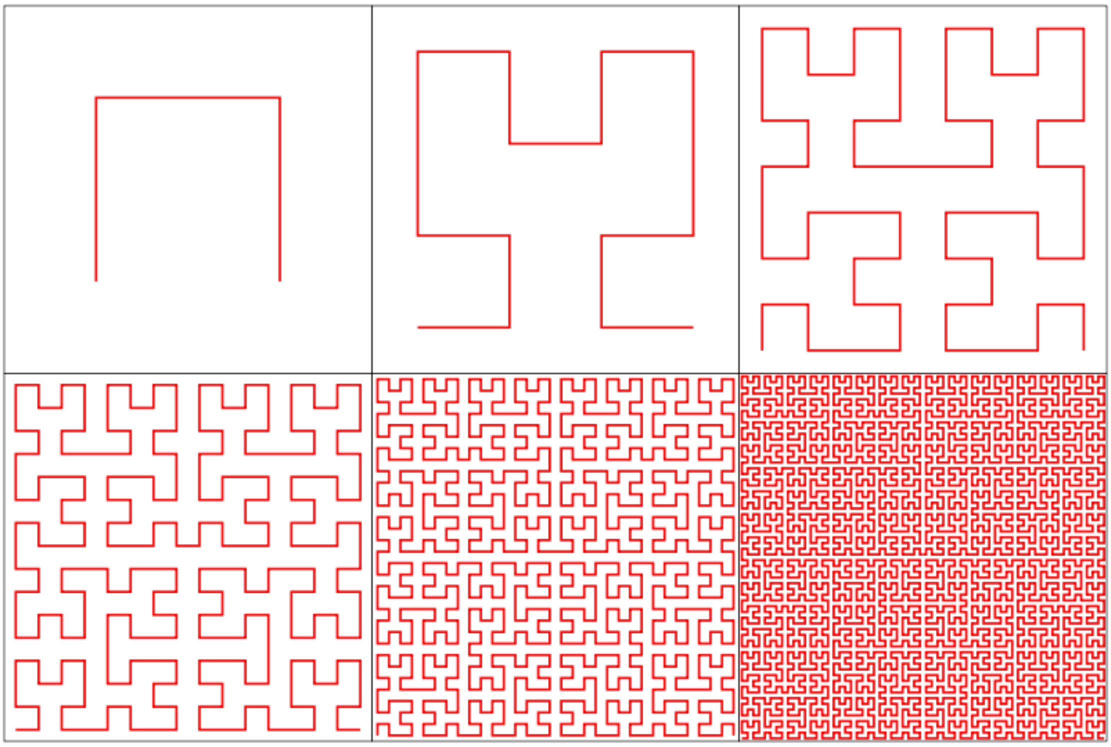
S = 3/1 = 3

D = log 4 / log 3 ≈ 1'262

Més que una línia.

El floc de neu i la corba tenen la mateixa dimensió fractal ja que el floc està format per tres corbes . El floc no parteix d'un pla. Només és una corba tancada.

La dimensió de la Corba de Hilbert



Primeres iteracions de la Corba de Hilbert

Iterem la funció en cada quart del cub inicial.

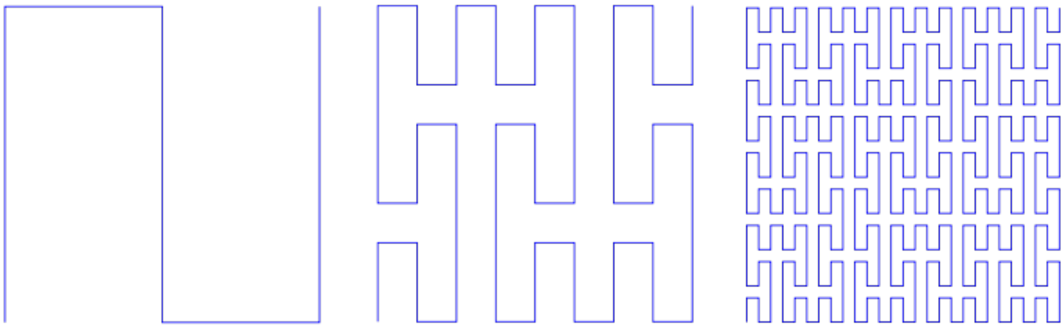
N = 4

S = 2/1 = 2

D = log 4 / log 2 = 2

Ja sabíem el que ens havia de sortir, si ho recordeu, les corbes de Hilbert i de Peano cobreixen el pla i tenen dimensió 2.

La dimensió de la Corba de Peano



Primeres iteracions de la Corba de Peano

Iterem la funció nou vegades dins el cub inicial.

N = 9

S = 3/1 = 3

D = log 9 / log 3 = 2

Com esperàvem.

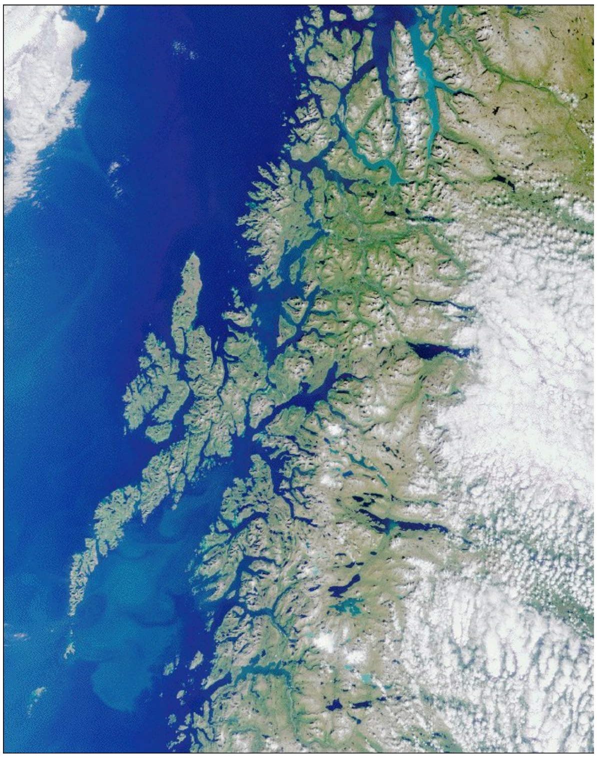
2.4. Càlcul de la dimensió fraccionària de la costa de Mallorca

El 1967, Mandelbrot va rebre l'encarreg de calcular la longitud de la costa de Gran Bretanya. Al primer cop d'ull, pot semblar una empresa fàcil de dur a terme valent-se d'un mapa de l'illa, però Mandelbrot va demostrar que no és tan senzill com sembla.

Si busquem en un atles, podrem llegir que la costa de Gran Bretanya mesura 12.429 kilòmetres (aquesta és la mesura oficial) però, com veurem, aquesta resposta no és ni molt menys correcta.

Mandelbrot es va valdre d'un compàs i un mapa i va realitzar varies mesures amb diferents angles d'obertura del compàs. Es va trobar amb que al variar l'angle d'obertura del compàs, també variava la mesura obtinguda de la longitud de la costa. Quan més petit era l'angle més gran era la longitud de la costa. La línia de la costa va resultar impossible de mesurar ja que té longitud infinita.

Imaginem-nos el mapa de Gran Bretanya o de qualsevol altre illa (per a fer-ho més simple) veurem irregularitats en la línia de la costa (golfs, puntes, entrades, platges...). Però què passaria si agaféssim un mapa en una escala més ampliada en que veiéssim l'illa més grossa? Veuríem, està clar, les mateixes entrades, golfs o platges però també distingiríem més entrades, més platges o més golfs de mida inferior a les que abans havíem vist i que en l'altre mapa no arribàvem a apreciar:



Si intentem de mesurar-ne el perímetre, ens trobarem que té una forma tan irregular que les unitats de mesura no s'adapten totalment a la seva línia de costa i una unitat més petita s'hi adaptaria més però la longitud resultant del càlcul seria molt més gran i que sempre hi hauria una altra unitat més petita que s'adaptaria més a d'irregularitat de la figura i ens donaria una mesura més gran. Aquesta dificultat a l'hora de mesurar la longitud d'una corba degut a les irregularitats que presenta la mateixa és una característica de les corbes y superfícies fractals.

Mandelbrot va publicar el mateix 1967 un article titulat: “Quan mesura la costa de Gran Bretanya?”(“How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional

Dimension") en el qual parlava del seu intent de mesurar la costa de Gran Bretanya i del problema amb que s'havia trobat. També parla de corbes auto similars que tenen dimensió fraccionaria entre 1 i 2 tot i que encara no les anomena fractals.

Mandelbrot va concloure de la seva experiència que les costes i altres corbes geogràfiques tenen la propietat d'auto similitud estadística. L'article, tot i així no assegura que la línia de la costa o alguna altra corba geogràfica sigui realment un fractal (ja que seria físicament impossible) només diu que les línies naturals poden comportar-se estadísticament com a fractals mesurades des de diferents escales i unitats de mesura.
Aquest va ser el principi de la recerca de Mandelbrot i el que el va introduir dins el món dels fractals, va impulsar-lo a estudiar-los per trobar una geometria per a les formes naturals.

Al principi dels anys setanta, Mandelbrot va publicar un llibre anomenat *"els objectes fractals", "Forma, atzar i dimensió"*. En aquest llibre, Mandelbrot exposa el seu estudi sobre fractals i la seva recerca i troballa d'una magnitud que pogués mesurar la complexitat de la costa de Gran Bretanya: la dimensió fraccionària. Així doncs, Mandelbrot mesura la dimensió fraccionària de la seva illa amb el resultat obtingut d'una dimensió d'1,25.

Imatges de l'illa de Gran Bretanya mesurades amb la mateixa unitat però a nivells d'apreciació diferents.

Igual que va fer Mandelbrot amb l'illa de Gran

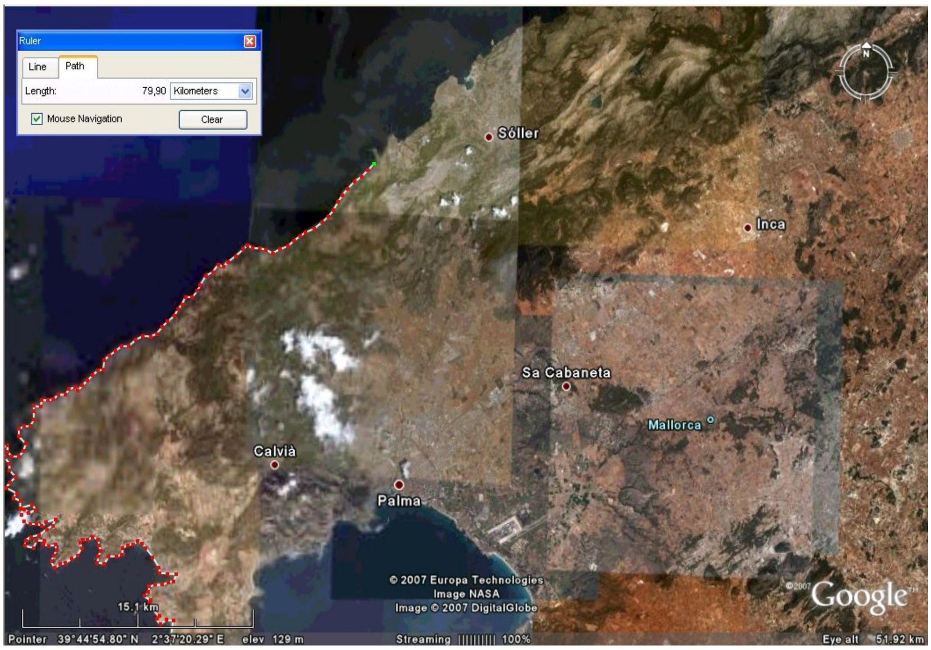
Bretanya intentarem mesurar la dimensió fractal de la línia de la costa de l'illa de Mallorca. Es tracta d'un fractal natural i per tant un nou repte per a la nostre recerca ja que, degut al seu origen natural, no té auto semblança estricte sinó estadística.

Per a fer aquesta pràctica, cal comparar la mesura feta del perímetre de la costa en una escala i la mateixa mesura però feta en una altre escala. És a dir la variació de la longitud de la costa en variació de la escala. (quant més exacte sigui l'escala més detalls es veuran i més llarga ens resultarà ser la línia de la costa.)

Per tal de obtenir un resultat el més ajustat a la realitat possible, en lloc de fer l'operació comparant dos escales diferents, ho farem comparant quatre escales diferents, cada una amb totes les altres, i així, després, podrem trobar la mitjana aritmètica de tots els resultats.

En quant a la dimensió, cal esperar un resultat comprés entre el 1 i el 2 ja que es tracta d'una línia fractal i per tant més que una línia però menys que un pla. Esperem un resultat proper, encara que superior al 1 ja que no es tracta d'una línia molt complicada.

El primer pas per al càlcul va ser trobar el mètode de mesura del perímetre sobre un mapa. Vam optar per fer servir el programa de “Google Earth” <**http://earth.google.com/intl/es**> que ens permetia mesurar a diferents escales (totes les desitjades). I mesurar amb més comoditat el perímetre de l'illa amb l'ajuda d'aquesta eina especial dissenyada per a mesurar camins, rutes o carreteres.



En aquesta captura de pantalla del programa, veiem com a la part inferior esquerra hi ha un segment representant l'escala gràfica visual. Tot el seguit de punts vermells que envolten una part de l'illa, són els "clics" que he anat fent amb ratolí resseguint la costa. Un cop completat el perímetre, el programa ens dóna la suma de tots els segments entre clics.

Captura de pantalla durant la realització de la pràctica

Aquest procés l'he repetit 4 vegades una per cada escala visual, i els resultats han estat els següents:

Escala visual: 30 Km --> Perímetre: 390,46 Km
Escala visual: 15 Km --> Perímetre: 430,35 Km
Escala visual: 10 Km --> Perímetre: 442,92 Km
Escala visual: 5 Km --> Perímetre: 492,90 Km

Per tal de poder treballar, necessitem totes les dades en una mateixa unitat. L'escala visual del programa és un segment de longitud fixe que apareix a la part inferior de la pantalla, sobre aquest segment hi ha un nombre que indica la longitud en quilòmetres del segment, valor que varia segons l'escala. Així, per exemple, si sobre el segment hi diu 5 Km si transportes al mapa la mateixa longitud aquesta ocuparà 5 Km però si hi diu 10 Km, el mapa tindrà menys resolució i el segment dins el mapa ocuparà el doble de tros. Així, podem considerar el segment com a unitat ja que necessitem mesurar la línia de la costa en les diferents imatges, amb la mateixa unitat (la mateixa regla).
Així doncs, ens servirà com a unitat. Hem passat totes les mesures preses en kilòmetres a unitats mitjançant l'equivalència expressada en la primera columna de la taula.

Escala	Mesura en Km	Mesura en unitats
1 unitat : **30** Km	390,46 Km	**13,02 u**
1 unitat : **15** Km	430,35 Km	**28,69 u**
1 unitat : **10** Km	442,92 Km	**44,29 u**
1 unitat : **5** Km	492,90 Km	**98,58 u**

Tenim la nostra fórmula:

D = log N / log S

On: **D** = Dimensió

N = Al número de peces petites necessàries per a cobrir la figura inicial. (la relació entre la longitud en una escala més precisa (l) i una altre més allunyada(L)). (N = l/L)

S = Al factor escala (el quocient entre una mida presa a l'escala gran (M) i la mateixa mida presa a l'escala petita (m) en una mateixa unitat.) (S = M/m)

I l'apliquem a les nostres dades:

	L	l	N = l/L	M	m	S = M/m	Dimensió
30-15	13,02	28,69	2,2	30	15	2	1,14
30-10	13,02	44,29	3,4	30	10	3	1,11
30-5	13,02	98,58	7,57	30	5	6	1,13
15-10	28,69	44,29	1,54	15	10	1,5	1,07
15-5	28,69	98,58	3,44	15	5	3	1,12
10-5	44,29	98,58	2,23	10	5	2	1,15
Mitjana aritmètica							**1,12**

El resultat de l'operació, la dimensió fractal de la línia de la costa de Mallorca, el resultat de l'operació és un valor dins del que es podia esperar ja que la dimensió ens diu que es tracta d'una mica més que una línia (supera encara que no per molt la dimensió 1 , la línia) però no es separa molt de la dimensió 1, raonable donades les propietats de l'objecte en qüestió. Ara tenim calculada la dimensió de la costa de l'illa de Mallorca i la podem comparar amb la dimensió de la costa de Gran Bretanya que va calcular Mandelbrot 1,25. Comparant-les es conclou que la Gran Bretanya té una costa una mica més fraccionada que la de Mallorca que s'acosta més a una línia.

3. Fractals al pla complex. (Mandelbrot i Julia)

Fins a l'arribada dels ordinadors, l'elevat nombre de càlculs que implicaven els fractals i l'increïble exactitud que demanaven, els feia impracticables en la vida real.
Benoit Mandelbrot va ser el primer a utilitzar els ordinadors per a fer representacions gràfiques de fractals **al pla complex** basant-se en fórmules descrites pel matemàtic francès Gaston Julia a començament de segle passat. Actualment les capacitats dels ordinadors permeten calcular a gran velocitat milers d'operacions i de representar amb molta exactitud gràfiques fractals.
El fractals al pla complex són una classe de fractals que neixen a partir de la iteració de funcions al pla complex. Aquest tipus de fractals, es representen només a partir de l'ordinador ja que requereixen moltes operacions i, sense una màquina que les faci, és complicat i llarg.

3.1. Iteració amb variable real

Ja hem esmentat més d'una vegada la paraula iterar, i és que els fractals es basen en fórmules matemàtiques iterades infinitament o fins que l'apreciació ho permeti. Iterar vol dir literalment repetir i iteració és l'efecte d'iterar o l'acció d'iterar.
Dins les matemàtiques es fa servir el concepte d'**iteració de funcions** que es tracta simplement d'aplicar una funció (es a dir un canvi) a un número o a una variable inicial i, després, tornar a aplicar la mateixa funció al resultat obtingut, repetir, iterar tantes vegades com calgui. També es podria dir que una funció iterada és una funció composta amb ella mateixa de forma repetida o infinitament.

Posarem un exemple senzill d'iteració:

Tenim una funció (unes normes per al canvi) :

A --> AB

B --> A

Comencem amb el conjunt:

A

I iterem la funció en el conjunt inicial:

1era iteració	AB
2ona iteració	ABA
3era iteració	ABAAB
4rta iteració	ABAABABA
5ena iteració	ABAABABAABAAB
6ena iteració	ABAABABAABAABABAABABA
7ena iteració	ABAABABAABAABABAABABAABAABABAABAAB

Així obtenim un conjunt iterat.

Recordem que una funció real **f** de variable real actua sobre un conjunt de nombres reals anomenat domini i és una funció que assigna a un nombre real **x** (que pertany al domini) un altre nombre real **y = f(x)**.

Provem-ho amb una funció simple:

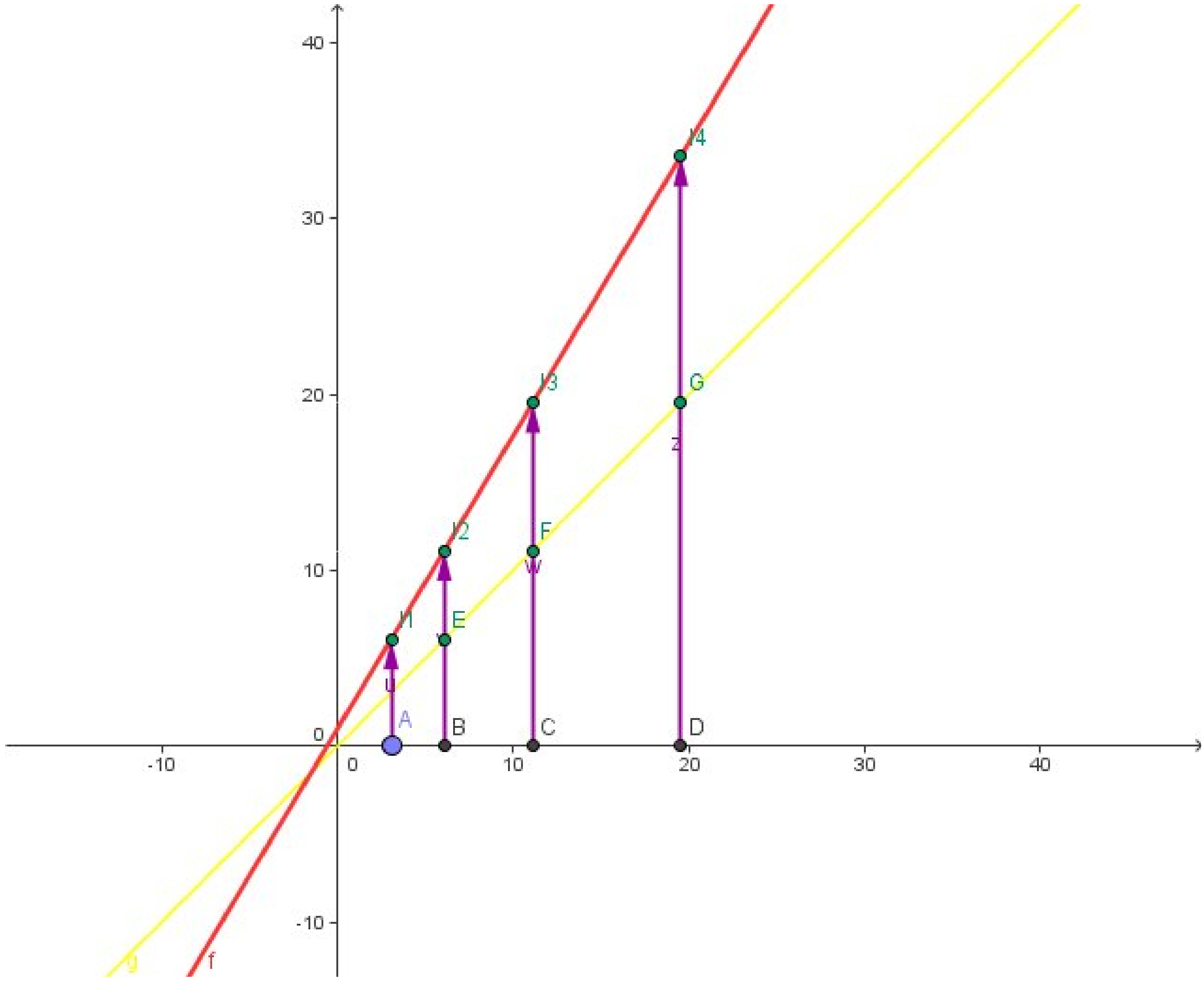
f(x) = 5/3 x + 1

Comencem donant a $\mathbf{x_0}$ el valor **3**.

I l'iterem en la funció **f(x)**.

1era iteració	f(3) ---> 6
2ona iteració	f(6) ---> 11
3era iteració	f(11) ---> 19,33
4rta iteració	f(19,33) ---> 33,22

Podem representar gràficament la funció amb les quatre iteracions anteriors:



La funció f (vermell) és la nostra funció (f(x) = 5/3x + 1), la funció g (groc) és la funció identitat (g(x) = x), el punt A (blau) és el número d'inici (**x_0** = 3), els punts I1,I2,I3,I4 (verds) són les diferents resultats de les iteracions representats a la funció. Els punts B,C,D (grisos) són les representacions sobre l'eix d'abscisses (x) dels diferents resultats de les iteracions sobre els quals tornem a aplicar la funció al iterar.

Observant la gràfica ens trobem que a cada iteració el resultat es va allunyant cada vegada més sobre la recta, és a dir els punts (**I_1**, **I_2**, **I_3** i **I_4**) cada vegada s'allunyen més un de l'anterior.

La funció identitat la representem perquè és la que ens fa de mirall entre l'eix X i Y, permetent-nos traslladar les imatges que van apareixent successivament al iterar un altre cop a l'eix X de partida per tal de tornar a iterar.

Quan apliquem una funció a un número (de l'eix x) la funció ens retorna una imatge (una y) i amb la combinació de les coordenades (x,y) obtingudes podem representar el punt. Quan iterem, transformem la imatge obtinguda en l'operació anterior en una x de manera que puguem tornar-la a introduir dins la funció. La funció identitat (y = x) transforma un número que li introdueixes en el mateix número com a imatge així que la podem fer servir per transformar una "x" en una "y" o viceversa com ens interessi de manera que li podem aplicar de nou la funció (iterar).

Donada una funció s'anomena òrbita d'un punt a la successió formada per aquest punt com a punt inicial i pels punts resultats de les diferents iteracions. Així l'òrbita del 3 és 3, 6, 11, 19.33, 33.22, ... on $x_n = f(x_{n-1})$

Si variem el número inicial o la funció en qüestió podem observar diversos comportaments de l'orbita:

1. Punt fix si la primera imatge coincideix amb el punt d'intersecció de la funció f amb la funció identitat. L'orbita es queda estancada en un punt.
2. Punts periòdics en que dos valors es van alternant.
3. Creixement a infinit: els valors no s'aturen de creixer.
4. Convergència, si la successió té un límit. (La successió tendeix a un número en concret sense arribar-hi mai.

Les funcions iterades i les òrbites tenen un gran importància dins el camp dels fractals i hi són intensament estudiades.

3.2. Nombres complexos

A la ESO i al batxillerat es treballa amb els conjunts de nombres naturals, enters, racionals i reals que com sabem són una successiva ampliació uns dels altres.

Cada una d'aquestes ampliacions neix per la necessitat de disposar de nombres que siguin solució d'equacions.

Així per exemple **$x+4=2$** és una equació de nombres naturals que sols té solució dins els enters.
També l'equació **$2x=-3$** és una equació de nombres enters que només té solució dins els racionals.
I l'equació **$x^2=3$** és una equació de nombres racionals que només té solució dins els reals.
Com havia passat les vegades anteriors, els matemàtics es van trobar amb que l'equació **$x^2=-1$** o la **$x^2+x+1=0$** no tenien solució dins el conjunt de nombres reals.
Van decidir ampliar els nombres reals amb un altre nombre el nombre **i** que es defineix com $\sqrt{-1}$, un nombre inexistent dins els reals. Aquest nombre nou, s'anomena unitat imaginària.

Els complexos (conjunt resultat de la nova ampliació) són nombres compotos per una part real **a** i una altre d'imaginària **bi**, on **a** i **b** són nombres reals i es diferencien per la marca **i** que duu la component imaginària.

Complexos ($\mathbb{C}$)

1. Reals ($\mathbb{R}$)
    - Racionals ($\mathbb{Q}$)
        1. Enters ($\mathbb{Z}$)
            - Naturals ($\mathbb{N}$)
            - Enters negatius
        2. Decimals
            - Exactes
            - Periòdics
    - Irracionals
2. Imaginaris

Hi ha tres maneres diferents de representar la part real i la part imaginaria dels nombres complexos. La primera i la que farem servir (**representació binomial**), representa els nombres complexos com a la suma de les dues components, la real (part real) i la imaginària (part imaginària) del nombre complex. Les altres representacions s'anomenen vectorial i cartesiana són equivalents a les dos nomenclatures utilitzades pels vectors: forma vectorial (polar en els vectors) expressa un mòdul i un angle i en forma cartesiana parelles ordenades (dos components cartesianes en els vectors).

Nosaltres ens centrarem en la seva representació binomial ja que es considera més convenient que les altres dos. Aquesta representació segueix aquest model: **z = a + bi**, on **z** és una nombre complex, **a** és la component real, i **b** és la component imaginària marcada per la **i**.
Formalment:

a = **Re(z)** (part real de **z**)

b = **Im(z)** (part imaginària de **z**)

Com a exemples de nombres complexos podem posar:

6, π, 3+6i, 8-2i o 96i

De moment, la lletra **i** només expressa quina de les dos parts és la part imaginària.
Els nombres complexos es poden representar gràficament en uns eixos cartesians on l'eix d'abscisses representa la component real i l'eix d'ordenades la component imaginaria. Tot número complex, doncs, es pot representar mitjançant un punt del pla i recíprocament. Quan s'estableix aquesta correspondència, es parla de **pla complex** que estudiarem en el següent apartat.

Les operacions amb nombres complexos no són massa diferents de les dels nombres reals ni més complicades però mereixen una consideració:

Per a la suma i la resta, simplement es suma (o resta) la part real amb la part real i la part imaginària amb part imaginària (ignorant el símbol i) i en resulta un altre nombre complex:

- $(x_1 + y_1 i) + (x_2 + y_2 i) = (x_1 + x_2) + (y_1 i + y_2 i)$

  Ex: $(3 - 2i) + (7 + 4i) = 10 + 2i$

La multiplicació és un pel més complicada. S'ha de manejar el símbol **i** com si fos la x de l'àlgebra ordinària:

- $(x_1 + y_1 i) \cdot (x_2 + y_2 i) = (x_1 \cdot x_2 + x_1 \cdot y_2 i + y_1 i \cdot x_2 + y_1 . y_2 i^2)$

Llavors entra en joc la especial propietat de l'element **i** ($i^2 = -1$) i l'expressió s'arregla:

- $(x_1 + y_1 i) \cdot (x_2 + y_2 i) = (x_1 \cdot x_2 - y_1 \cdot y_2 + x_1 \cdot y_2 i + y_1 i \cdot x_2)$

  Ex: $(3-2i) \cdot (7 + 4i) = 21 + 12i - 14i - 8i^2 = 29 - 2i$

Així doncs, podem dir que l'aritmètica dels nombres complexos es idèntica a la dels nombres reals, amb la regla addicional $i^2 = -1$.

Hem dit que tots els nombres reals formen part del grup dels complexos, és a dir que tots els reals són complexos. Doncs noteu que, si sumem $(x_1 + 0i)$ i $(x_2 + 0i)$ dos nombres complexos amb part imaginària igual a zero, ens donarà $(x_1 + x_2 + 0i)$ i si els multipliquem ens donarà $(x_1 \cdot x_2 + 0i)$. Per tant, podem associar el nombre complex (x, 0i) amb el conegut nombre real x i tots els nombres reals com a nombres complexos amb part imaginària zero.

El conjunt dels nombres complexos no és ordenat com estem acostumats en el conjunt dels reals. Així, $z_1 < z_2$ (la lletra z s'utilitza sovint per a simbolitzar nombres complexos) només té sentit si z_1 i z_2 pertanyen al conjunt dels nombres reals ($z_1, z_2 \in \mathbb{R}$).

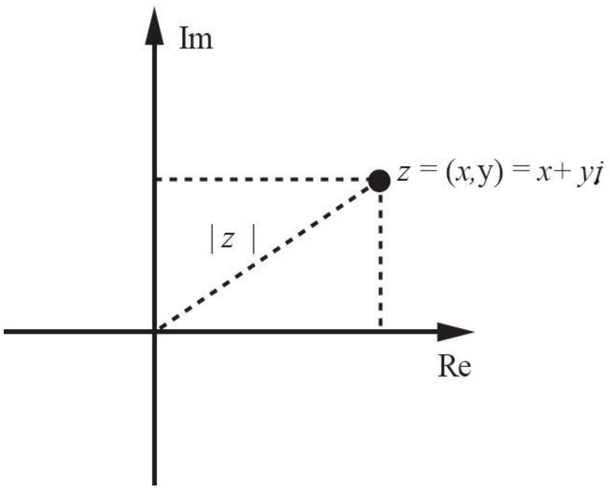
Els números complexos són objecte d'estudis i una eina molt utilitzada avui en dia per a recursos físics o químics que utilitzen les matemàtiques complexes en el seu desenvolupament.

3.3. El pla complex

Els nombres complexos estan formats, com s'ha vist, per dos parts una real i l'altre imaginària. És fàcil que un nombre complex ens recordi a un vector en el pla degut a la seva duplicitat comuna. I tal com els vectors, el números complexos també es poden representar en un pla: en dos eixos cartesians.

Així tot número complex es pot representar mitjançant un punt del pla i recíprocament a qualsevol punt del pla se li pot atribuir un número complex.

Aquesta correspondència entre el pla i el conjunt de nombres fa que neixi un nou concepte, el **pla complex,** que és aquell on l'eix d'abscisses representa la component real i l'eix d'ordenades la component imaginaria del conjunt de nombres compostos.



El nombre complex **z = x + yi** es correspon en el pla complex (o pla R_2) amb el punt de coordenades (x, y), i es pot representar per un vector amb origen en el punt (0, 0) i extrem en el punt (x, y)
En aquesta representació geomètrica podem anomenar els eixos: eix real i eix imaginari (abscisses i ordenades respectivament).

El mòdul d'un nombre complex és el mateix que el mòdul del vector corresponent.

$$|z|=|a+bi|=|\overrightarrow{(a\,,b)}|=\sqrt{a^2+b^2}$$

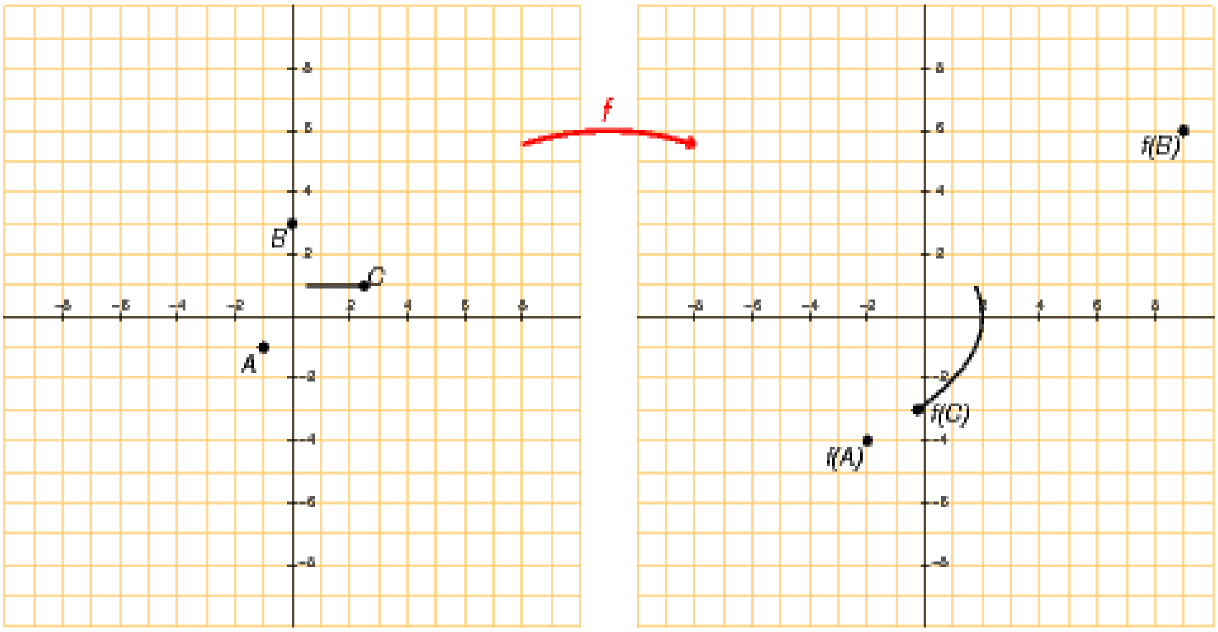
Hem dit que els nombres complexos no es poden ordenar. Però cada nombre complex té un mòdul que es pot entendre com la mida del nombre, per tant podem dir que un nombre complex és més gran que un altre segons si el seu mòdul és major o menor.

3.4. Funcions de variables complexa

Si pensem en una funció de variable real, recordarem que **f** (funció de variable real), actua sobre un conjunt de nombres reals (domini: D) i és una funció que assigna a un nombre real **x** (que pertany al domini) un altre nombre real **y = f(x)** (imatge).
Aquest concepte es pot generalitzar fàcilment per a les funcions de variable complexa:
Una funció complexa de variable complexa f actua sobre un conjunt definit de nombres complexos (domini: D) i és una funció que assigna a cada nombre complex **z** (del domini) un altre nombre complex (imatge) **w = f(z)**.
El conjunt **D** es diu, igual que en el cas de les funcions reals, domini de **f**. Igualment, també parlem del conjunt de les imatges de **f** o recorregut (**R**) igual que en les funcions de variable real.

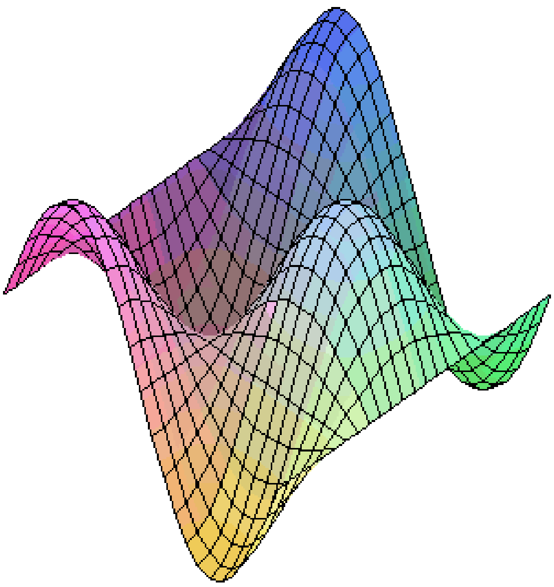
S'ha explicat que es pot representar tot nombre complex amb un punt del pla complex però, com ens ho farem per a representar una funció complexa? Quan treballem amb nombres reals i no complexos, cada un d'ells es pot representar per un punt d'una recta (1 dimensió) i les funcions es representen en dos dimensions ja que és necessari representar a la vegada la **x** i la **y** (cada número amb la seva imatge).
Però en el cas dels números complexos, cada un d'ells es representa amb un punt del pla (dos dimensions) i també necessitem expressar sobre la gràfica la relació de cada nombre amb la seva imatge, un altre nombre complex. Així que necessitaríem dos plans complexos lligats o relacionats entre si per reproduir les funcions de variable complexa necessitaríem **un espai de quatre dimensions**. El que fa que es plantegi un problema.
Hi ha diverses solucions emprades per a solucionar aquest problema; La primera i la més senzilla és fer servir **dos plans complexos** per a representar una sola funció: **un per al domini** (les successives **z** que vas introduint dins la funció) i **un altre per al recorregut** (les imatges (**w**) obtingudes com a resultat d'haver introduït una **z** en la funció). Aquests dos plans també es poden representar superposats. Cal pensar en alguna manera per poder relacionar cada número amb la seva imatge i viceversa. I en el cas de ser plans sobreposats, alguna manera de diferenciar els valors del domini i els del recorregut. Per exemple, amb una combinació de formes dels punts, utilitzant diferents colors...

Aquí tenim un exemple d'una funció **f** de la qual s'han representat dos punts i un segment. El primer pla és el del domini i s'hi poden veure els dos punts (**A** i **B**) (nombres complexos) escollits per a buscar-ne la imatge i el segment dels punts del qual es veuen representades les imatges en el segon pla on també es poden veure les imatges d'**A** i **B**.



Aquest exemple, relaciona els dos punts i el segment amb les seves imatges indicant-ho mitjançant lletres però si augmentes el nombre de punts a representar aquest recurs deixa de ser còmode.

També se solen fer representacions en tres dimensions introduint el canvi de color per tal de suggerir la quarta coordenada o animacions en tres dimensions per tal de representar les quatre dimensions.



L'estudi de les funcions de variable complexa se'l coneix com anàlisis complexa i té una gran quantitat d'usos com a eina tant dins les pròpies matemàtiques com en altres branques de les ciències.

3.5. Iteració amb variable complexa

Ja sabem que tot fractal neix a partir d'un conjunt d'iteracions i hem vist com funcionen les iteracions en una funció amb variable real.
Hi ha una classe de fractals que anomenem els fractals del pla complex que s'aconsegueixen iterant una funció de variable complexa en el pla complex.
Recordem que en la iteració de les funcions reals, introduíem un número en la funció i tornàvem a introduir dins la mateixa funció el resultat de l'operació anterior.
Iterar una funció de variable complexa no és pas més difícil si s'està avesat a operar amb nombres complexos.
Posem un exemple:

Tenim una funció complexa:

f(z) = 2z + 5-3i

Comencem amb el nombre complex:

2+7i

I iterem:

f(**2+7i**) = 2(2+7i) + 5-3i = 4+14i + 5-3i = **9+11i**

f(**9+11i**) = 2(9+11i) + 5-3i = 18+22i + 5-3i = **23+19i**

f(**23+19i**) = 2(23+19i) + 5-3i = 46+38i + 5-3i = **51+35i**

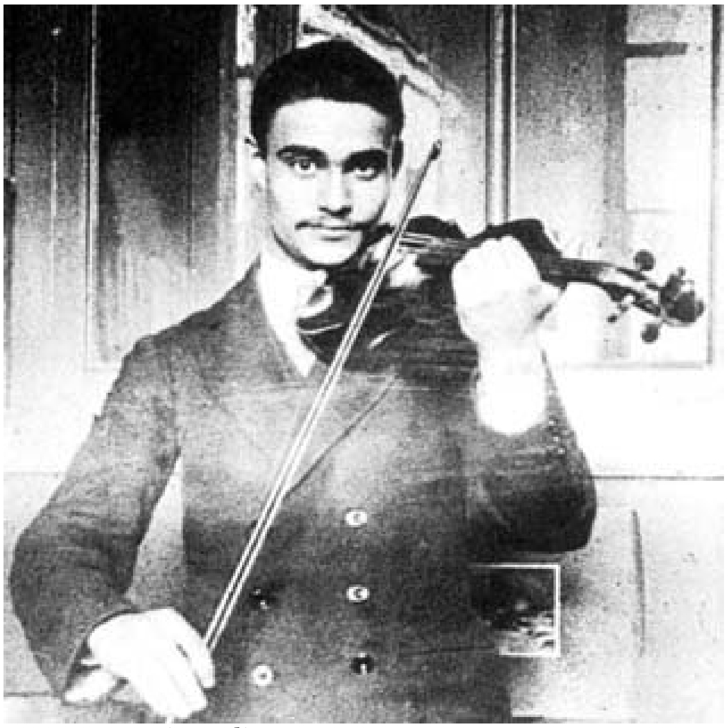
f(**51+35i**) = 2(51+35i) + 5-3i = 102+70i + 5-3i = **107+67i**

f(**107+67i**) = 2(107+67i) + 5-3i = 214+134i + 5-3i = **219+131i**

f(**219+131i**) = 2(219+131i) + 5-3i = 438+262i + 5-3i = **443+259i**

Amb la iteració al pla complex es creen molts gràfics fractals que són els anomenats fractals del pla complex. Entre ells es troben els conjunts de Julia i el conjunt de Mandelbrot peces fonamentals en el món dels fractals.
La manera més comuna de representar fractals del pla complex fruits d'iteració d'una funció complexa, és mitjançant un pla on cada nombre té els seu color que varia segons el comportament (distribució, punts fixos...) de l'òrbita (successió de nombres resultants de les iteracions).
Hi ha moltes fórmules diferents i molts mètodes utilitzats. En descriurem dos, els més comuns; el conjunt de Mandelbrot i els conjunts de Júlia.

3.6. El conjunt de Julia


Gaston Maurice Julia

Gaston Maurice Julia (**1893-1978**) va ser un matemàtic francès que va treballar en el camp de la dinàmica complexa i és el pare d'una família molt gran de fractals anomenats conjunts de Julia.

Un conjunt de Julia $\mathbf{J_c}$ és un fractal definit al pla complex que es basa en la iteració de la fórmula **$f(z) = z^2 + c$**. On **c** és un nombre complex que fa de paràmetre fix i específic per a cada conjunt de Julia (canvia per a cada conjunt). Hi ha tants conjunts de Julia com nombres complexos en el pla complex ja que per cada nombre complex **c**, es defineix una conjunt de Julia $\mathbf{J_c}$ diferent.
Z és la variable on introduirem les diferents iteracions.
La successió $|z_0|, |z_1|, |z_2|, |z_3|, \ldots$ que és anomenada l'òrbita de $\mathbf{z_0}$ pot ser acotada o anar creixent cap a infinit.
$\mathbf{Z_0}$, el primer valor de l'orbita, serà un nombre complex que representarà un punt del pla que pintarem d'un color o un altre segons el comportament de la resta de l'orbita.

El conjunt de Julia **Jc** , on **c** és un complex, es defineix com el conjunt de nombres que iterats per funció **$f(z) = z^2 + c$** tenen l'orbita acotada .

Si el punt ($\mathbf{z_0}$) no pertany al conjunt **J** es pintarà de blanc. En canvi, si el punt ($\mathbf{z_0}$) pertany al conjunt es pintarà de negre.

Posem un exemple per entendre-ho millor:

Construirem el conjunt de Julia $\mathbf{J_{0,108+0,620i}}$:
Amb fórmula **$f(z) = z^2 + 0{,}108+0{,}620i$** per tant amb la variable
c = (0,108+0,620i).

Hem de iterar tots els nombres complexos però en el nostre exemple, agafarem, de moment, el nombre **1+i** com a $\mathbf{z_0}$.

I iterarem la funció:

f(1+i) = $(1+i)^2+0{,}108+0{,}620i = 1-2i+i^2+0{,}108+0{,}620i = 0{,}108-1{,}38i$

$|0{,}108-1{,}38i|$ = **1,384**

f(0,108-1,38i) = $(0{,}108-1{,}38i)^2+0{,}108+0{,}620i = 0{,}011664-0{,}13392i+1{,}9044i^2+0{,}108+0{,}620i = -1{,}784736+0{,}48608i$

$|-1{,}784736+0{,}48608i|$ = **1,849**

f(-1,784736+0,48608i) = $(-1{,}784736+0{,}48608i)^2+0{,}108+0{,}620i = 3{,}1852825-1{,}735048950i+0{,}236273766i^2+0{,}108+0{,}620i = 3{,}057008734-1{,}115048950i$

$|3{,}057008734-1{,}115048950i|$ = **3,254**

L'òrbita resultant d'iterar $\mathbf{z_0}$ està formada per nombres complexos amb mòdul creixent, ho sigui que tendeix a infinit. Per tant **1+i** no pertany al conjunt de Julia en qüestió.

Podríem intentar trobar un punt dins del conjunt de manera que així puguem veure el comportament de l'òrbita.

Tornem a iterar la funció $\mathbf{J_{0,108+0,620i}}$ amb un nombre inicial diferent, el punt **0+0i** (el punt real 0) com a $\mathbf{z_0}$

f(0+0i) = $(0+0i)^2+0{,}108+0{,}620i = 0{,}108+0{,}620i$

$|0{,}108+0{,}620i|$ = **0,629**

f(0,108+0,620i) = $(0{,}108+0{,}620i)^2+0{,}108+0{,}620i = 0{,}011664+0{,}13392i+0{,}3844i^2+0{,}108+0{,}620i = -0{,}264736+0{,}75392i$

$|-0{,}264736+0{,}75392i|$ = **0,799**

f(-0,264736+0,75392i) = $(-0{,}264736+0{,}75392i)^2+0{,}108+0{,}620i = 0{,}070085149-0{,}39927953i+0{,}568395366i^2+0{,}108+0{,}620i = -0{,}390310217+0{,}22072047i$

$|-0{,}390310217+0{,}22072047i|$ = **0,448**

f(-0,390310217+0,22072047i) = $(-0{,}390310217+0{,}22072047i)^2+0{,}108+0{,}620i = 0{,}152342065-0{,}172298909i+0{,}048717525i^2+0{,}108+0{,}620i = 0{,}21162454+0{,}447701091i$

$|0{,}21162454+0{,}447701091i|$ = **0,144**

f(0,21162454+0,447701091i) =

(0,21162454+0,447701091i)^2+0,108+0,620i =

0,044784945+0,189489074i+0,200436266i^2+0,108+0,620i = -0,047651321 +0,809489074i

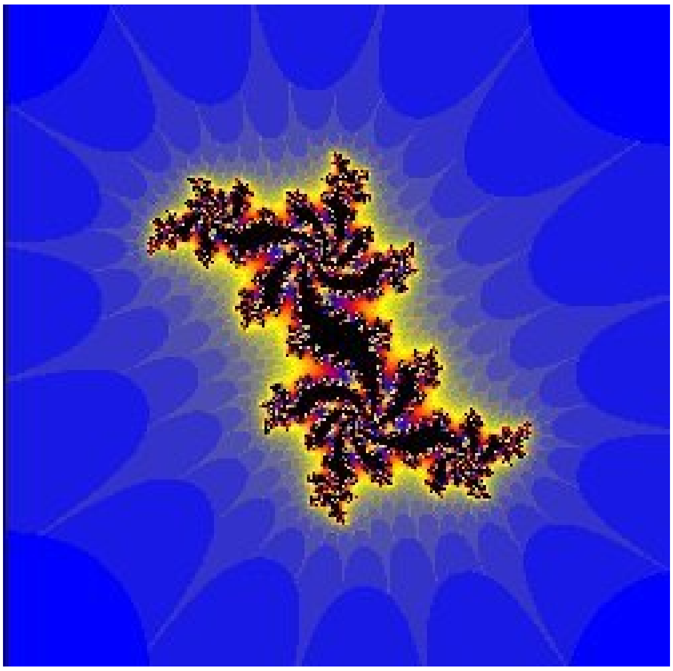
|-0,047651321+0,809489074i| = **0,81**

f(-0,047651321+0,809489074i) = (-0,047651321 +0,809489074i)^2+0,108+0,620i = 0,002270648-0,077146447i+0,655272561i^2+0,108+0,620i = -0,545001913+0,542853553i

|-0,545001913+0,542853553i| = **0,769**

Hauríem de fer moltes més iteracions per comprovar que de veritat el punt pertany al conjunt de Julia però, amb aquestes que tenim fetes en l'exemple, podem veure que el mòdul no passa de 1, es queda restringit i no va augmentant. Per tant, i per la informació que en tenim "externa", podem afirmar que el punt **0+0i** pertany al conjunt de Julia $\mathbf{J_{0,108+0,620i}}$.

El conjunt de Julia $\mathbf{J_{0,108+0,620i}}$ que hem fet servir per a fer l'exemple, té aquesta imatge: Els eixos, que no hi són representats, dividirien la imatge en 4 parts iguals.

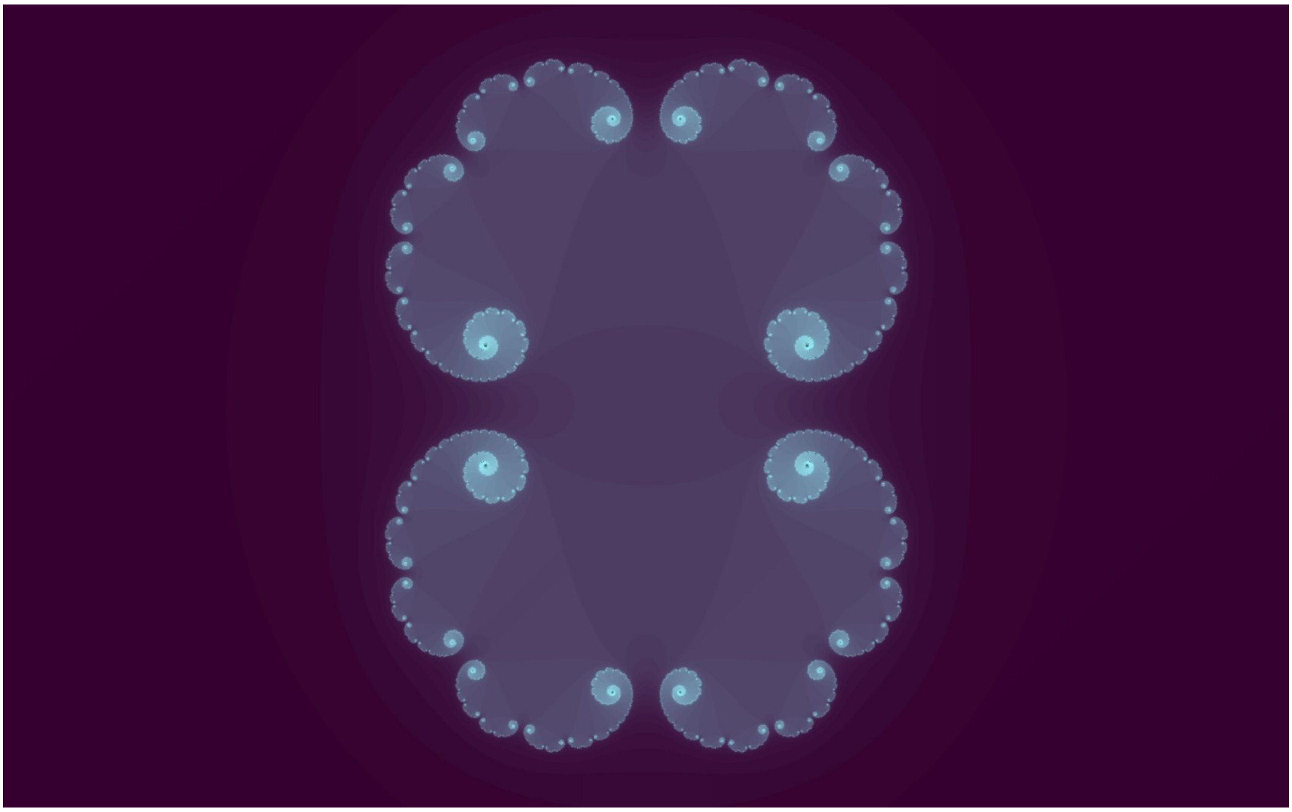


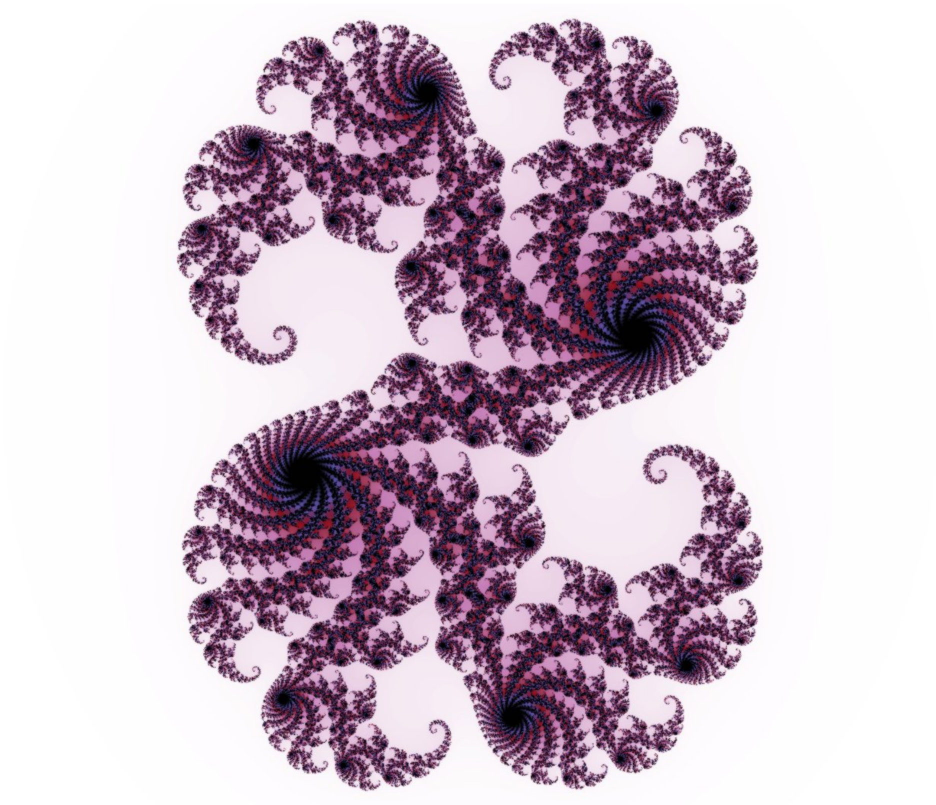
Imatge extreta de : <**http://brain.cc.kogakuin.ac.jp/~kanamaru/Chaos/e/Julia/**>

L'operació s'haurà de repetir per tots els punts del pla complex tot iniciant la iteració amb

els diferents nombres complexos que representen els diferents punts.
Així en els nostres exemples anteriors, pintaríem el punt **1+i** de color blanc ja que la seva òrbita tendeix a infinit i no pertany a aquest conjunt de Julia i de color negre al punt **0+0i** ja que l'òrbita es queda acotada i pertany al conjunt de Julia en qüestió.

Recentment, s'han utilitzat més colors per a representar els conjunts de Julia potser amb una finalitat estètica. Els colors diferents a negre, representen la velocitat amb que la successió de nombres van cap a infinit i per tant tots els punts que no siguin negres no pertanyen al conjunt de Julia. Així el punt **1+i** està pintat de blau en la imatge anterior.
Els conjunts de Julia estan íntimament lligats amb el conjunt de Mandelbrot que explicarem a continuació.
Si pensem en la gran quantitat i diversitat de conjunts de Julia que apareixen variant el valor de c podrem comprendre la gran amplitud de la família de fractals al pla complex definits per Julia.
Això són altres exemples de conjunts de Júlia:





Conjunt de Julia f(z) = z^2 + 0.285+0.013i

3.7. El conjunt de Mandelbrot

El conjunt de Mandelbrot és un fractal que s'obté iterant una família de funcions sobre el conjunt dels números complexos del pla complex. Aquesta família de funcions està formada per totes les funcions del tipus:

$$\mathbf{f(z) = z^2 + c}.$$

On **c** és el nombre complex que estudiem , i **z** és el valor que s'itera amb $\mathbf{z_0}$ sempre igual a zero.

Aquesta fórmula ens pot recordar als conjunts de Julia però hi ha una diferencia fonamental entre uns i l'altre. Fixem-nos en que per a fer un conjunt de Julia el valor que marcava el punt del pla a representar era el valor inicial amb que començàvem a iterar, però en el conjunt de Mandelbrot el valor que ens marca el punt del pla a representar és el paràmetre c que varia cada vegada que canviem de punt i en canvi, la **z** inicial ($\mathbf{z_0}$) sempre és zero. Així, doncs, observem que cada valor de **c** dóna lloc a un polinomi diferent, canvia la fórmula per cada punt del pla complex.

Si iterem aquesta funció per diferents valors de **c** i sempre des de valor inicial zero $\mathbf{z_0 = 0}$ ens adonarem de que per a alguns valors de **c**, l'òrbita dels resultats de les iteracions resta acotada, mentre que per altres valors tendeix a infinit.

El conjunt de Mandelbrot està format per tots els punts **c** del pla complex que tenen una òrbita, fruit dels resultats de les iteracions, que roman acotada, i que gràficament representem en negre. Si l'òrbita d'un punt **c** no roman acotada sinó que tendeix a infinit aquest punt **c** no pertanyen al conjunt de Mandelbrot i en pinten de blanc.
El problema, és que per determinar si una successió de nombres tendeix a infinit o no calen infinites iteracions per tant calen infinits càlculs per determinar si un punt pertany a Mandelbrot o no. Per aquesta raó s'ha creat una condició suplementària.
Només podem fer un nombre limitat de càlculs segons els recursos que tinguem. Així que assignem a cada punt del pla complex un nombre màxim d'iteracions **p**. Si al finalitzar totes les iteracions, el nombre complex resultant de l'última és més petit que **2** (té un mòdul inferior a **2**) llavors acceptem que aquell punt pertany al conjunt de Mandelbrot. En canvi, en el moment en que el resultat d'una de les diferents iteracions sigui més gran que **2** (tingui un mòdul superior a **2**), entendrem que aquell punt no pertany al conjunt de Mandelbrot.
Quan més gran sigui **p** més detallat serà el gràfic i més es podran veure les propietats fractals del conjunt.

En cas de que utilitzem més de dos colors per a representar el conjunt de Mandelbrot, si **c** pertany al conjunt el pintarem de color negre però en cas contrari, li assignem un color diferent segons la iteració en que es comprova que **c** no pertany a Mandelbrot (segons el número de la iteració en que el mòdul del resultat supera **2**).

El resultat de tot aquest complex mètode és una estructura molt complexa, especialment a les fronteres de la figura, on es pot veure fàcilment la propietat d'auto semblança. El conjunt de Mandelbrot és connex (podem anar d'un punt a un altre del conjunt de Mandelbrot mitjançant un camí continuu que passi en tots els seus punts pel conjunt de Mandelbrot)

Aquest conjunt té també una altra propietat suplementària: és el conjunt de valors de **c** pels quals el corresponent conjunt de Julia $\mathbf{J_c}$ és connex.

El conjunt de Mandelbrot i els conjunts de Julia, són l'inici, els pares, d'un gran grup de fractals anomenats fractals al pla complex.
L'estudi d'aquests conjunts té força importància en el terreny de les matemàtiques superiors en el que es coneix com a estudi de **Sistemes dinàmics en variable complexa**.

Darrerament aquest tipus de fractals s'han "popularitzat" degut a la gran complexitat i la atractiva estètica de les imatges creades amb mètodes relativament simples ja que avui en dia hi ha molts programes que els dibuixen.

En aquesta web hi ha un generador de Conjunts de Julia a partir de punts de la gràfica de Mandelbrot (triant **c**) on es veu l'orbita de cada punt de la gràfica del conjunt de Julia creat.
<< **http://math.bu.edu/DYSYS/applets/JuliaIteration.html** >>

4. Fractals formats per la iteració de funcions. L-system

4.1. En què consisteix el sistema, utilitats i orígens.

L'anomenat L-system o sistema d'en Lindenmayer és una gramàtica formal inicialment creada i utilitzada per a descriure formalment el procés de desenvolupament de les plantes. És a dir, és un conjunt de regles i símbols (un procediment formal) per a descriure patrons del creixement de les plantes, com també de colònies de bacteris, etc.
Actualment, però, és a empleat per a generar gràfics fractals de dimensió entre 1 i 2.

El sistema es basa en un procés de creació d'elements successivament més complexos substituint (reemplaçant) una part de l'element simple usant un sistema de regles de substitució.

Aquest llenguatge formal va ser introduït el pel biòleg Hungarià Aristid Lindenmeyer (**1925-1989**) de la Universitat de Utrecht.

Lindenmayer va treballar amb el llevat i va estudiar els patrons de creixement de varies classes d'algues.

L'estructura d'un sistema-L ens porta a l'auto similitud degut al seu caràcter recursiu i per tant facilita la producció de formes tipus fractal.

Els sistemes-L són coneguts també per sistemes-L paramètrics, definits com un conjunt $\mathbf{G}_{(V,\,C,\,I,\,R)}$

On: **V** és un conjunt de símbols o elements que poden ser reemplaçats (**variables**)

C és un conjunt de símbols o elements que es mantenen fixes (**constants**)

I és una cadena de símbols de **V** que constitueix el conjunt inicial del sistema, de l'operació (**inici**)

R és un conjunt de regles o normes que defineixen la manera en la que les variables poden ser reemplaçades per combinacions de constants i d'altres variables.

Així, **R** indica inicialment com s'han de canviar les variables **V** del conjunt inicial **I** per una combinació de més variables **V** i constants **C.**

Les regles **R** s'apliquen iterativament al conjunt a partir del conjunt inicial **I.**

Posem un exemple simple però posant nom als diferents elements per tal d'aclarir els conceptes:

Aquest sistema-L va ser creat per a modelar el creixement de les algues.

V (variables) : **A, B**
C (constants) : **Cap**
I (conjunt inicial) : **A** (la $\mathbf{z_o}$ de l'orbita que ja coneixem)
R (regles) : **(A-->AB)** A on hi ha A hi posem AB
(B-->A) A on hi ha B hi posem A

Quan iniciem aquest procés iterant les variables produïdes en el conjunt de regles, es produeix això:

$\mathbf{R}_{(A)}$ = AB
$\mathbf{R}_{(AB)}$ = ABA
$\mathbf{R}_{(ABA)}$ = ABAAB
$\mathbf{R}_{(ABAAB)}$ = ABAABABA

Un sistema-L és lliure quan cada substitució (aplicació de **R**) afecta solament a un símbol o un conjunt de símbols (variables **V**) independentment dels símbols veïns. Quan això no passa, i la norma a seguir en una substitució depèn també del conjunt veí i no solament del conjunt o el símbol variable **V** en qüestió, es diu que el sistema-L és sensible al context.

Quan hi ha només una norma de substitució (canvi) per a cada variable **V**, es diu que el sistema és determinista. Si hi ha varies opcions de canvi i s'escull cada una d'elles amb una probabilitat determinada, es diu que el sistema-L és estocàstic.
Per a crear imatges mitjançant un sistema-L necessitem que cada símbol (constant o variable) faci referència a elements d'un dibuix.

Un exemple clar de fractal creat per un sistema-L és aquest sistema que tenim a continuació que dóna com a resultat el conjunt de Cantor:

V (variables) : **AB**

C (constants) : **Cap**

I (conjunt inicial) : **A**

R (regles) : **(A-->ABA)** A on hi ha A hi posem ABA

(B-->BBB) A on hi ha B hi posem BBB

Iterant les variables ens les regles R començant pel conjunt inicial, obtenim:

$\mathbf{R}_{(\mathbf{A})}$ = ABA

$\mathbf{R}_{(\mathbf{ABA})}$ = ABABBBABA

$\mathbf{R}_{(\mathbf{ABABBBABA})}$ = ABABBBABABBBBBBBBBABABBBABA

Si, es relaciona la variable A com a dibuixar un pas cap endavant i la B com a avançar cap a endavant (sense dibuixar) obtindrem:

$C_{(A)}$ =

$R_{(A)}$ =

$R_{(ABA)}$ =

R(ABABBBABA) =

R(ABABBBABABBBBBBBBBABABBBABA) =

R(ABABBBABABBBBBBBBBABABBBABABBBBBBBBBBBBBBBBBBBBBBBBBBBABABBBABABBBBBBBBBABABBBABA) =

Repetint infinitament aquest procés obtenim el famós fractal conegut com a Conjunt de Cantor.

Podem posar un altre exemple en el que fem servir alguna constant:

Un exemple també conegut que podreu reconèixer:

V (variables) : **A**

C (constants) : **+ -**

I (conjunt inicial) : **A**

R (regles) : **(A-->A+A-A+A)** A on hi ha A hi posem A+A-A+A

Amb aquestes condicions podem iterar i obtindrem:

$R_{(A)}$ = A+A-A+A

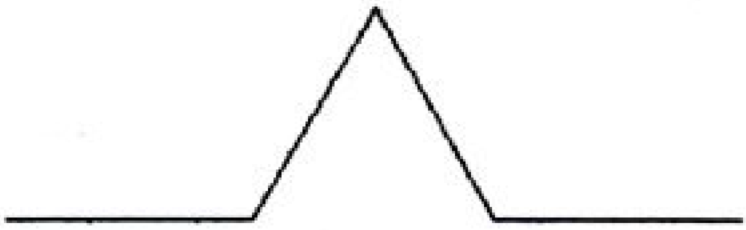
$R_{(A+A-A+A)}$ = A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A

$R_{(A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A)}$ = A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A+A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A+A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A

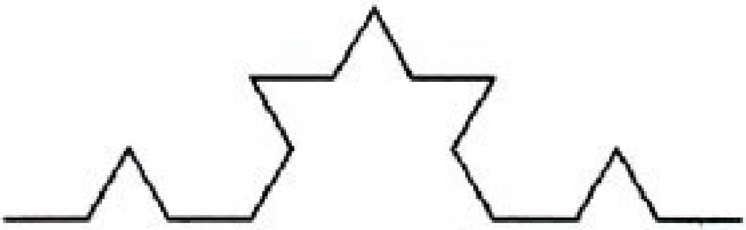
Si relacionem la variable **A** com a dibuixar cap a endavant i les constants **+** i **-** girar 60º cap a la esquerra i girar 120º cap a la dreta respectivament, obtindrem la corba de Koch:

$C_{(A)}$ =

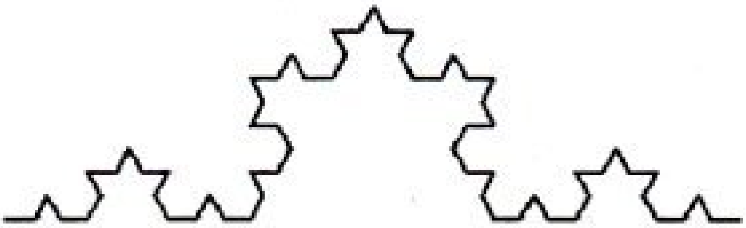
$R_{(A)}$ =



$R_{(A+A-A+A)}$ =

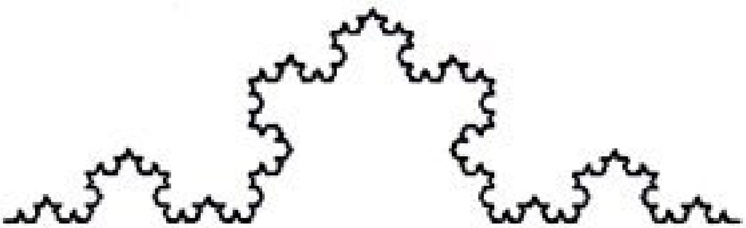


$R_{(A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A)}$ =



$R_{(A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A+A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A-A+A-A+A+A+A-}$

$_{A+A+A+A-A+A+A+A-A+A-A+A-A+A+A+A-A+A)}$ =



El sistema-L és el sistema que farem servir per les pràctiques exposades en el punt 4.3. d'aquest treball.

4.2. El llenguatge Logo

El llenguatge Logo és un llenguatge de programació complet, d'alt nivell, en part funcional i en part estructurat de fàcil aprenentatge pel que és molt útil per a treballar amb nens o persones que s'inicien en la programació.

Va ser dissenyat a finals dels anys 60 i principis dels 70 per Danny Bobrow, Wally Feurzeig i Seymour Papert de l'Institut Tecnològic de Massachussets (MIT) basant-se en la sintaxis del llenguatge Lisp (un llenguatge pel tractament de temes d'intel·ligència artificial).

El llenguatge Logo és, encara, una eina útil per a ensenyar el procés de l'aprenentatge i del pensament. Es pot usar per a ensenyar molts conceptes de lògica, programació i funcionament de les màquines ja que et permet manipular i guiar la màquina. D'aquesta manera, et fa conscient de la forma en que construeixes les teves idees.

Construeix un mètode per dialogar amb l'ordinador, un ambient en el que l'ordinador és el manat i fa el que tu li demanes, entén des de instruccions senzilles fins a instruccions molt més sofisticades i complicades. Està basat en un número relativament petit d'instruccions bàsiques que serveixen per a generar altres instruccions no tant simples que poden constituir al seu torn un programa, es converteixen elles mateixes en part del llenguatge i, alhora, es poden utilitzar per a crear altres instruccions o procediments. En aquest sentit el llenguatge Logo és un llenguatge de procediment.

És un dels pocs llenguatges de programmació que té versions amb instruccions en espanyol.
El llenguatge Logo és molt extens i permet dissenyar produccions molt ambicioses, no està limitat a una matèria específica tot i que s'ha explorat més en l'àmbit de les matemàtiques. És una eina útil per a projectes molt diversos des de estadístiques de tot tipus fins a gràfics molt complexos.

S'ha definit el llenguatge Logo com a amigable i compassiu però també poderós i sobretot extensible i flexible.

L'eina més senzilla i primària del llenguatge Logo és la tortuga. La tortuga és un punt orientat que pot ser guiat des de l'ordinador que es desplaça per la pantalla de l'ordinador seguint les teves instruccions Logo i dibuixant les figures indicades.
Quan els ordinadors personals no permetien el dibuix de gràfics, Logo disposava d'un sistema de traçat de gràfics: la tortuga mecànica. Era un petit robot que era capaç de rebre instruccions simples de Logo i que movent-se per damunt d'un paper arrossegant un llapis, dibuixava gràfics complexos.
Les versions actuals disposen de la possibilitat de controlar múltiples tortugues i s'utilitzen com a eina de simulació de les ciències socials, econòmiques i experimentals .

El llenguatge Logo es va fer molt popular com a filosofia per a l'ensenyament els anys que van seguir a la seva creació però tot i així, no va tenir l'èxit esperat en l'educació.

Per a entendre el funcionament del llenguatge Logo en la pràctica, la millor opció és explicar-ho amb un exemple.
Coneixent unes poques instruccions enteses per l'ordinador és fàcil començar a comprovar que és possible usar la creativitat i la imaginació.

De les diferents versions disponibles del llenguatge de programació Logo hem escollit l'interprete en anglès NetLogo, que es pot descarregar gratuïtament a l'adreça <**http://ccl.northwestern.edu/netlogo/**>. És una versió actualitzada, amb molts exemples ja programats i que a més a més permeten que el resultat dels programes es puguin posar dins de qualsevol pàgina web.

L'exemple que hi ha a continuació està fet seguint el patró de funcionament d'aquest programa.

Intentem dibuixar un quadrat.
Primer de tot, necessitem crear un espai, una tortuga, situar-la i indicar-li que dibuixi una línia per tot allà on passi. Després li farem fer el trajecte.
Crearem una instrucció anomenada ***neteja*** que consistirà en netejar la pantalla del

dibuix anterior i crear una tortuga, de moment.

Així ens queda:

```
to neteja
    clear-all
    create-turtles 1
end
```

Hem creat la instrucció neteja amb la paraula ***to*** i li indiquem les instruccions corresponents a continuació: ***clear-all*** (ho neteja tot), ***create-turtles 1*** (crea el nombre de tortugues indicat) i finalment tanquem la instrucció amb ***end***.

També, afegirem a l'instrucció neteja una ordre que situa a la tortuga el lloc desitjat:

```
to neteja
    clear-all
    create-turtles 1
    ask turtles [    setxy
                         min-pxcor + 5
                         min-pycor + 5  ]
end
```

Li demanem que vagi a la cantonada inferior esquerre amb un marge de 5.

Afegirem també, la instrucció que indica a la tortuga que marqui el camí que faci (***pen-down***):

```
to neteja
    clear-all
    create-turtles 1
    ask turtles [    setxy
                         min-pxcor + 5
                         min-pycor + 5
                     pen-down ]
end
```

Un cop preparat l'escenari comencem a fer el quadrat:

```
to quadrat
    forward 10
    right 90
    forward 10
```

right 90

forward 10

right 90

forward 10

right 90

end

O el que és el mateix:

to quadrat

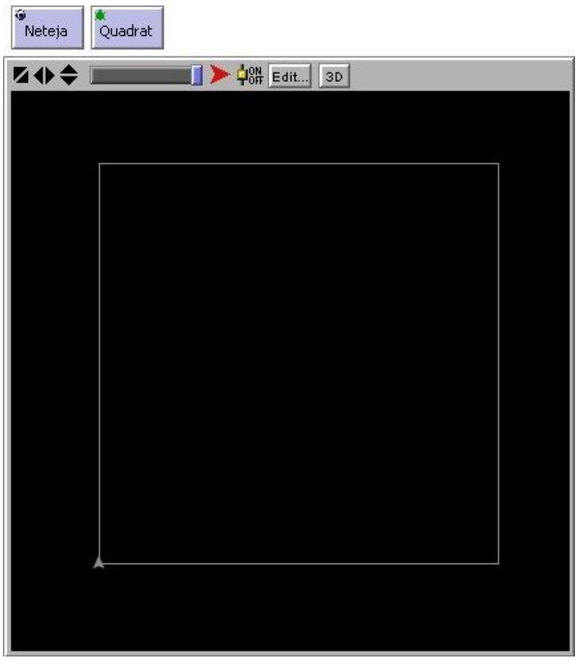
repeat 4 [forward 10

right 90]

end

Així, ja obtenim el nostre quadrat:

```
to neteja
  clear-all
  create-turtles 1
  ask turtles [
     setxy
         min-pxcor + 5
         min-pycor + 5
     pen-down
              ]
end

to quadrat
  forward 25
  right 90
  forward 25
  right 90
  forward 25
  right 90
  forward 25
  right 90
end
```

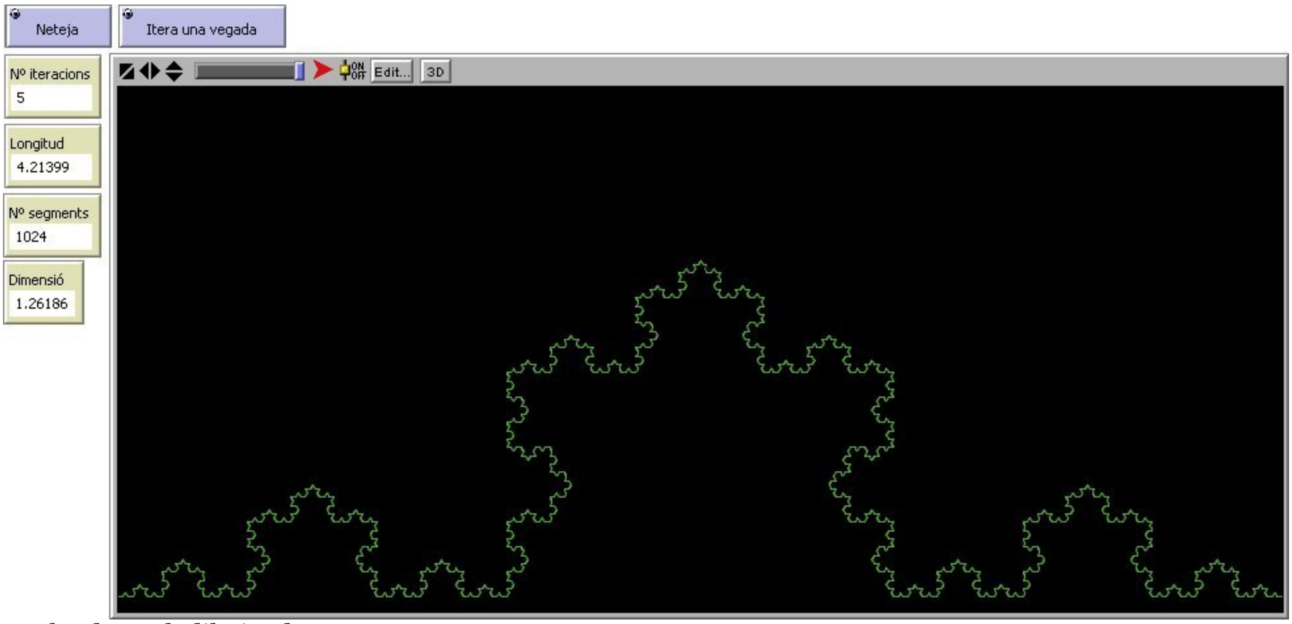


Això és un exemple molt senzill del funcionament del llenguatge Logo. Un cop hi he treballat he comprovat que les seves capacitats i possibilitats són molt grans i toquen moltes matèries.

4.3. Els fractals simples construïts amb Logo

Amb l'ajuda de l'interpret del llenguatge de programació NetLogo, hem reproduït mitjançant el sistema-L una mostra d'alguns dels fractals geomètrics exposats anteriorment.

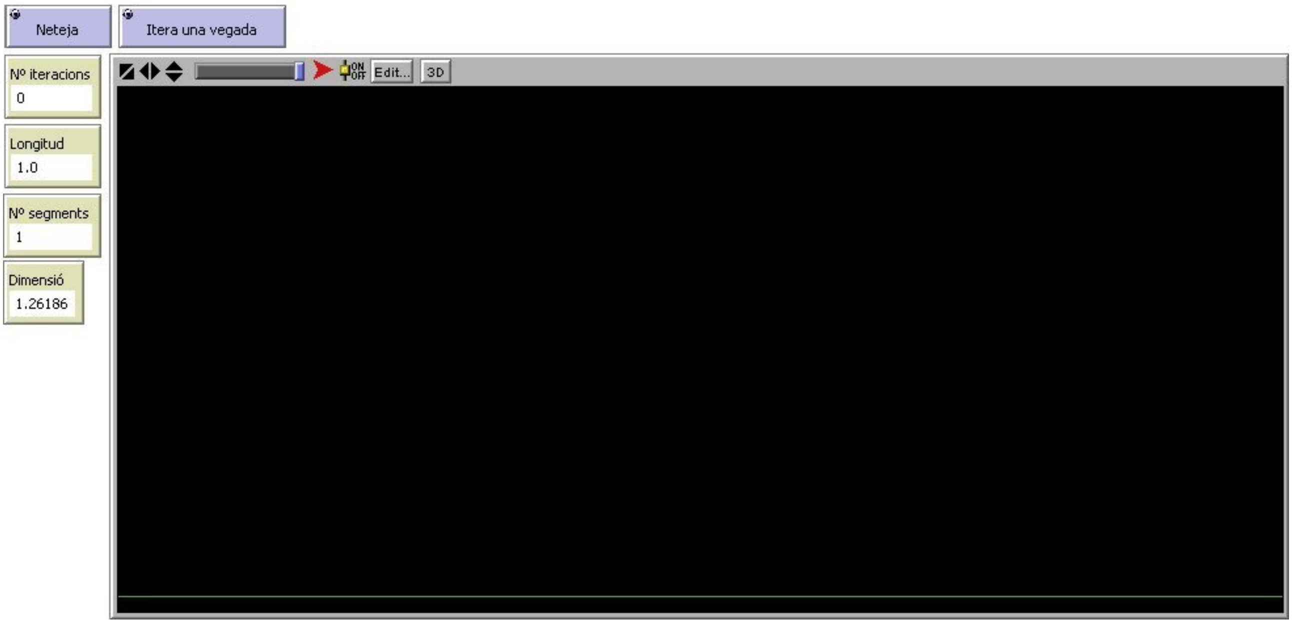
La corba de Kock



Corba de Koch dibuixada per NetLogo

Aquesta és el resultat final de la pràctica en una de les seves fases.
Dins el programa hi ha dos instruccions que es poden accionar amb les botons Neteja (preparació de l'escenari per a començar a iterar) i Itera una vegada (itera la funció en tots els segments presents a la pantalla el aquell moment).
Hem afegit, a més de la imatge i els botons que accionen les diferents opcions (instruccions), una serie de marcadors que indiquen el nombre d'iteracions dutes a terme, la longitud de la corba prenent com a unitat el segment inicial, el nombre de segments de la imatge (valors canviants segons el nombre d'iteracions que hàgim dut a terme) i l'ultim, que indica la dimensió fractal de la corba obtinguda.
Podem veure la Corba de Koch en totes les seves fases d'iteració:

El segment de longitud 1 sobre el qual iterarem la funció.



Veiem com el marcador de les iteracions marca 0, la longitud és de una unitat i hi ha un segment. La dimensió, indica ja la dimensió de la figura per la qual està preparat el programa, aproximadament 1,262.

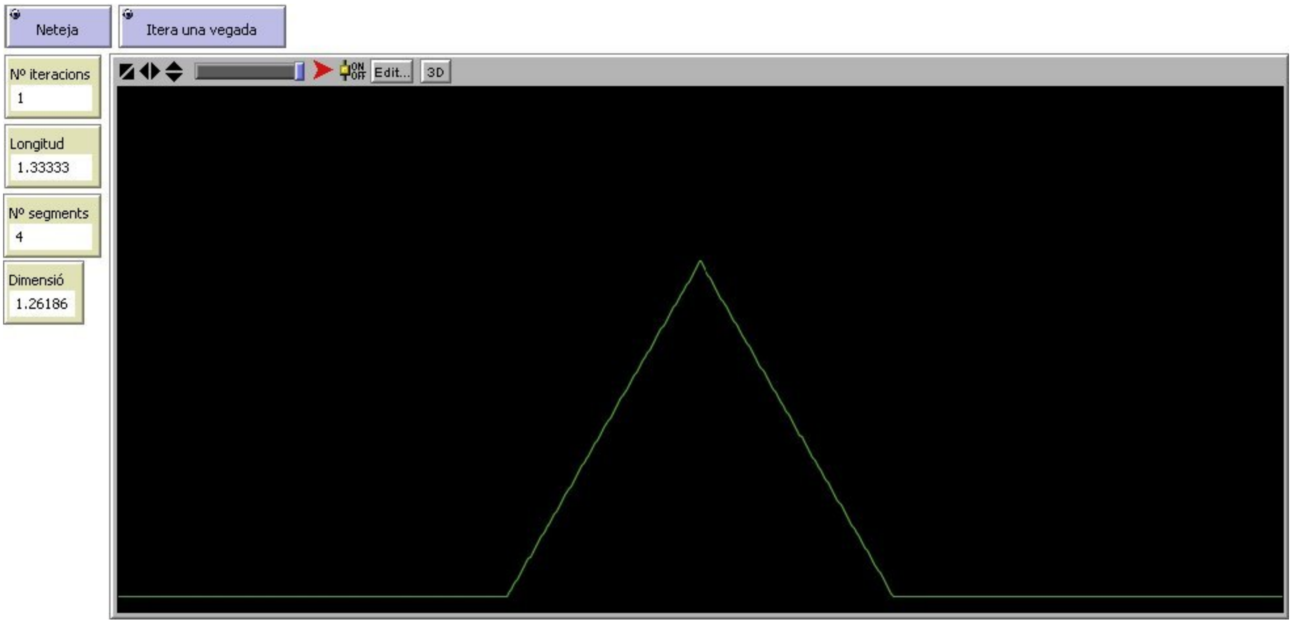
Per a cada marcador hi ha una fórmula o una variable que calcula el que volem expressar, així:

El marcador de les iteracions indica el valor d'una variable que hem inventat per tal de contar cada vegada que es duu a terme la instrucció d'iterar (cada vegada que premem el botó ***Itera una vegada***. El valor inicial és zero i anirà augmentant un punt cada vegada que premem el botó (que iterem).

El marcador de la longitud, indica el valor d'una fórmula: $(4/3)^{iteracions}$ que varia segons varia la variable iteracions.

El nº de segments indica el nombre de tortugues presents en la pantalla ja que cada tortuga dibuixa un segment, i es va renovant constantment cada vegada que aquest nombre varia.

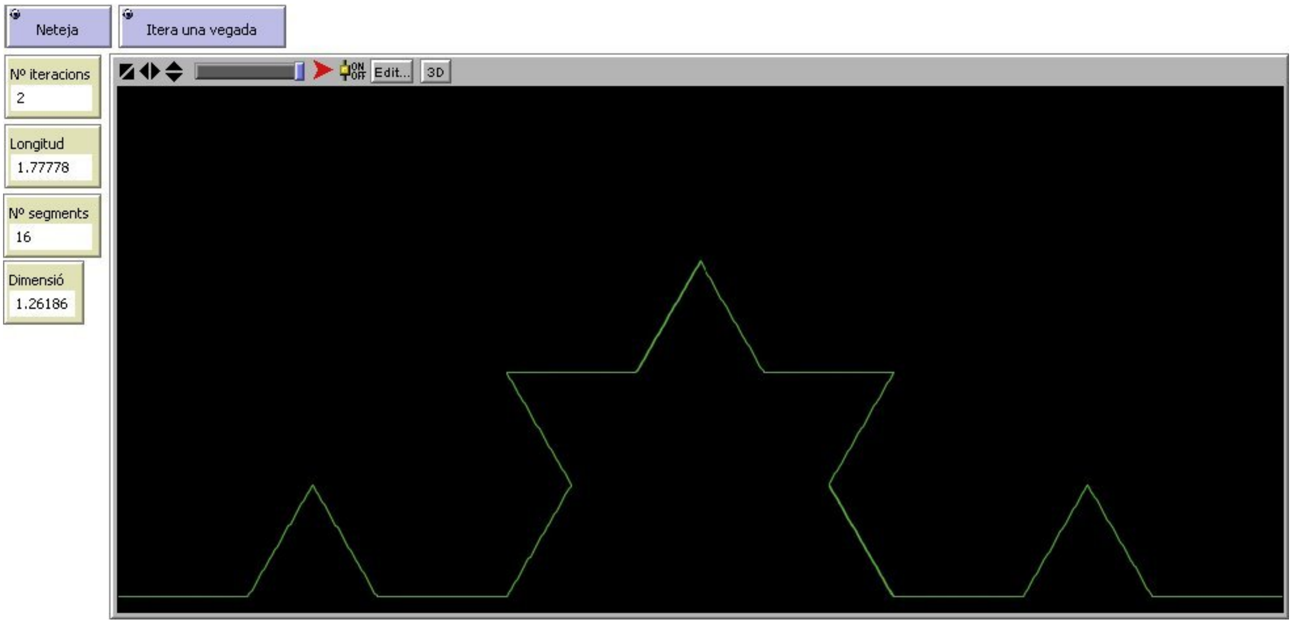
La primera iteració:



Podem veure com el marcador d'iteracions ha pujat i la longitud del segment i el nombre de segments també han augmentat.

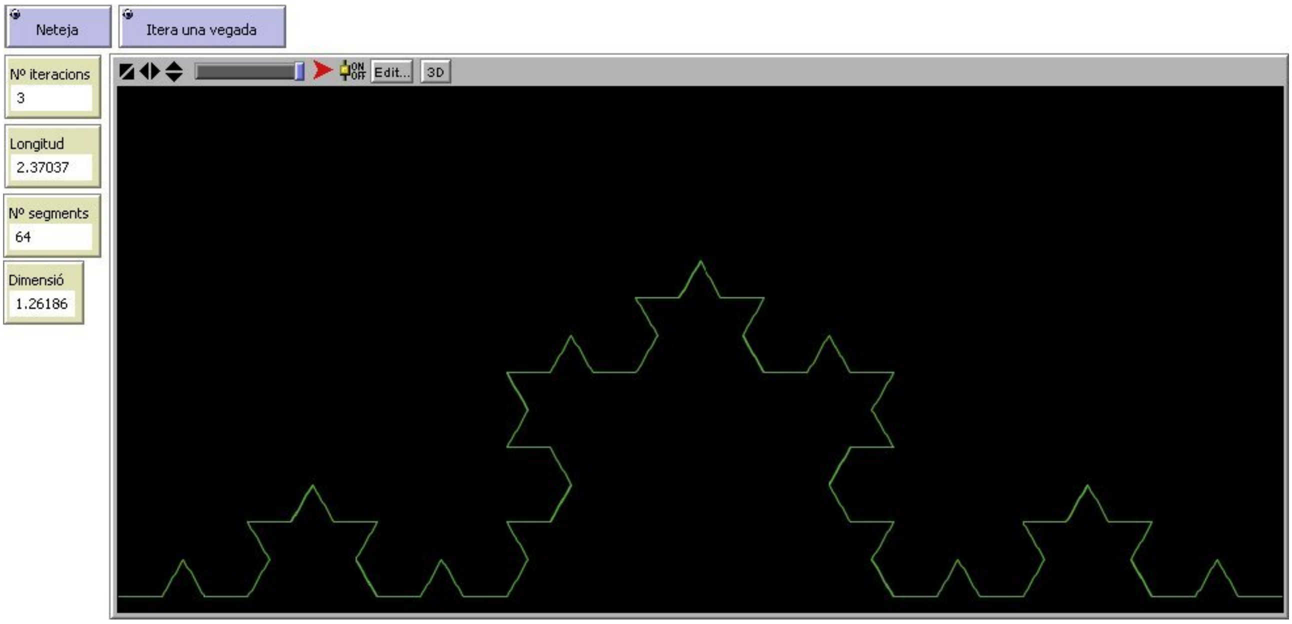
El procediment de la pràctica és que cada "tortuga" col·locada a l'extrem esquerre d'un segment i orientada paral·lelament al segment, dibuixa una corba com la de la imatge de la iteració 1. La mida de la corba varia segons la iteració, cada vegada és un terç més petita. Però, a més a més, cada tortuga prepara el terreny per tal de que es pugui realitzar la següent iteració, és a dir, deixa una tortuga "filla" a l'extrem esquerre de cada segment de manera que quan tornem a iterar aquesta tortuga faci el camí en una mida més reduïda i dibuixi la part de la corba corresponent al seu segment anterior i a la vegada, deixi tortugues preparades per a la següent iteració.

La segona iteració:



Ja comencem a reconèixer la forma característica de la corba.
Les tortugues que a la passada iteració s'han deixat preparades, han fet el seu camí i han dibuixat, cada una d'elles, una corba igual però reduïda a la corba de la imatge de la iteració 1. Alhora, aquestes tortugues han tornat a deixar preparades un seguit de tortugues, una per a cada segment, que a la següent iteració faran, cada una, el seu tros de corba.

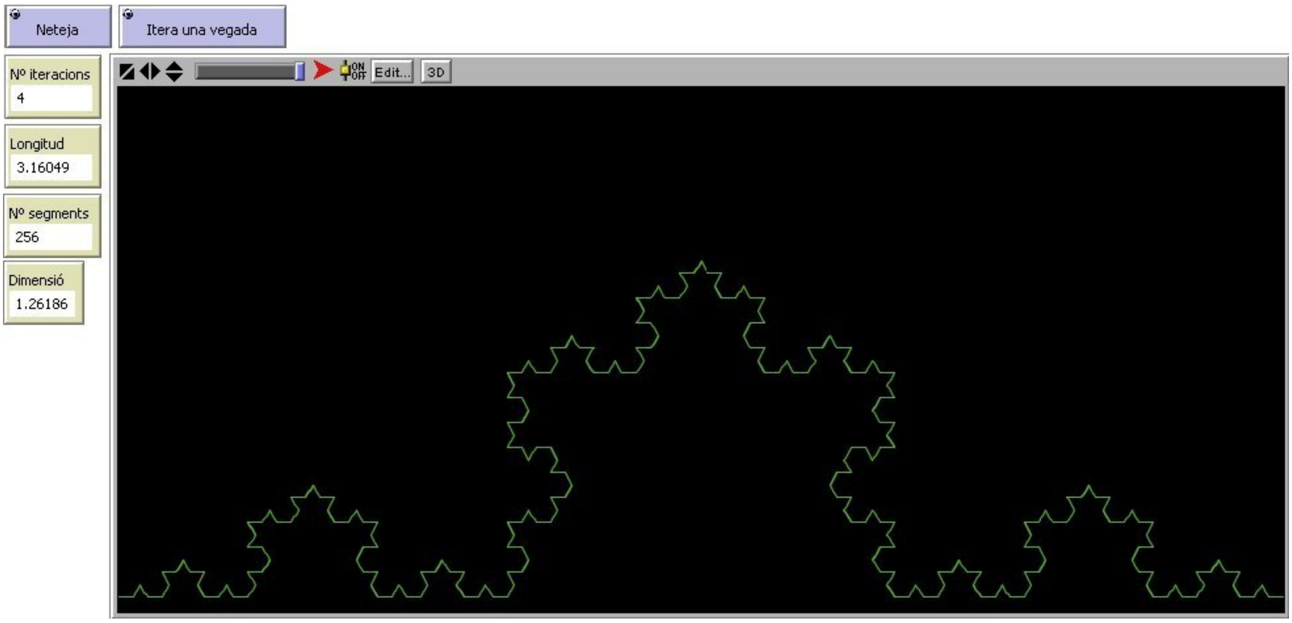
La tercera iteració:



El procés es repeteix i les tortugues abans preparades han tornat a fer el seu camí deixant

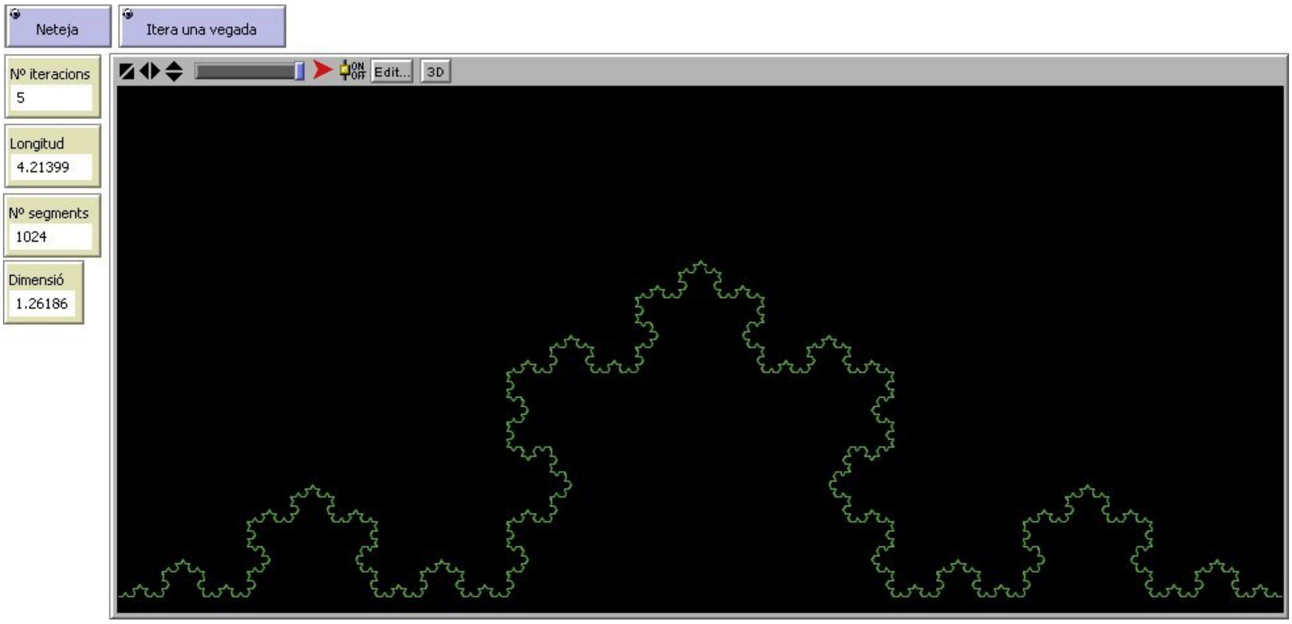
16 trossos semblants a la imatge de la iteració però en una mida molt més reduïda. I han tornat a deixar tortugues.

La quarta iteració:

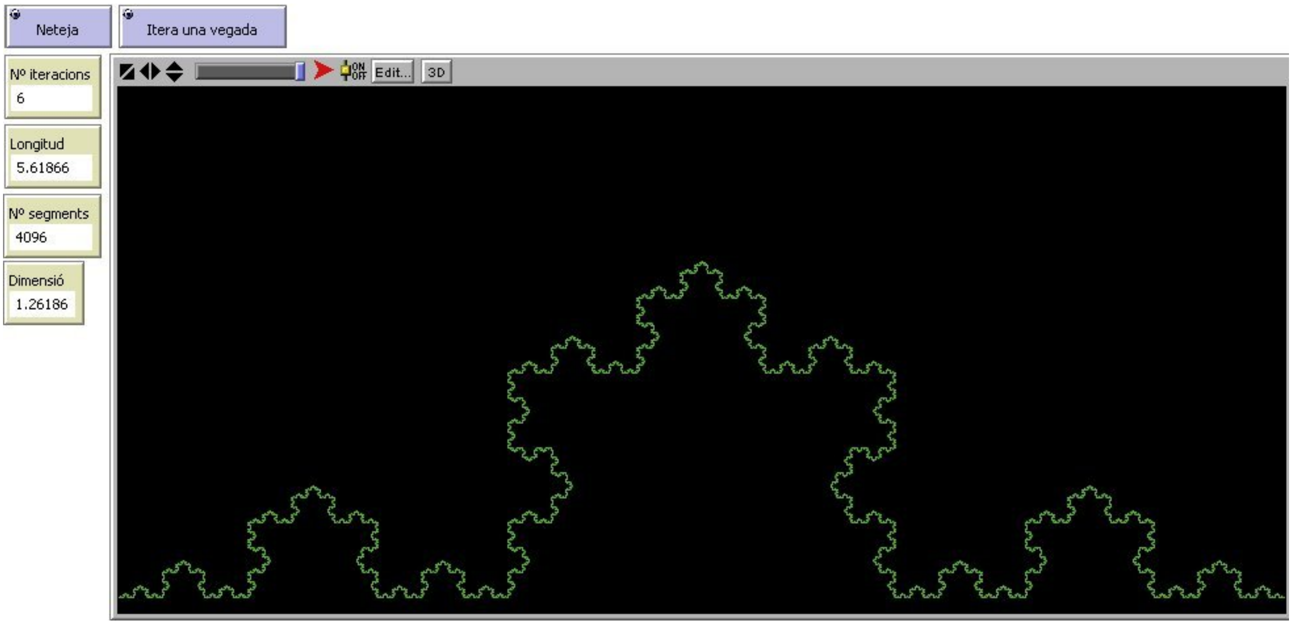


Continua el mateix procés amb una mida molt més petita i moltes més tortugues.

La cinquena iteració:



La sisena iteració:



Veiem com l'apreciació de la imatge ja arriba al seu límit però la corba no s'acaba aquí, ni molt menys, segueix iterant-se indefinidament. I la longitud també augmenta: recordem que la corba de Kock té longitud infinita.

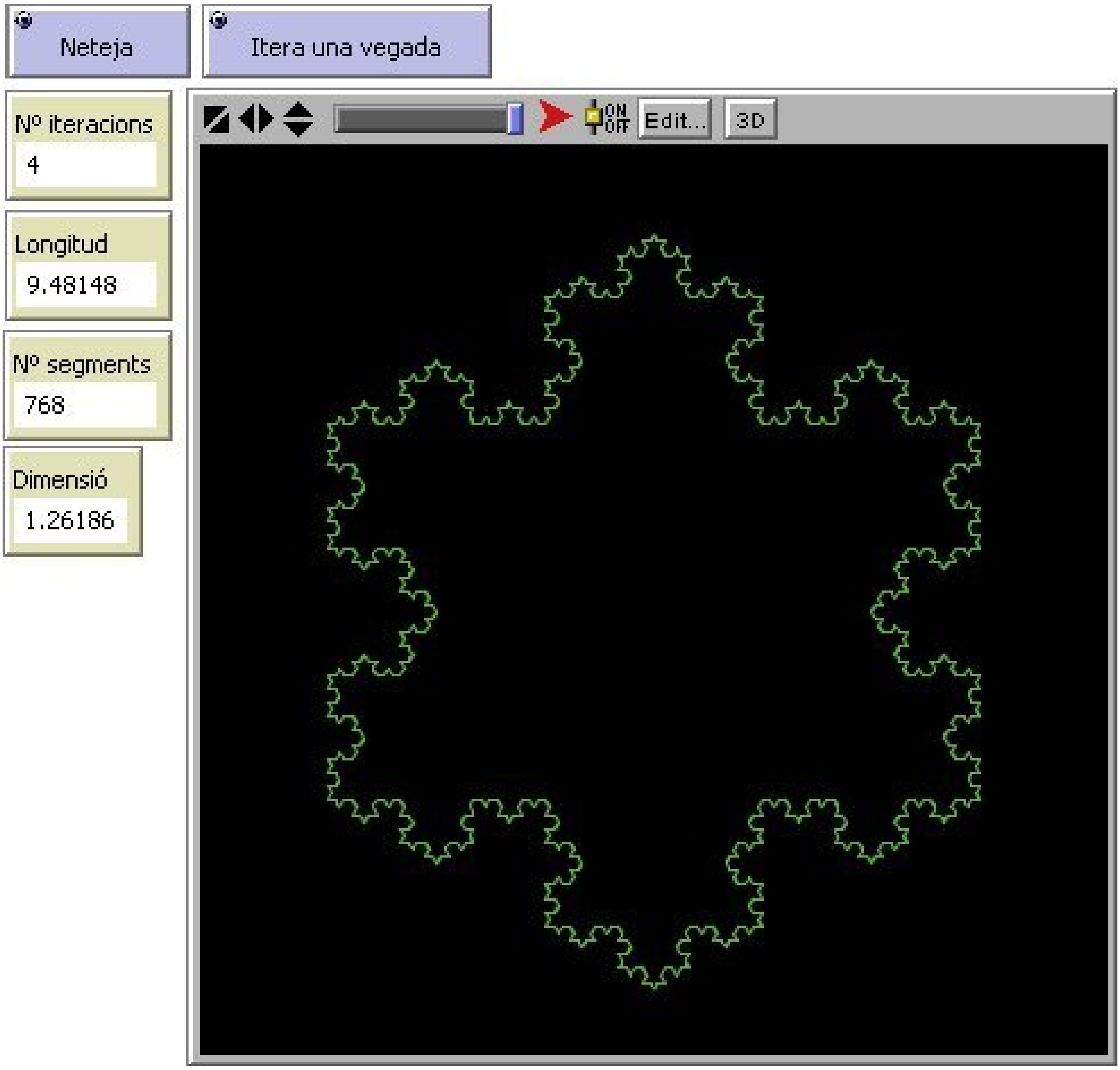
Les instruccions donades al programa per a aconseguir aquestes imatges encadenades són:

```
globals [iteracio mida]

to neteja
        ca
        set iteracio 0
        create-turtles 1
        ask turtles
                [setxy          min-pxcor
                                min-pycor + 5
                set heading 90
                set color green
                set mida max-pxcor * 2
                pen-down
                fd mida
                lt 180
                fd mida
                lt 180
```

```
        hide-turtle  ]
end

to cami
    if iteracio = 6 [stop]
    clear-drawing
    set mida (mida / 3)
    ask turtles
        [itera
        hide-turtle]
    set iteracio iteracio + 1
end


to itera
    hatch 1 [ ]
    fd mida
    lt 60
    hatch 1 [ ]
    fd mida
    rt 120
    hatch 1 [ ]
    fd mida
    lt 60
    fd mida
    lt 180
    fd mida
    lt 180
end
```

Aquestes instruccions són més complexes que les de l'exemple anterior. Utilitzen variables globals com són la **iteracio** que és un comptador de les vegades que s'ha fet **cami** o la **mida** que fem servir per a variar les distàncies segons augmenten les iteracions.
També fem servir més instruccions que formen part del llenguatge Logo. Cada una d'elles requereix un context determinat, efectua una funció diferent.

El floc de neu de Koch



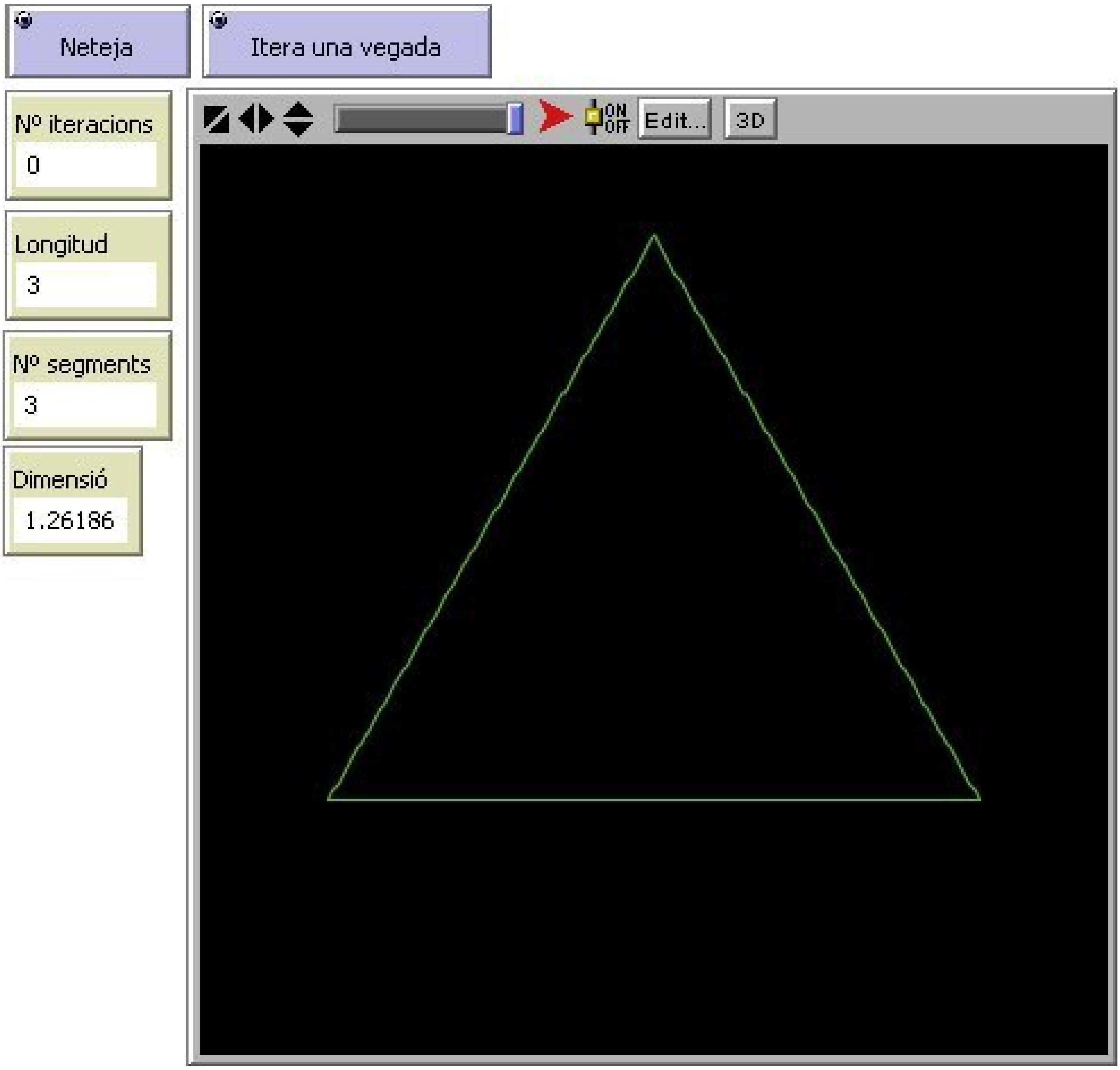
Aquesta és una de les imatges produïdes pel programa en una de les iteracions més avançades.

El floc de neu, com podem veure i recordar, són tres corbes de Koch els segments inicials de les quals són els tres costats d'un triangle equilàter.

Així que el procediment tindrà una part semblant a la de la corba de Koch però hem hagut de adaptar el programa per tal d'aconseguir des de l'anterior programa dissenyat per a la Corba de Koch l'estrella anomenada Floc de neu de Koch.

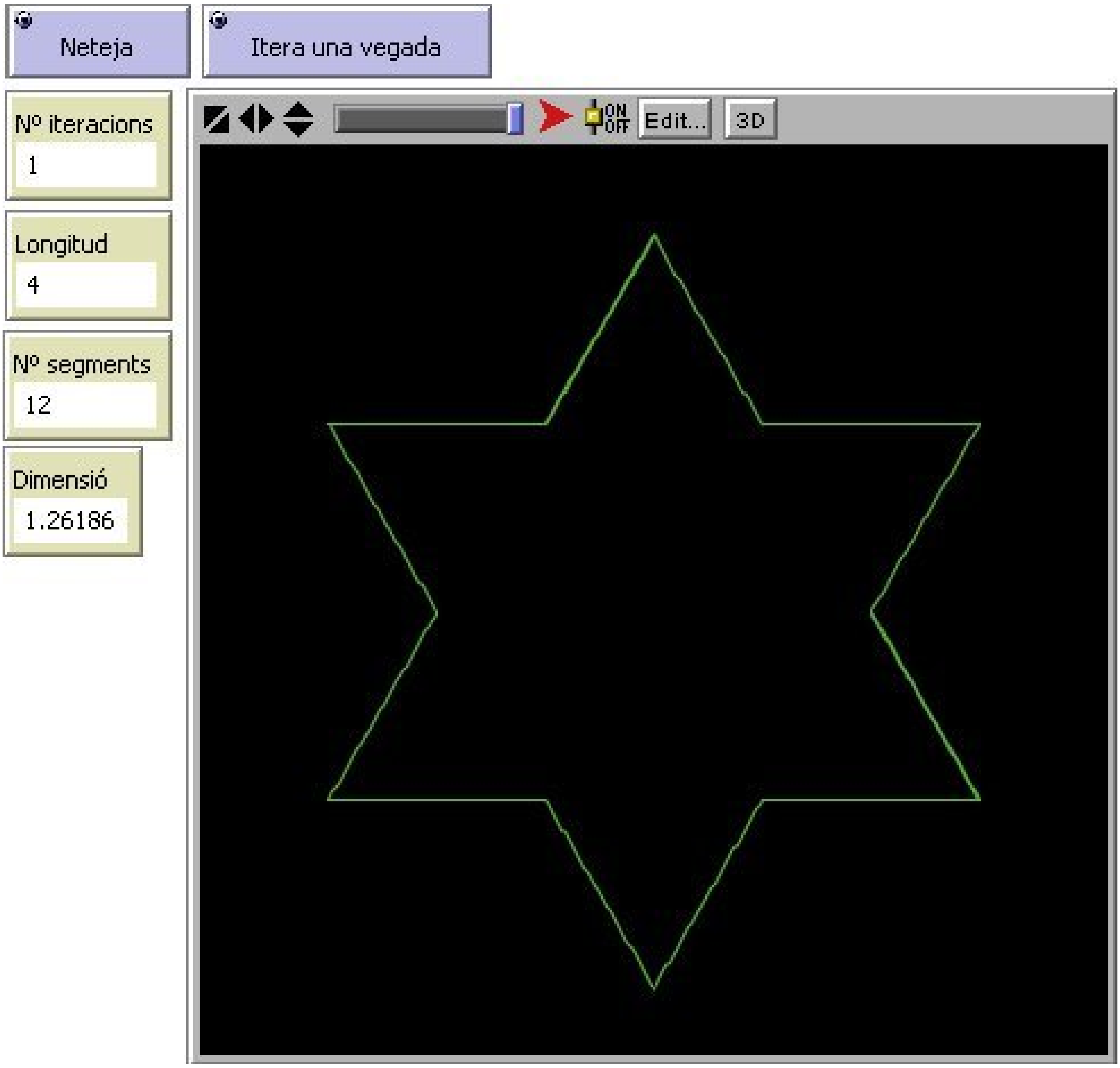
Aquest canvi, es basa bàsicament en crear tres segments en comptes d'un, disposar-los de manera que formin un triangle equilàter i aplicar la iteració de la corba de Koch en cada un d'ells.

Així doncs, a la iteració zero podem veure un triangle equilàter:



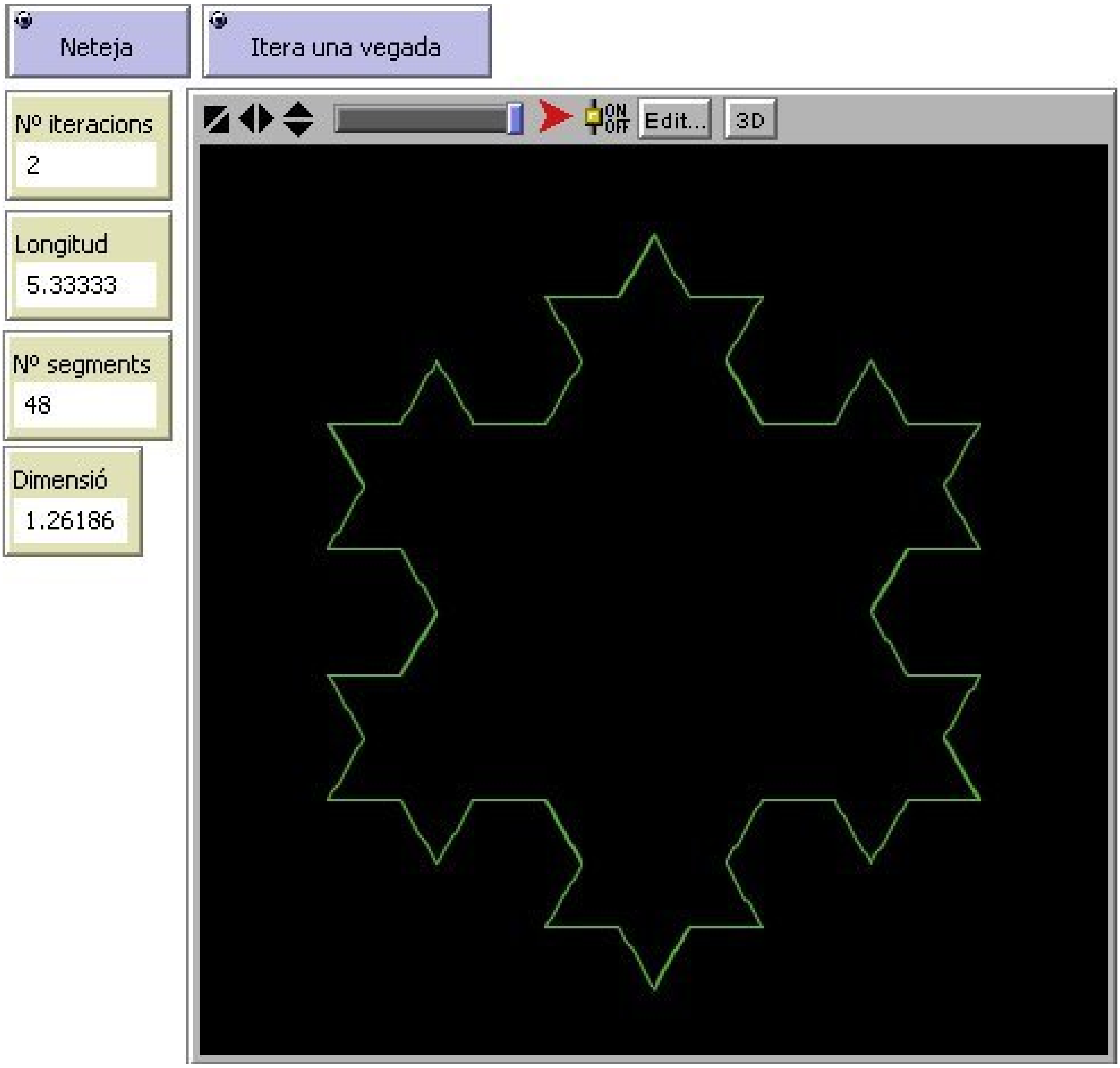
Podem veure que als marcadors, aquesta vegada comencen amb el comptador a 3 ja que cada segment val de longitud 1. El nombre de segments i la iteració aniran pujant com en el programa de la Corba de Koch,

La primera iteració:

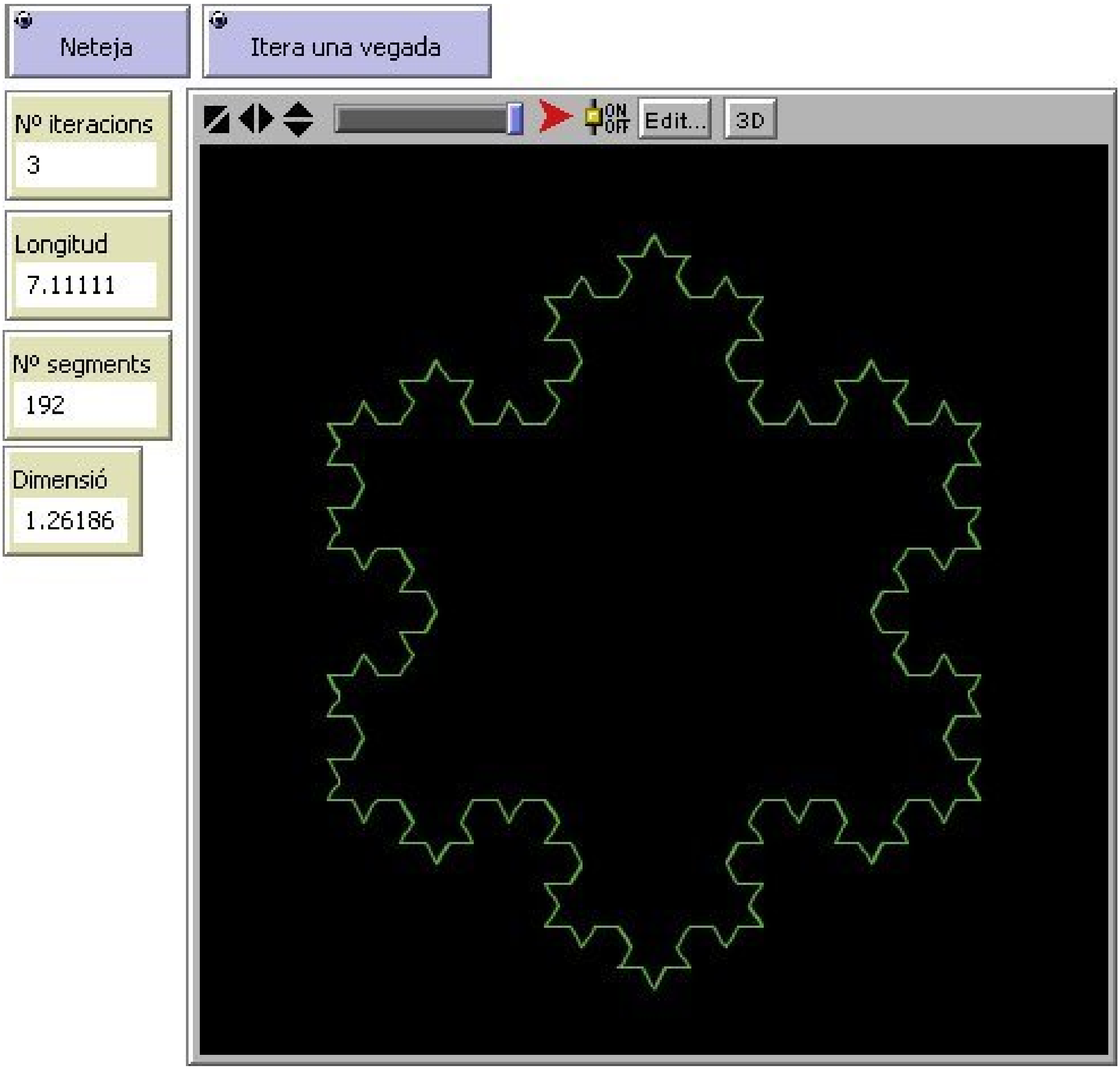


Tenim l'estrella de sis puntes i els comptadors han variat.

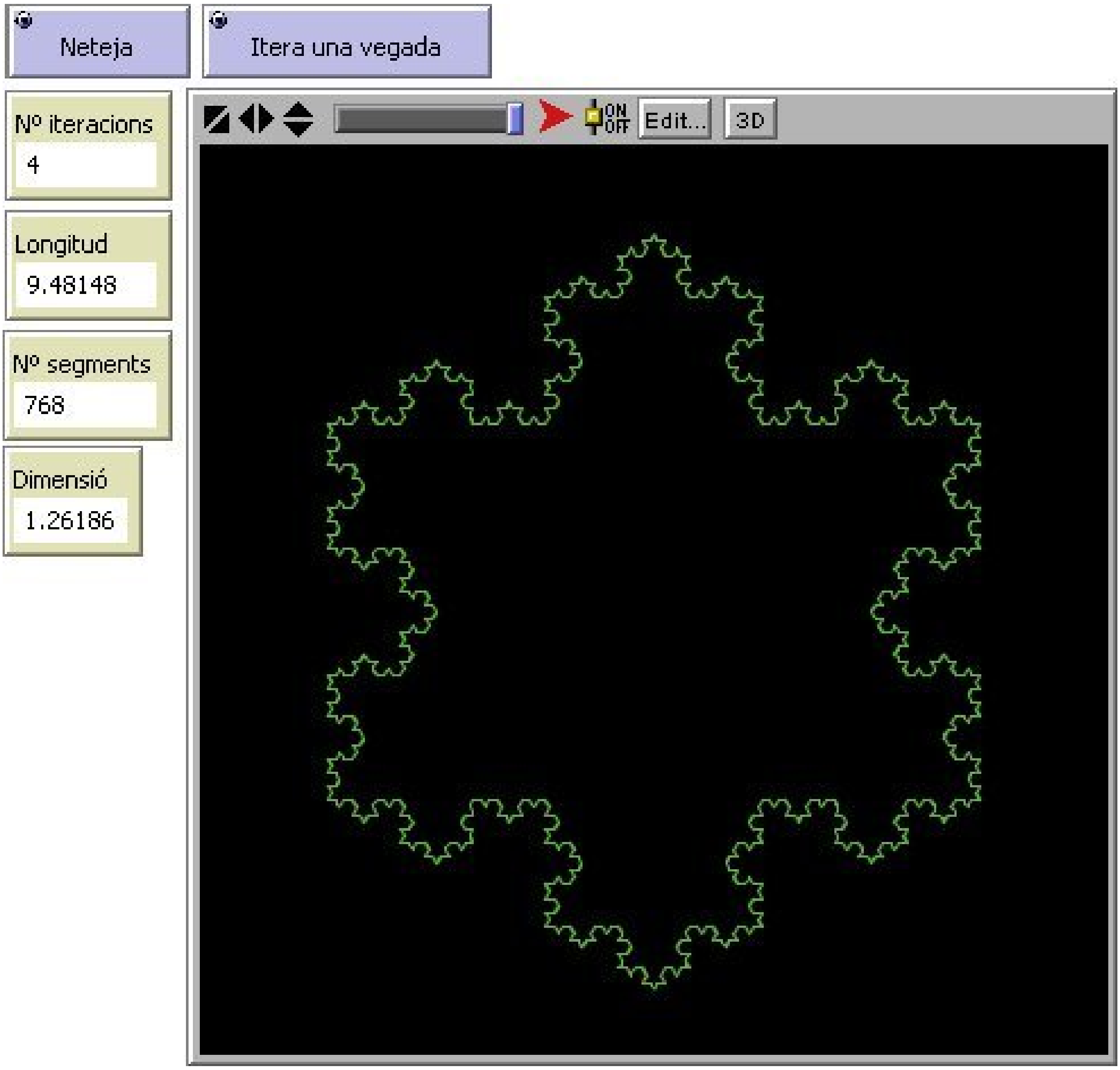
La segona iteració:



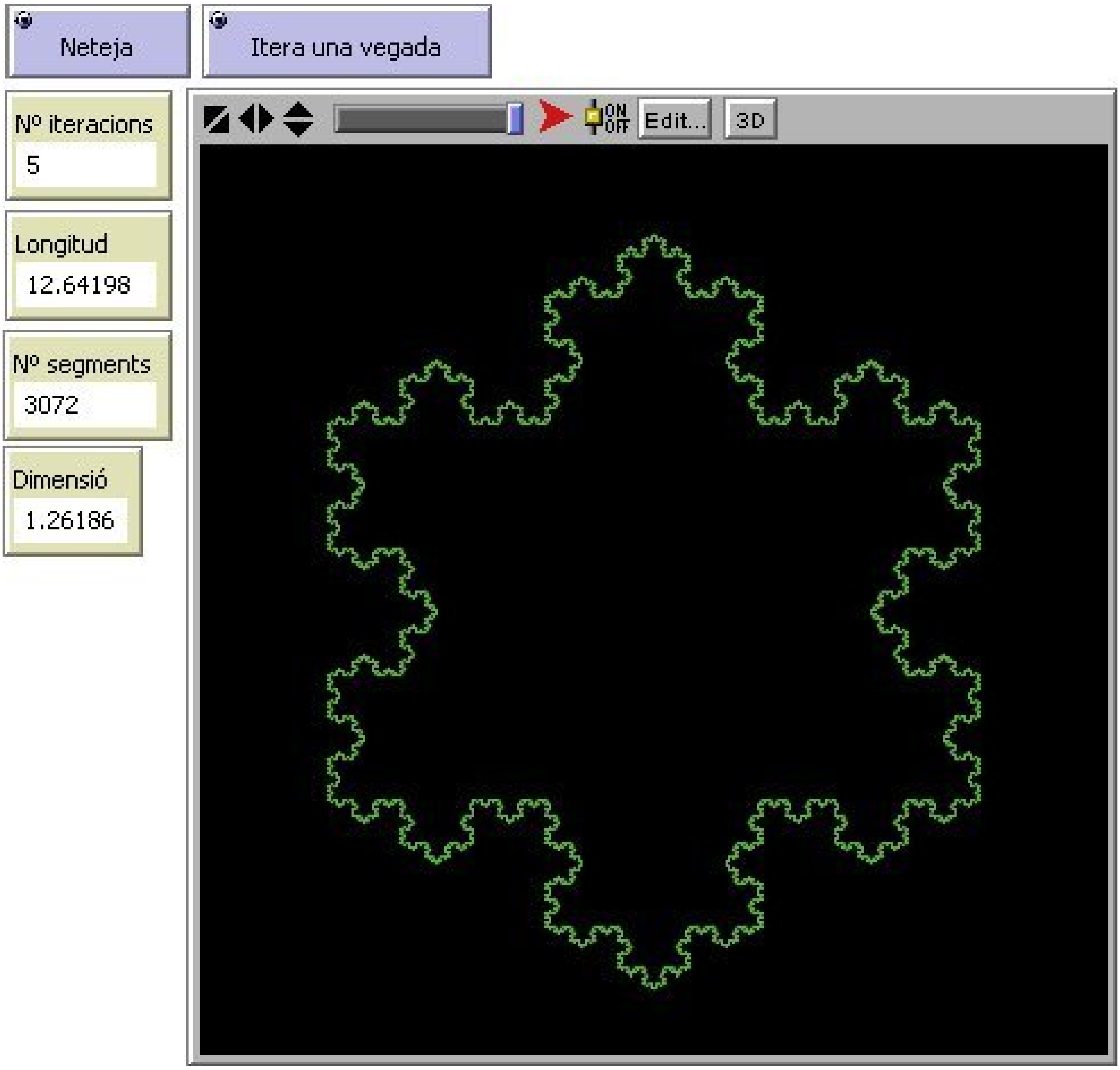
La tercera iteració:



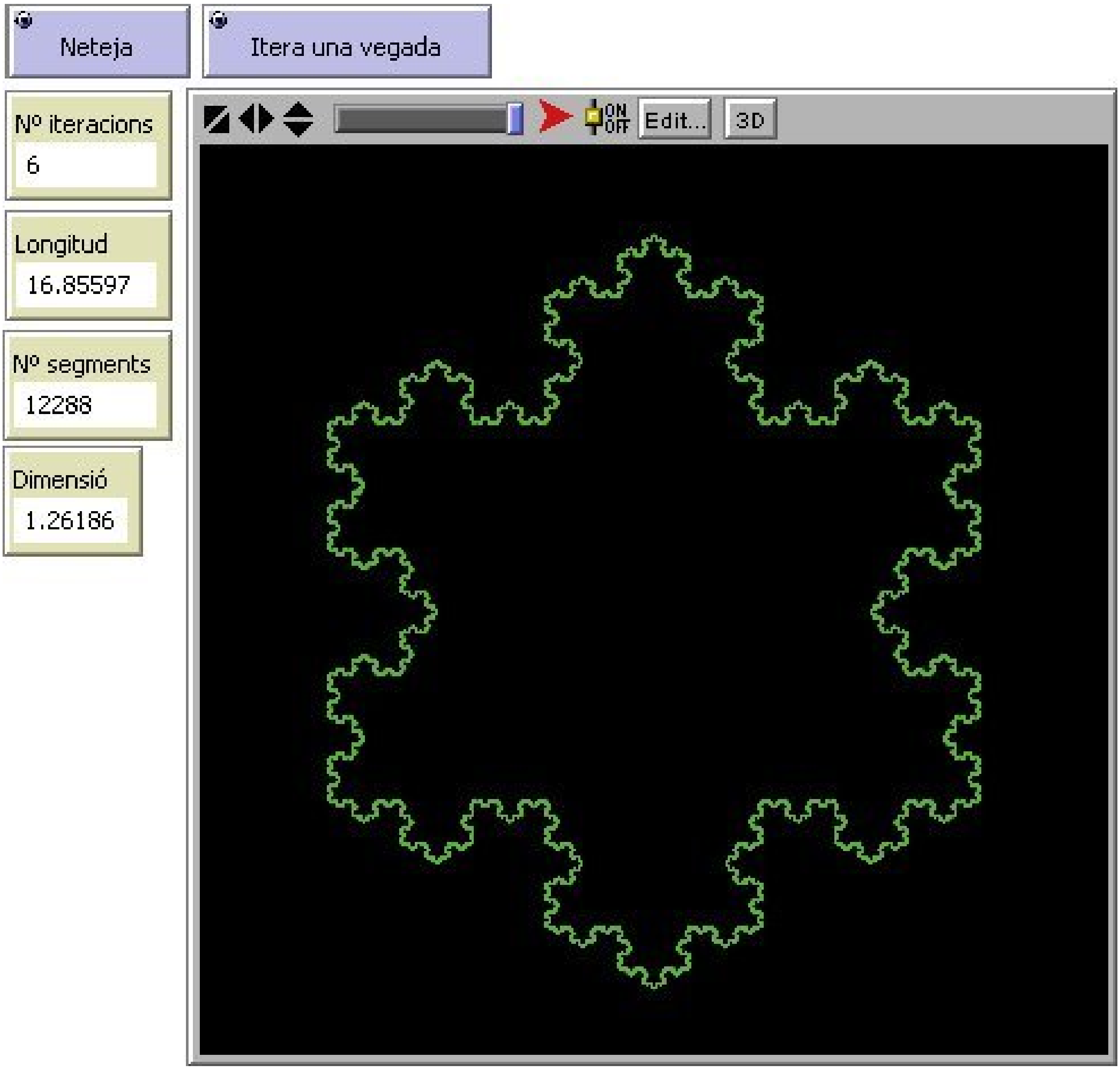
La quarta iteració:



La cinquena iteració:



La sisena iteració:



Els marcadors, aquesta vegada, s'assemblen força a les de la pràctica anterior ja que un deriva de l'altre.

El nombre d'iteracions també expressa el valor de la variable iteracions que augmenta un punt per a cada vegada que fem la instrucció iterar.

La longitud, en canvi, si que varia ja que tenim tres segments de una unitat de longitud cada un. Així que tenim com a longitud inicial 3 i cada vegada que iterem, augmentem 4/3 la longitud. Així que 3*4/3 = 4 i la fórmula canvia i és: $4^{iteracions}$

El nombre de segments continua corresponent al nombre de tortugues vigents en el moment el que passa és que aquest cop hi ha el triple de tortugues que en l'altre pràctica

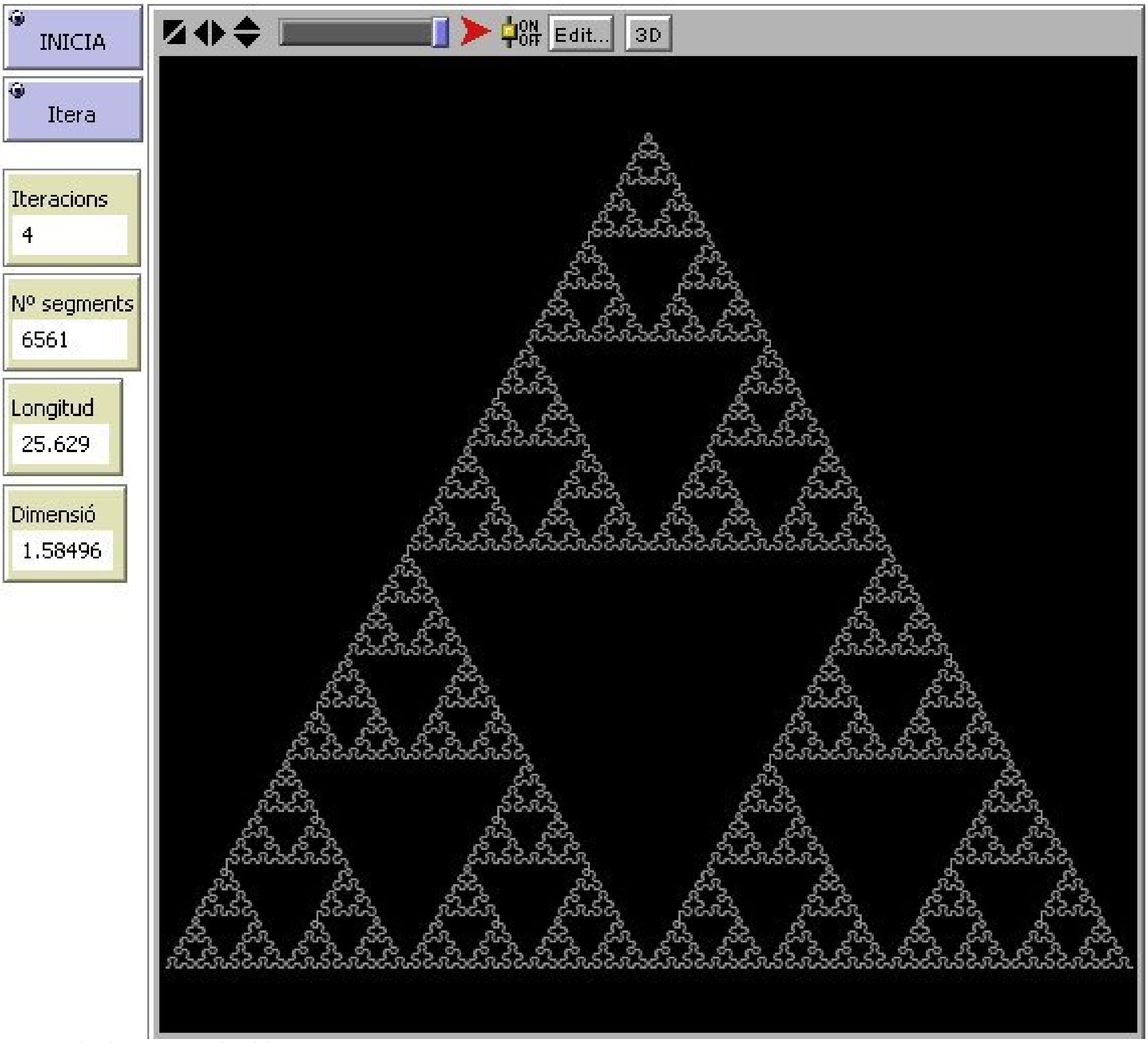
per a les mateixes iteracions.
Les instruccions que rep el programa per a crear el gràfic són adaptades, com hem dit de la pràctica anterior per tal de modificar el programa i obtenir el gràfic desitjat són:

```
globals [iteracio mida]

to neteja
        ca
        set iteracio 0
        create-turtles 1
        ask turtles
                [ setxy         min-pxcor + 25
                                min-pycor + 50
                set heading 30
                set color green
                pen-down
                set mida 130
                fd mida
                rt 120
                hatch 1 []
                fd mida
                rt 120
                hatch 1 []
                fd mida
                rt 120  ]
        ask turtles [hide-turtle]
end

to cami
        clear-drawing
        set mida (mida / 3)
        ask turtles
                [pen-down
                itera
                hide-turtle]
        set iteracio iteracio + 1
end

to itera
```

```
        hatch 1 [ ]
        fd mida
        lt 60
        hatch 1 [ ]
        fd mida
        rt 120
        hatch 1 [ ]
        fd mida
        lt 60
        fd mida
        lt 180
        fd mida
        lt 180
end
```

Com veiem, la principal diferencia és que a la instrucció neteja, fem fer a la tortuga un recorregut inicial per tal de dibuixar el primer triangle i de deixar preparades dos tortugues que després faran les corbes corresponents i, tornar al punt inicial, col·locar-se en posició per tal de que al iterar cada tortuga dibuixi cada una d'elles una corba diferent.

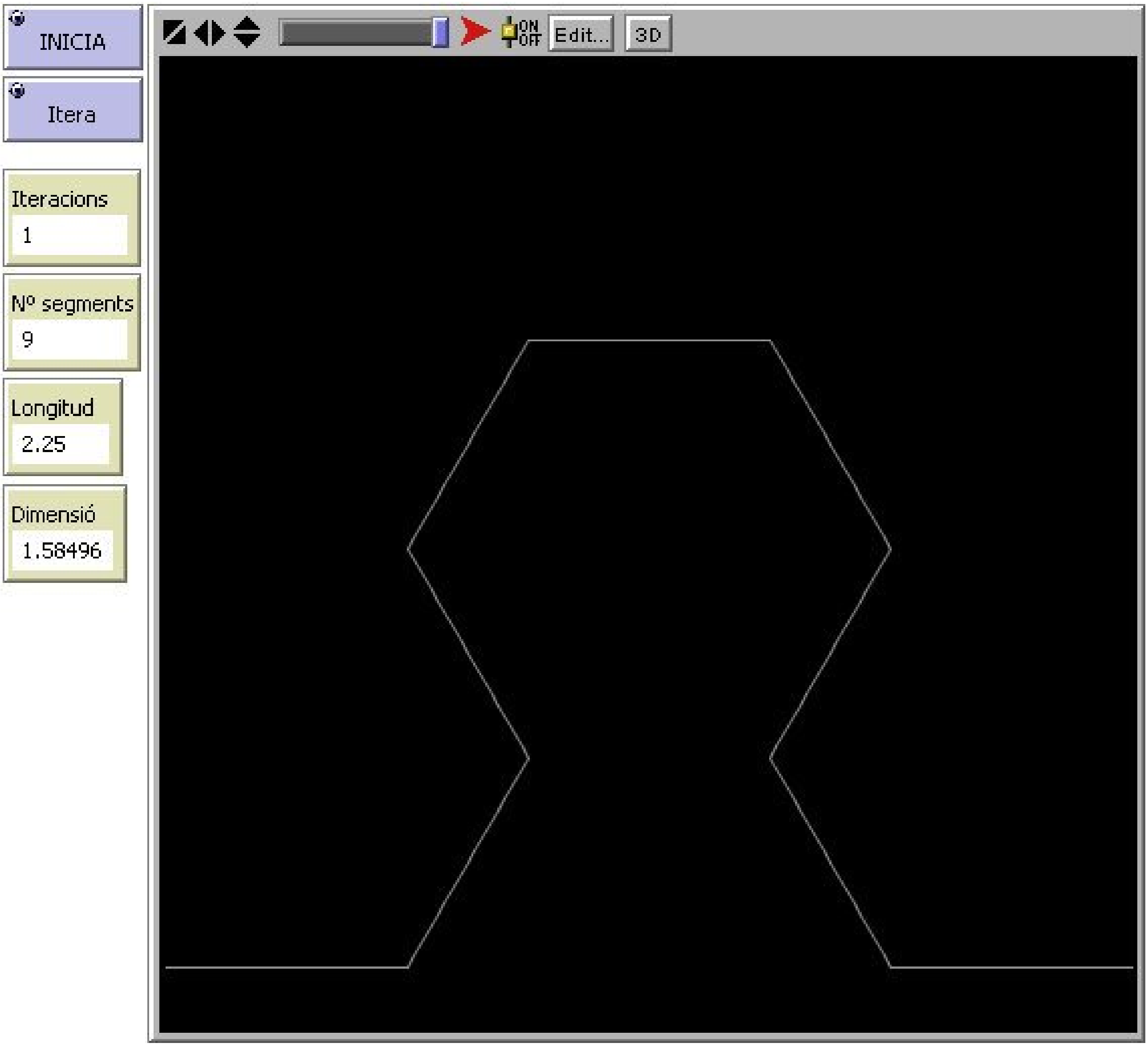
El Triangle de Sierpiński



Triangle de Sierpiński dibuixat per NetLogo

Aquest triangle està construït pel programa creat, que fa servir un sistema-L per a crear el triangle de Sierpiński i per tant, és diferent al que hem explicat ja que està format per segments en lloc de triangles. Tot i aquesta diferencia evident, com podem veure s'arriba al mateix resultat.

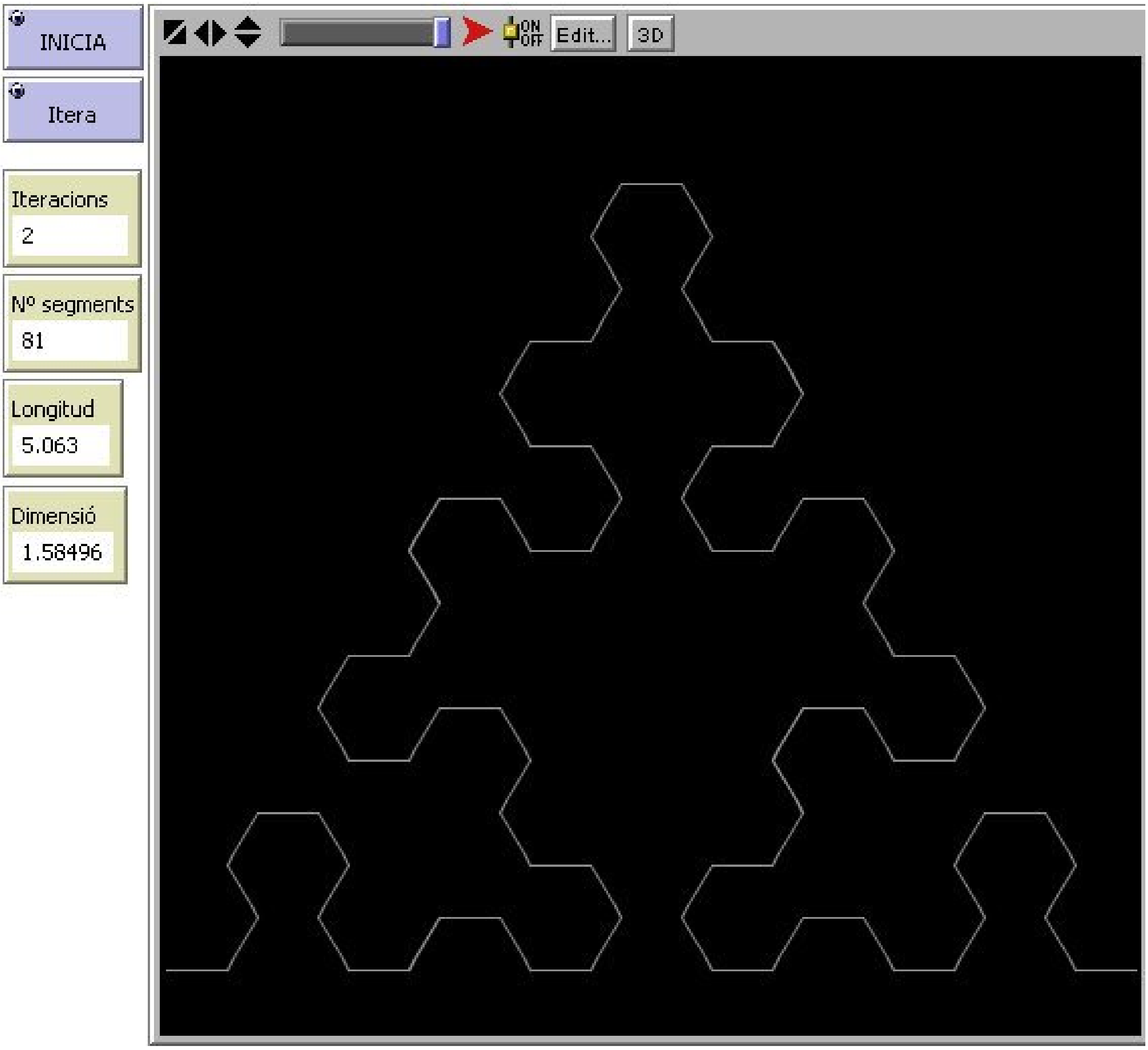
La funció que iterem, s'aplica sobre un segment de longitud 1 i, seguint el mateix mètode que en el que hem vist a la pràctica de la Corba de Koch, crea una corba com la que veiem a la imatge inferior(iteració 1). Sobre cada segment tornarem a iterar la mateixa funció i obtindrem així, el Triangle de Sierpiński.



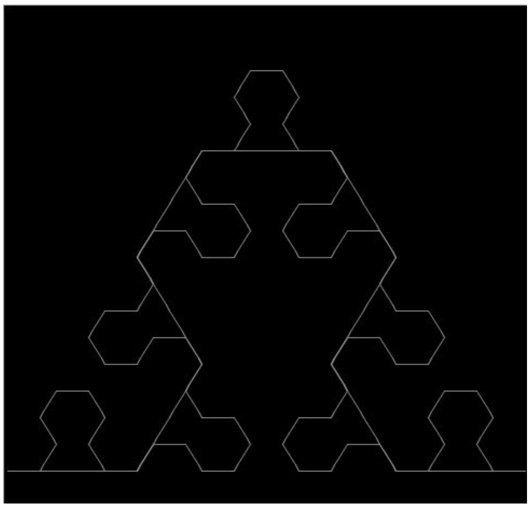
Aquesta funció transforma un segment en una estructura de nou segments de longitud 1/4 de la longitud del segment inicial.
Així que la longitud de la corba és 9/4 de la longitud del segment o de la corba anterior.
Igual que en les pràctiques anteriors, en el extrem esquerre de cada segment hi ha col·locada una tortuga preparada per a fer la instrucció que provoca una corba com la de la imatge superior però en una escala més petita i sembrar, preparar les tortugues que pròximament, en la següent iteració faran el seu tros de corba en una escala cada vegada

més petita.

La segona iteració ens ensenyarà com es torna a aplicar la funció en cada segment (com cada tortuga preparada fa el seu camí corresponent:

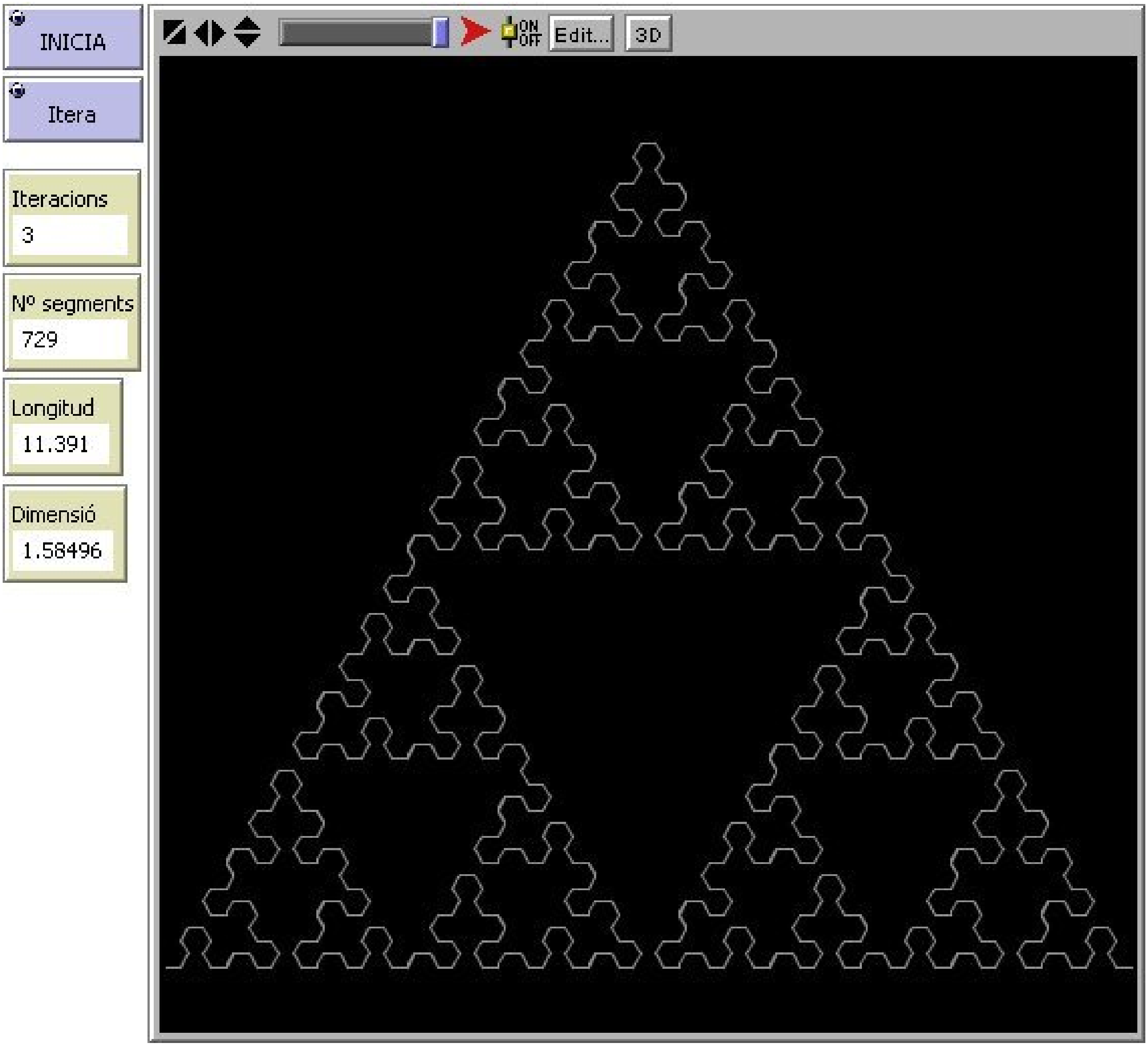


Aquí, trobem una dificultat afegida en el dibuix d'aquest gràfic. Que podrem apreciar millor si sobreposem les dos imatges de la primera i la segona iteració:



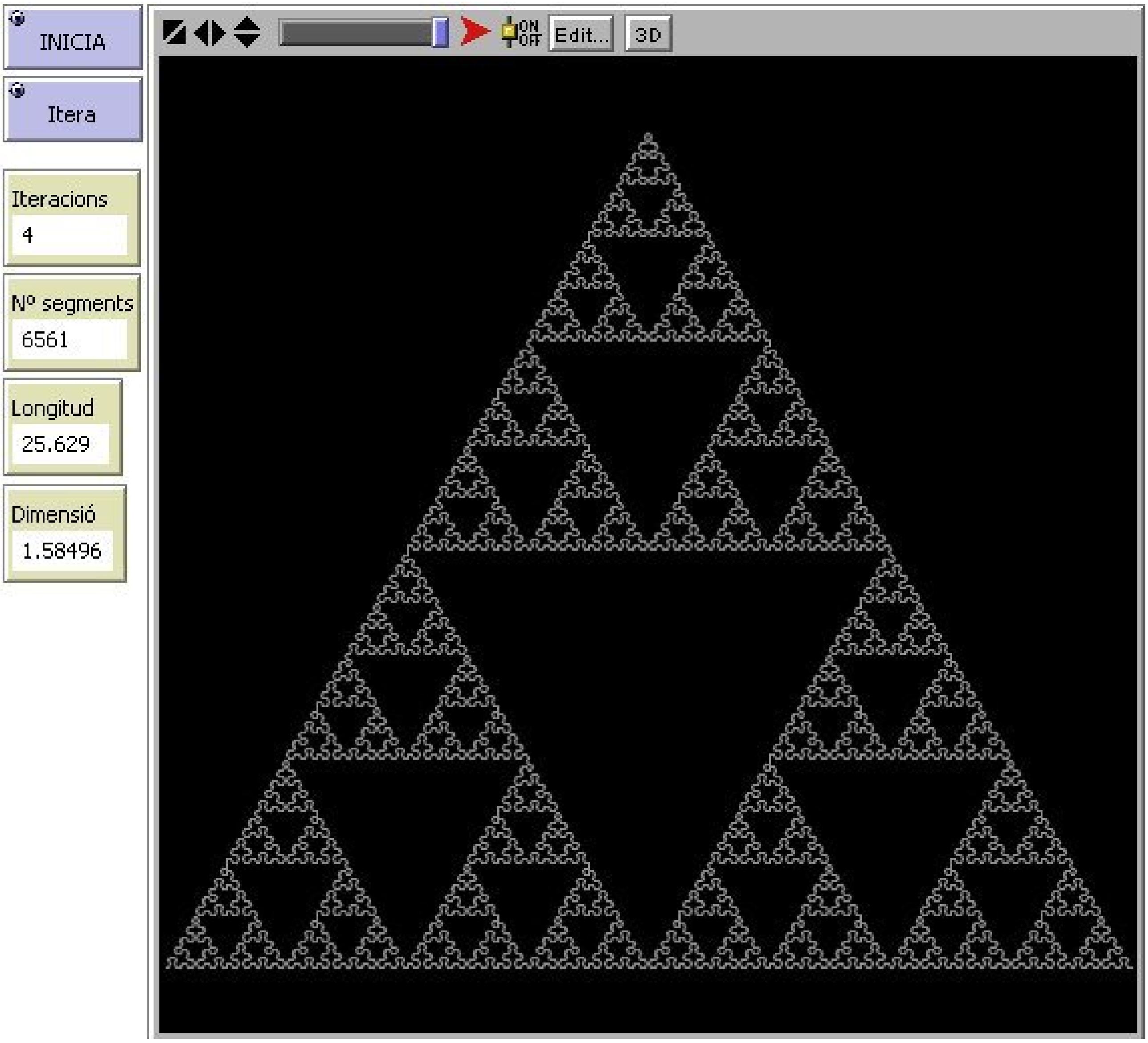
Hem iterat la funció a cada segment de la corba de la primera iteració. Però no totes les iteracions són fetes pel mateix cantó de la corba. Així que hem hagut d'iterar arbitràriament a una banda i a l'altre de la corba per tal de aconseguir la figura desitjada. Això, s'ha aconseguit orientant de manera diferent les tortugues, que si havien de fer la iteració per sota, es col·locaven a l'extrem dret i no esquerre del segment.

La tercera iteració:

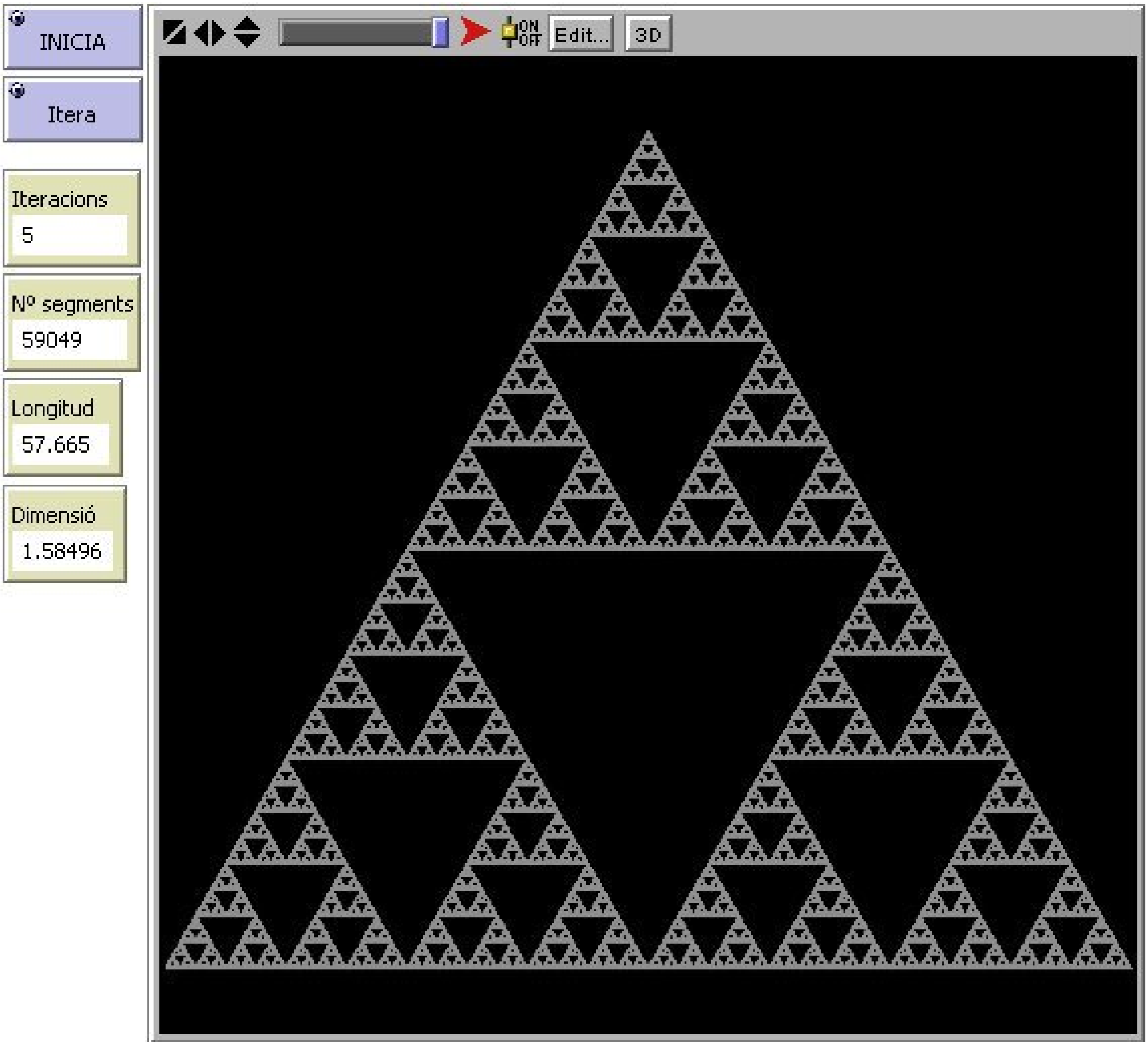


Ja podem veure la estructura del Triangle de Sierpiński formada per aquest corba.

La quarta iteració:



La cinquena iteració:



La imatge ja no dóna per a més així que ens aturarem de mostrar les iteracions però sabem que el Triangle de Sierpiński continua iterant-se indefinidament i que té àrea nul·la. I en el cas de la nostra figura formada per una corba degut al sistema-L utilitzat, la corba té longitud infinita.

En aquesta nova pràctica, hi han els mateixos marcadors que en les dos anteriors però, adaptats a la nova figura.

Aquest cop, la longitud de la corba és 9/4 que l'anterior i el segment inicial val 1 de longitud, així que la fórmula de la longitud és $(9/4)^{iteracions}$.

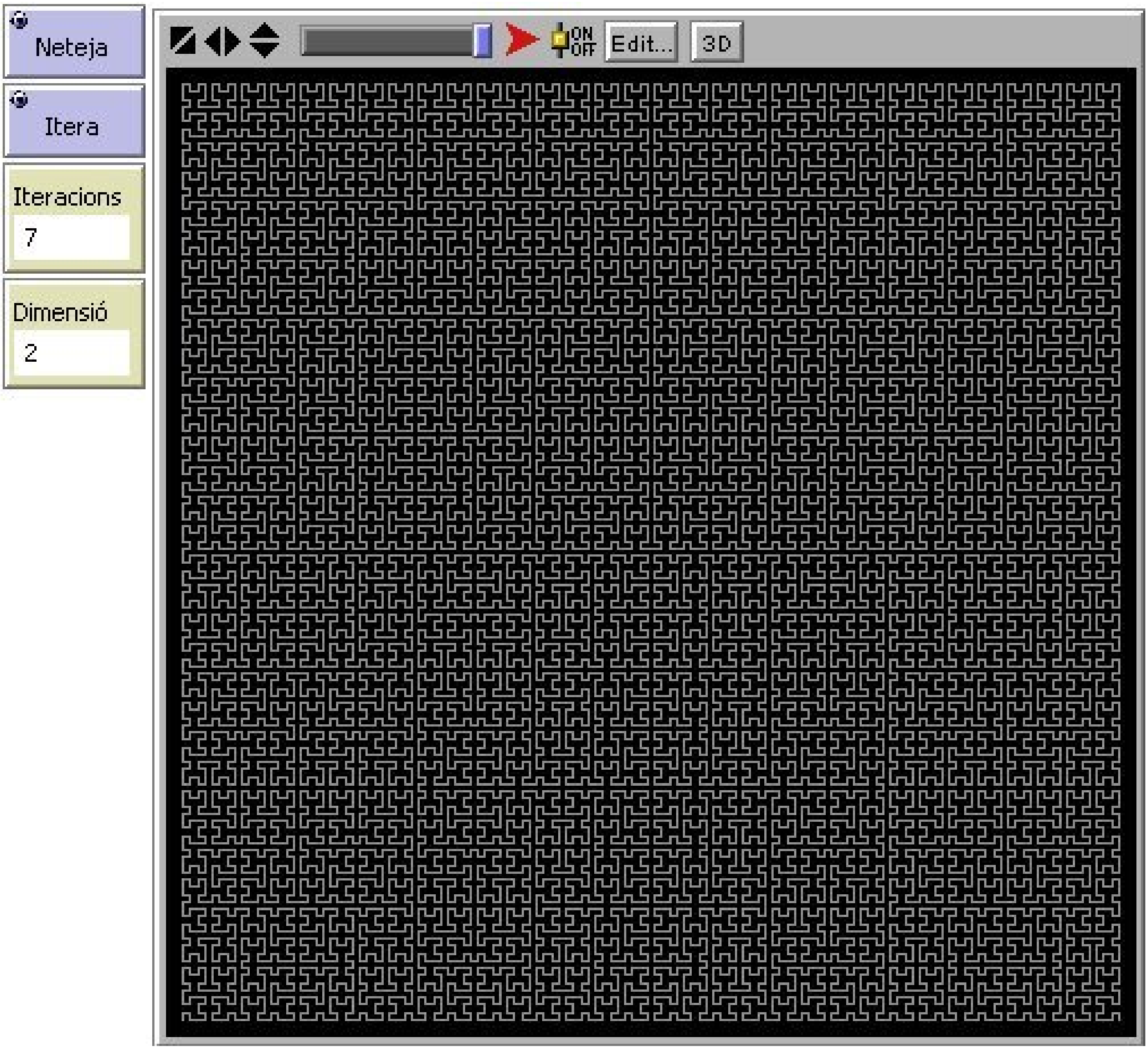
Iteracions	Nº segments	Longitud	Dimensió
0	1	1.0	1.58496
1	9	2.25	1.58496
2	81	5.063	1.58496
3	729	11.391	1.58496
4	6561	25.629	1.58496
5	59049	57.665	1.58496

Les instruccions donades al programa per a que faci aquest gràfic són:

```
globals [ mida iteracions ]

to neteja
        clear-all
        create-turtles 1
        ask turtles [
                setxy          min-pxcor
                               min-pycor + 5
                set heading 90   ]
        set mida max-pxcor * 2
        set iteracions 0
end

to cami
        if iteracions = 6 [stop]
        clear-drawing
        ask turtles [pen-down]
        itera
        ask turtles [hide-turtle]
        set mida mida / 4
        set iteracions iteracions + 1
end
to itera
```

```
    ask turtles [
        hatch 1 []
        fd mida / 4
        lt 60
        fd mida / 4
        lt 180
        hatch 1 []
        lt 180
        lt 60
        hatch 1 []
        fd mida / 4
        rt 60
        fd mida / 4
        lt 180
        hatch 1 []
        lt 180
        rt 60
        hatch 1 []
        fd mida / 4
        rt 60
        fd mida / 4
        lt 180
        hatch 1 []
        lt 180
        rt 60
        hatch 1 []
        fd mida / 4
        lt 60
        fd mida / 4
        lt 180
        hatch 1 []
        lt 180
        lt 60
        fd mida / 4
        lt 180
        fd mida / 4
        lt 180 ]
end
```

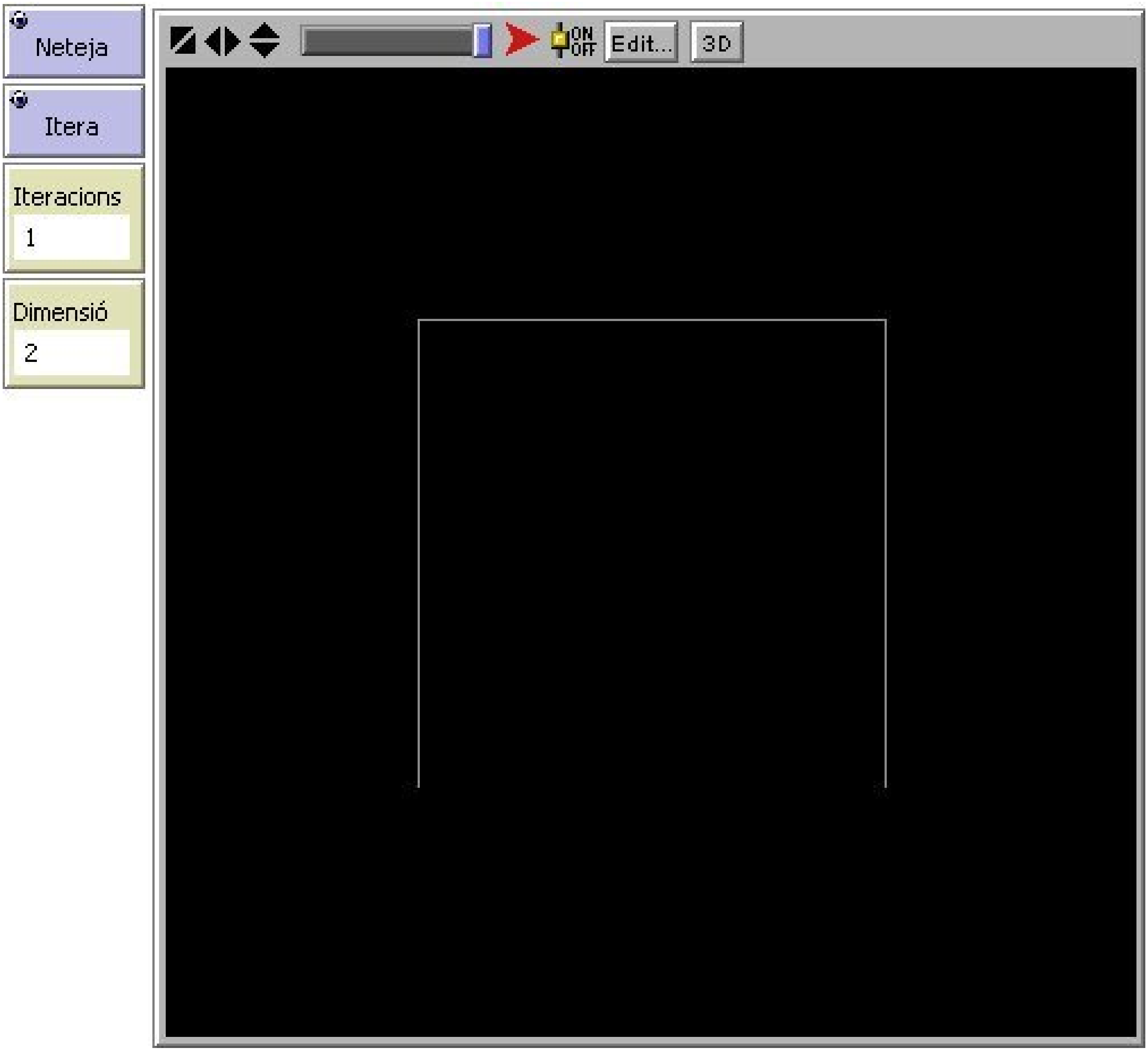
La corba de Hilbert



Aquesta és una imatge de una de les iteracions més avançades feta pel programa creat per a fer la corba de Hilbert. Recordem que aquest corba té dimensió 2, s'equival a un pla com veurem.

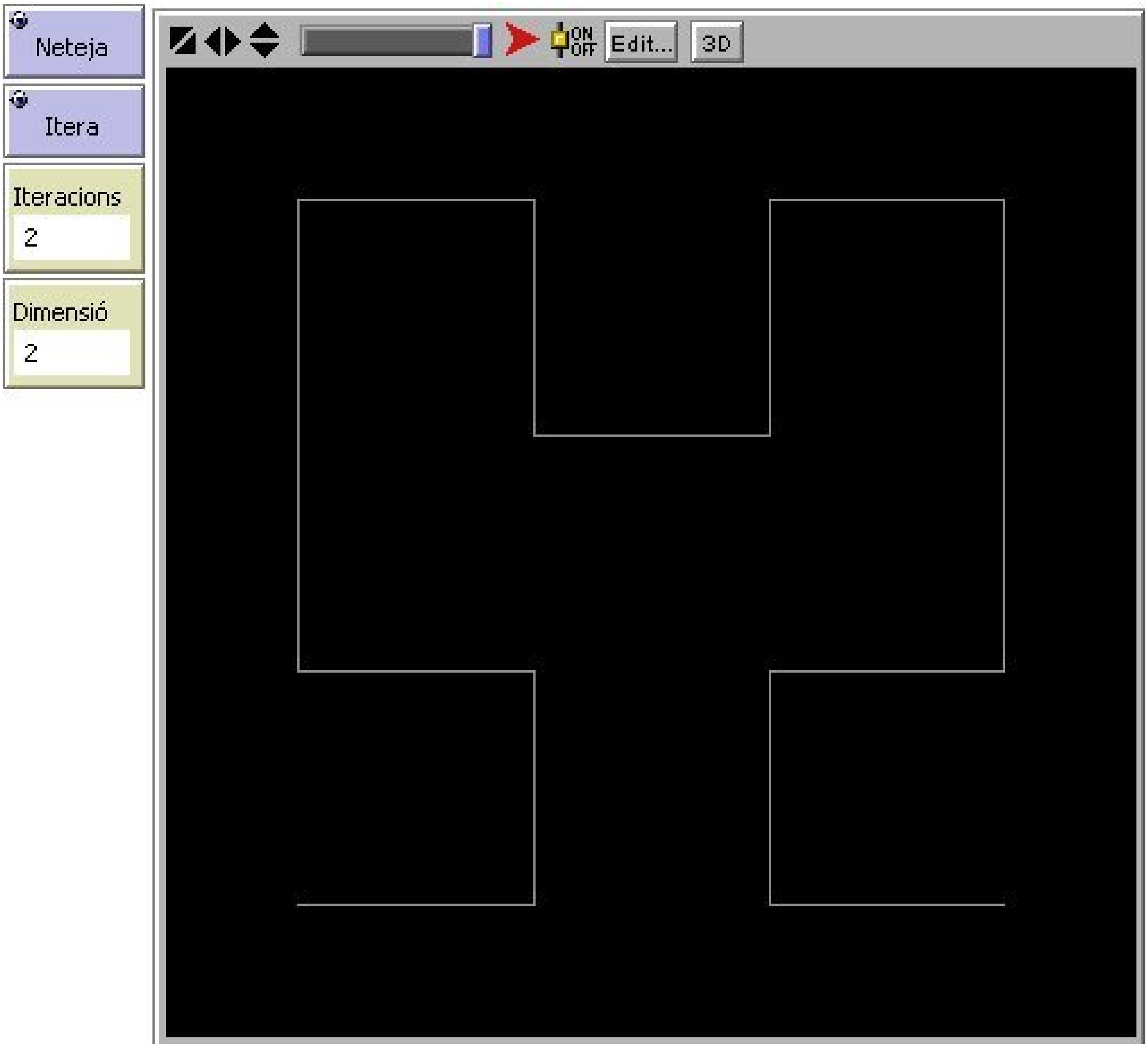
Per a aquest procés, no es comença a iterar des d'un segment, sinó des de tres segments però una sola tortuga que fa tot el camí:

(iteració 1)



Sobre els quals aplicarem la funció:

(iteració 2)



Així, obtenim quatre conjunts com els de la imatge anterior (iteració 1) sobre les quals podrem tornar a aplicar la funció. Però si us hi fixeu, hi ha uns segments suplementaris que fan de connectors entre conjunts iterables.
Aquests conjunts connectors són el que complica aquesta pràctica.

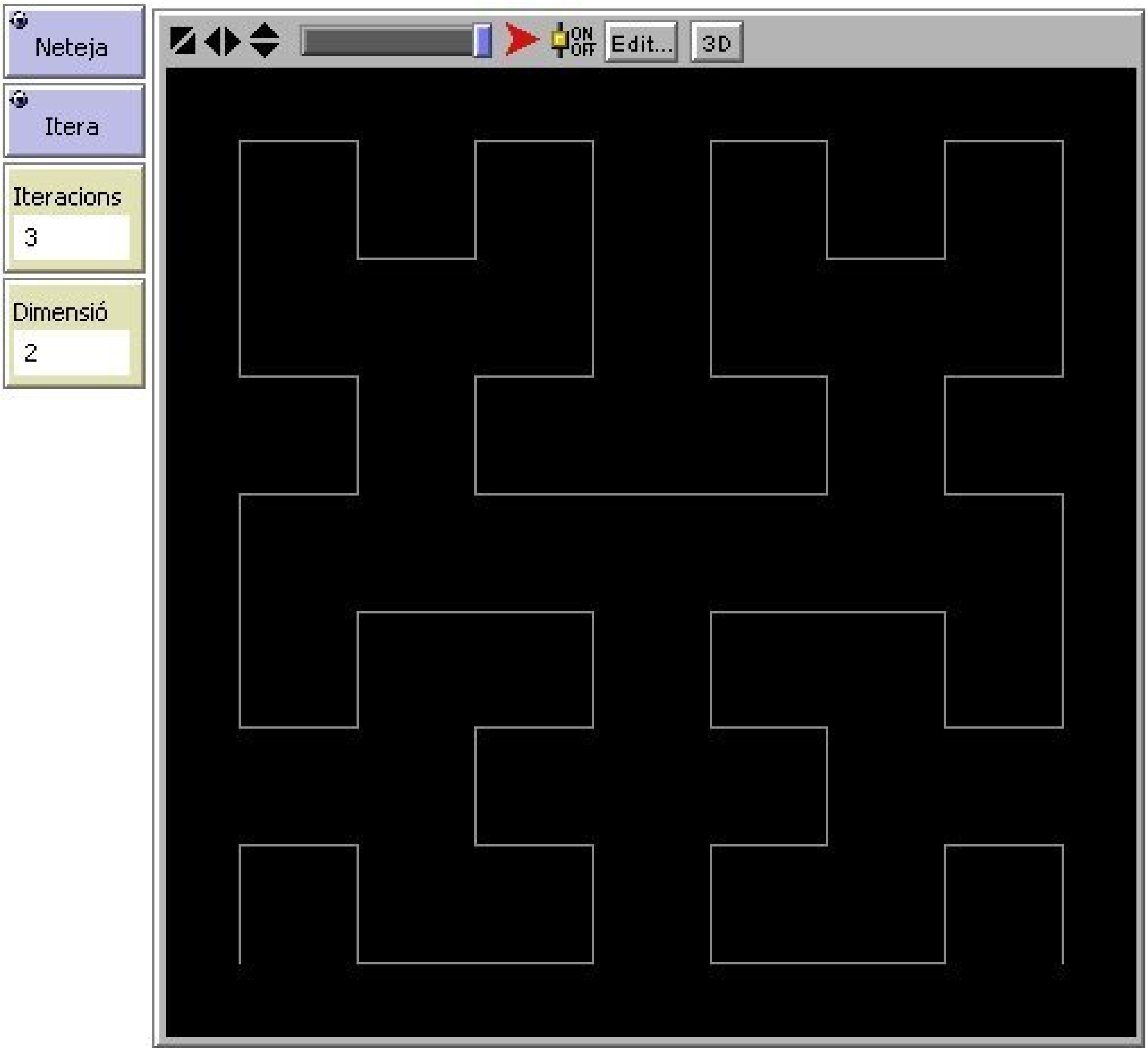
Aquesta pràctica és diferent a les altres tres explicades. Aquesta vegada, no totes les tortugues fan el mateix camí. Sinó que hi ha dos categories de tortugues. Unes fan una camí com el que es veu representat a la imatge de la iteració 1 però les altres, fan de connectors, simplement fan un segment de mida igual a un dels costats del camí de les primeres.

Per a dibuixar la corba de Hilbert, quan una tortuga fa el seu camí, alhora, ha d'anar no només creant tortugues per allà on passa i on toca, sinó que les ha de classificar (indicar quina classe de tortuga és per tal de fer, després, el camí correcte) i les ha de col·locar en la posició correcte per a que al iterar la funció facin el camí que els hi pertoca en l'orientació, posició que els hi pertoca.

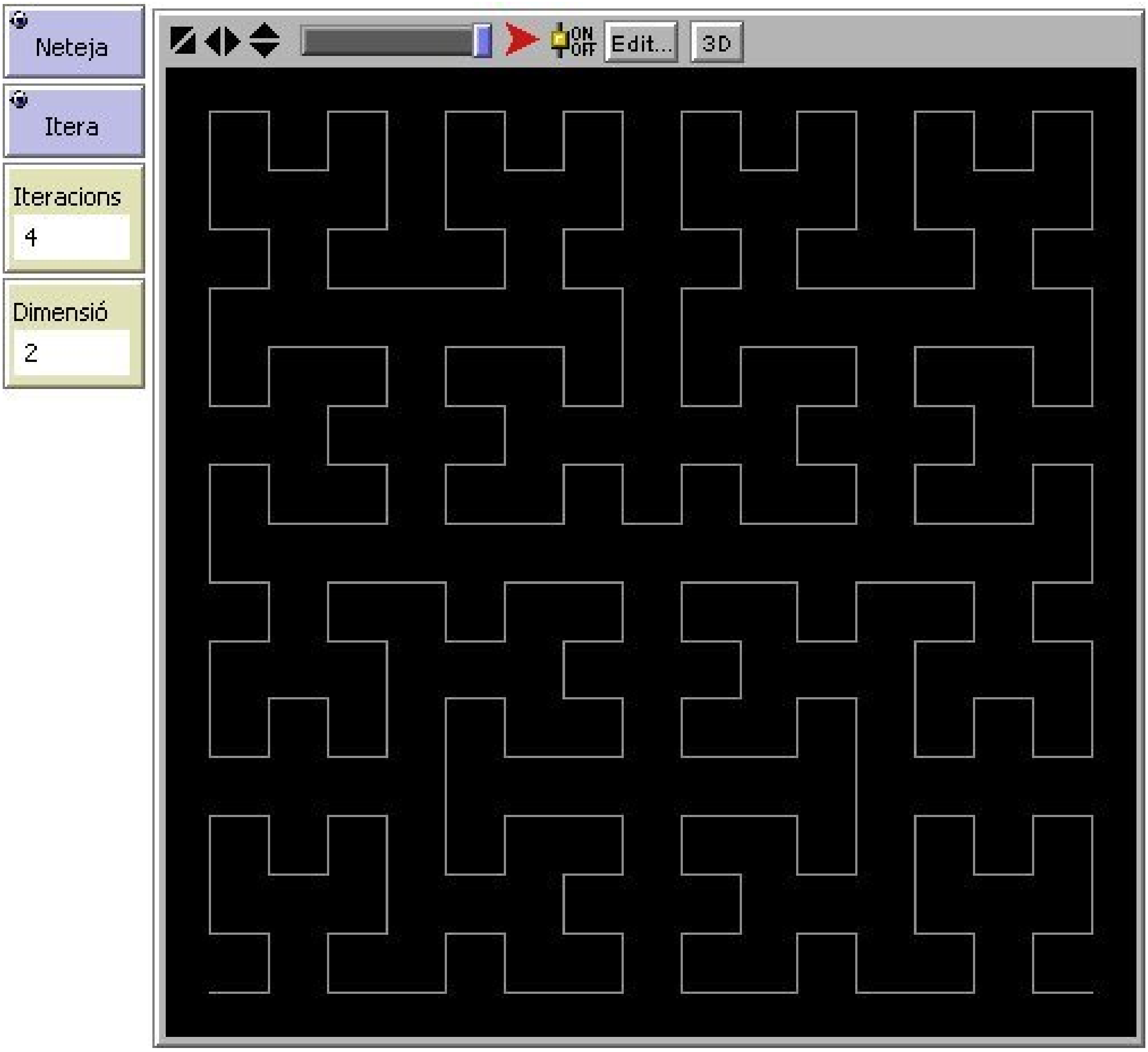
Si us hi fixeu, notareu que la corba no comença en el mateix punt de la pantalla en les dos imatges que hem mostrat. En la imatge de la iteració 2, la corba comença més aprop de la cantonada inferior esquerre que no pas la corba de la imatge de la iteració 1. El mateix passa amb el final de la corba. A això ens referim quan volem dir que s'han de col·locar les tortugues en la posició adequada. Quan una tortuga fa el camí que li pertoca ha d'anar repartint per tota la imatge tortugues d'un tipus o d'un altre segons correspongui per tal de deixar preparades totes les tortugues que faran la seva funció en la següent iteració.

Podem comprovar el resultat e les iteracions que segueixen:

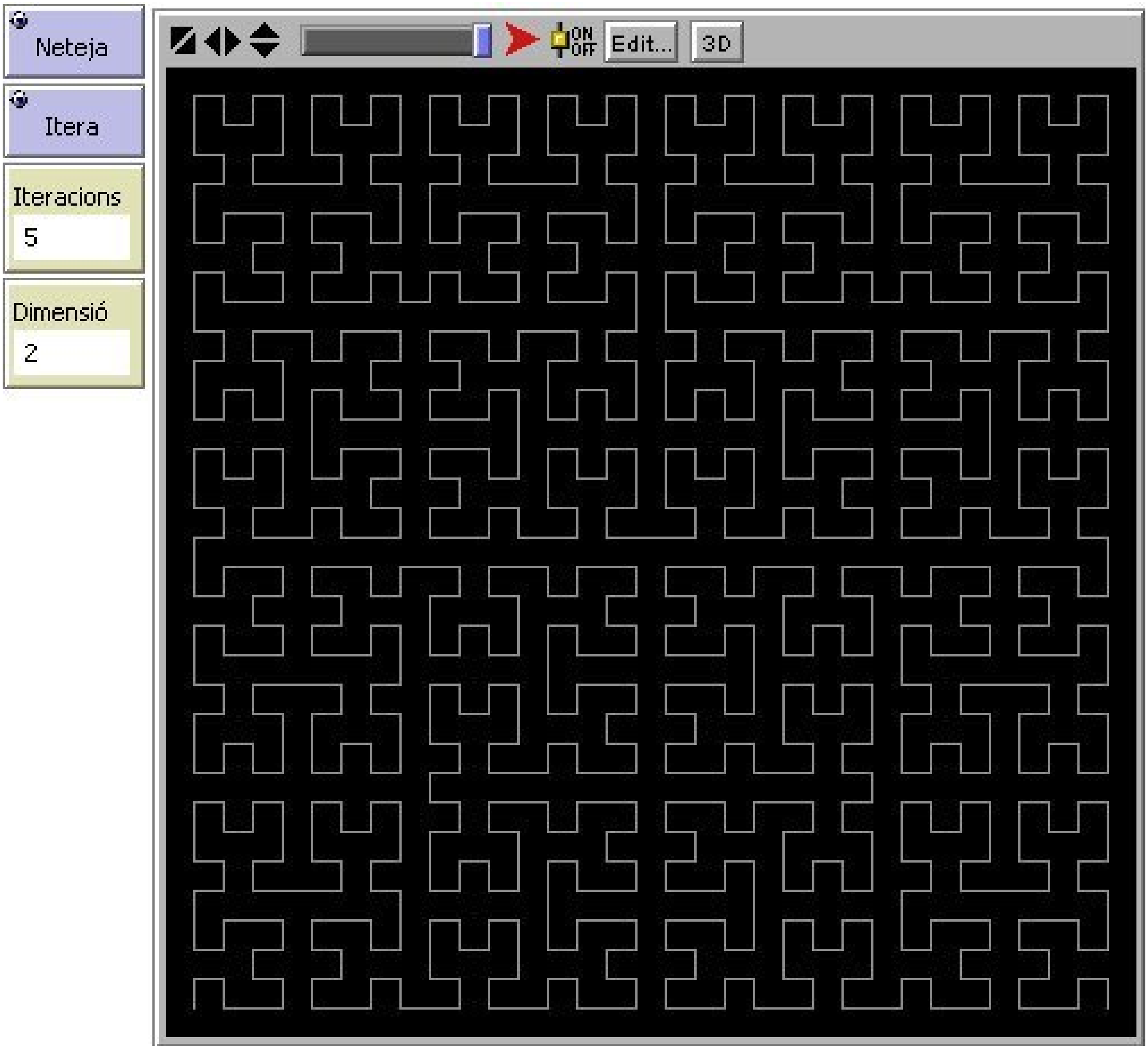
Tercera iteració:



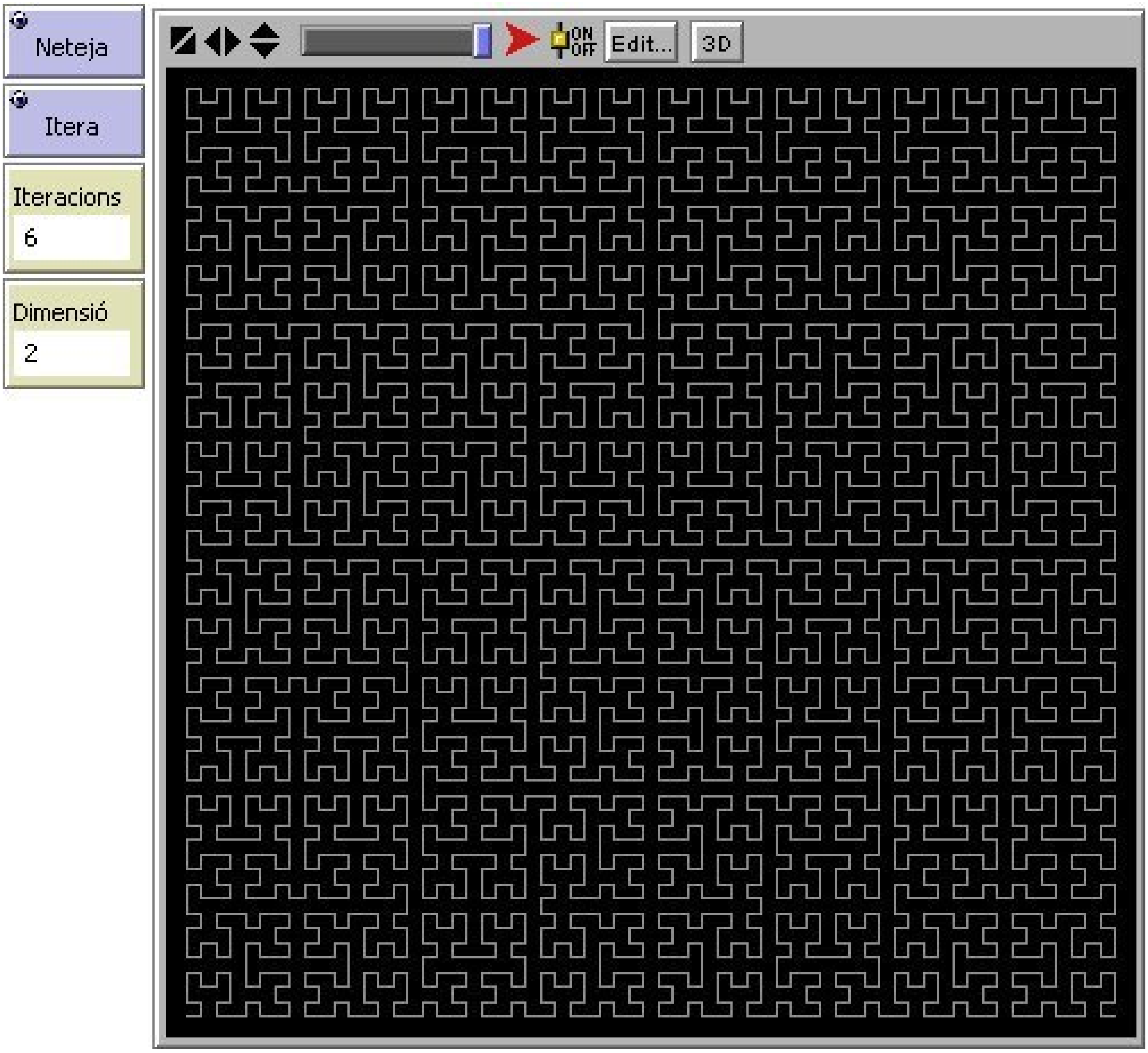
Quarta iteració:



Cinquena iteració:

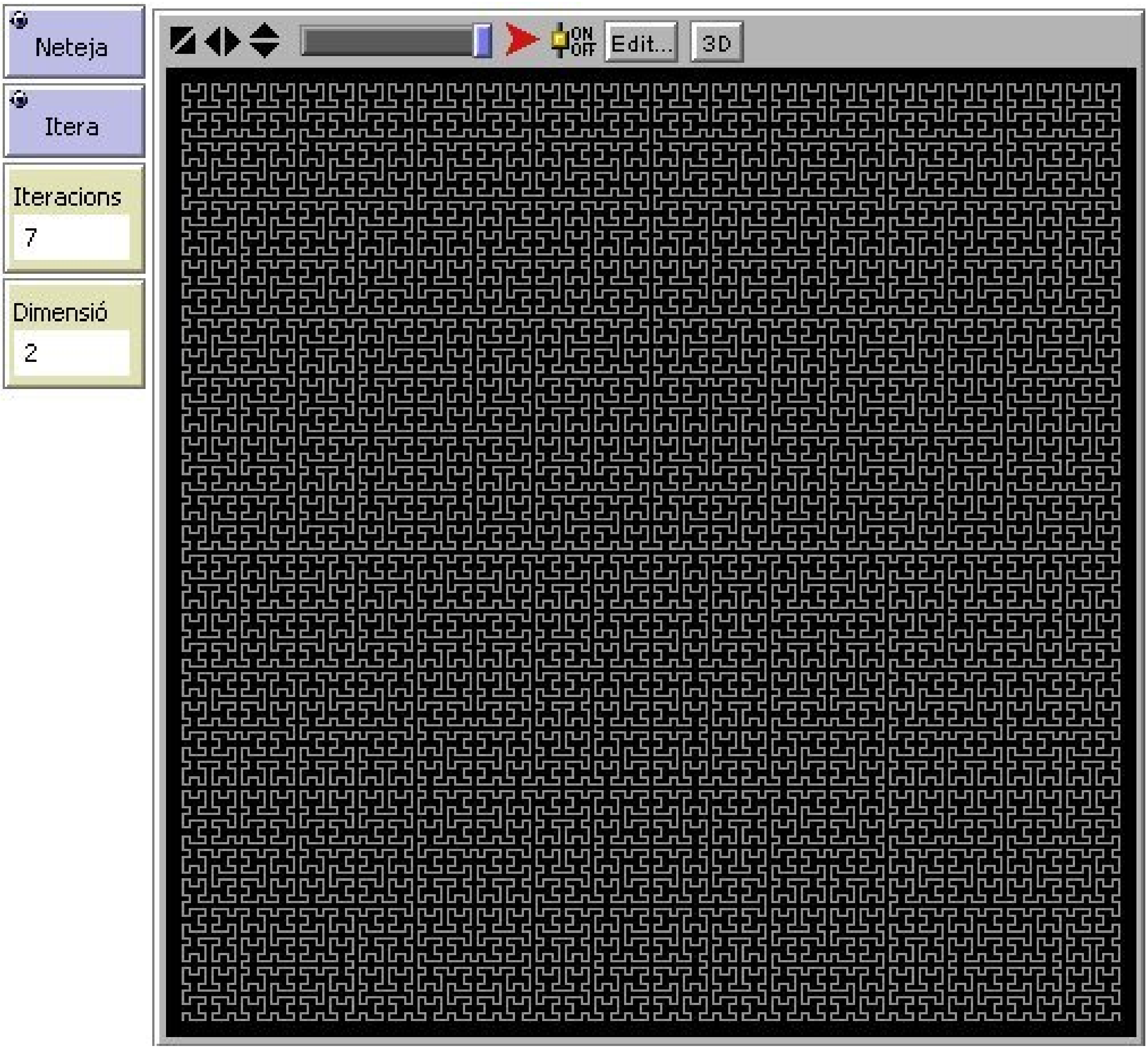


Sisena iteració:

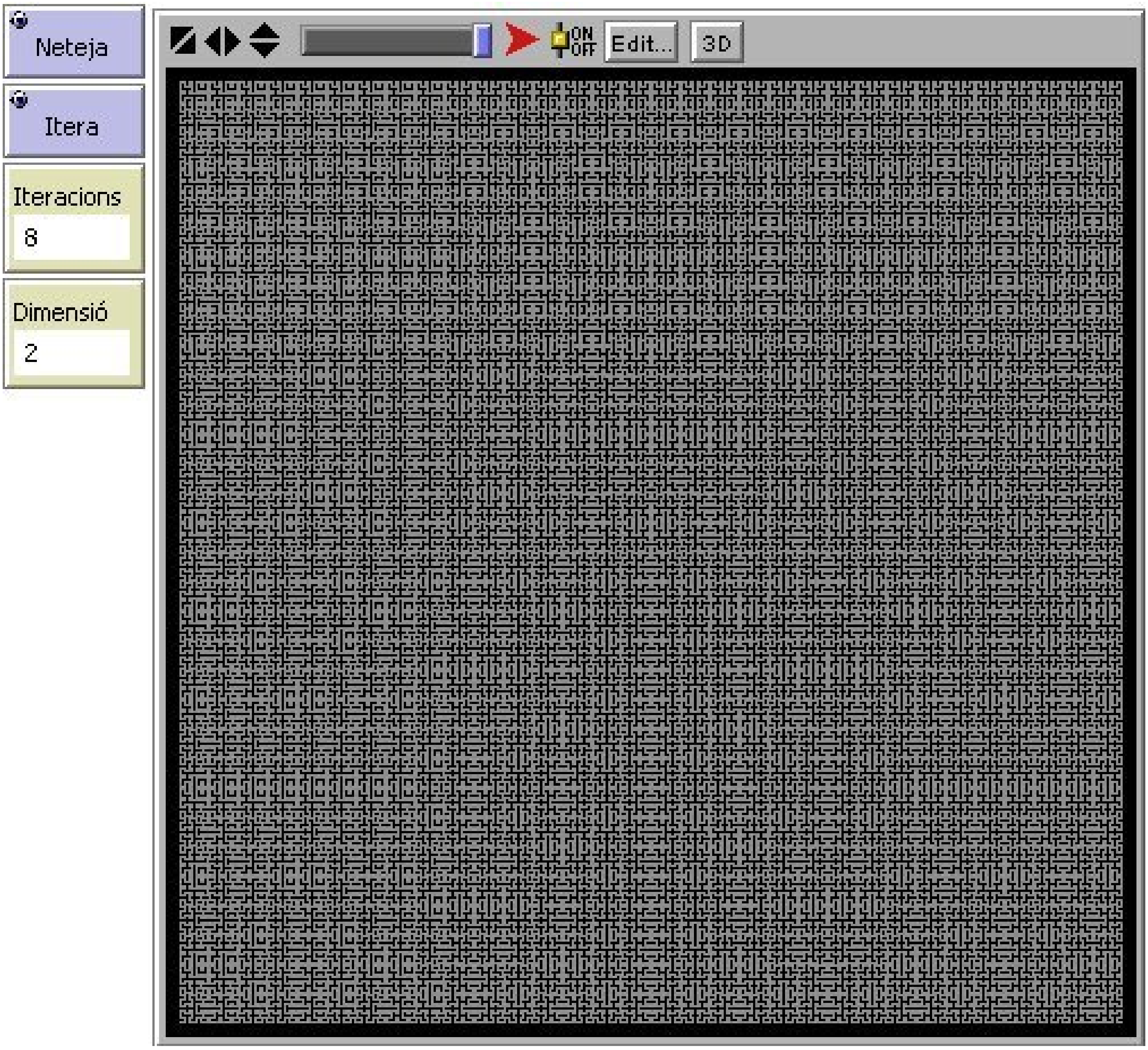


Poc a poc, la corba va omplint tot l'espai del pla.

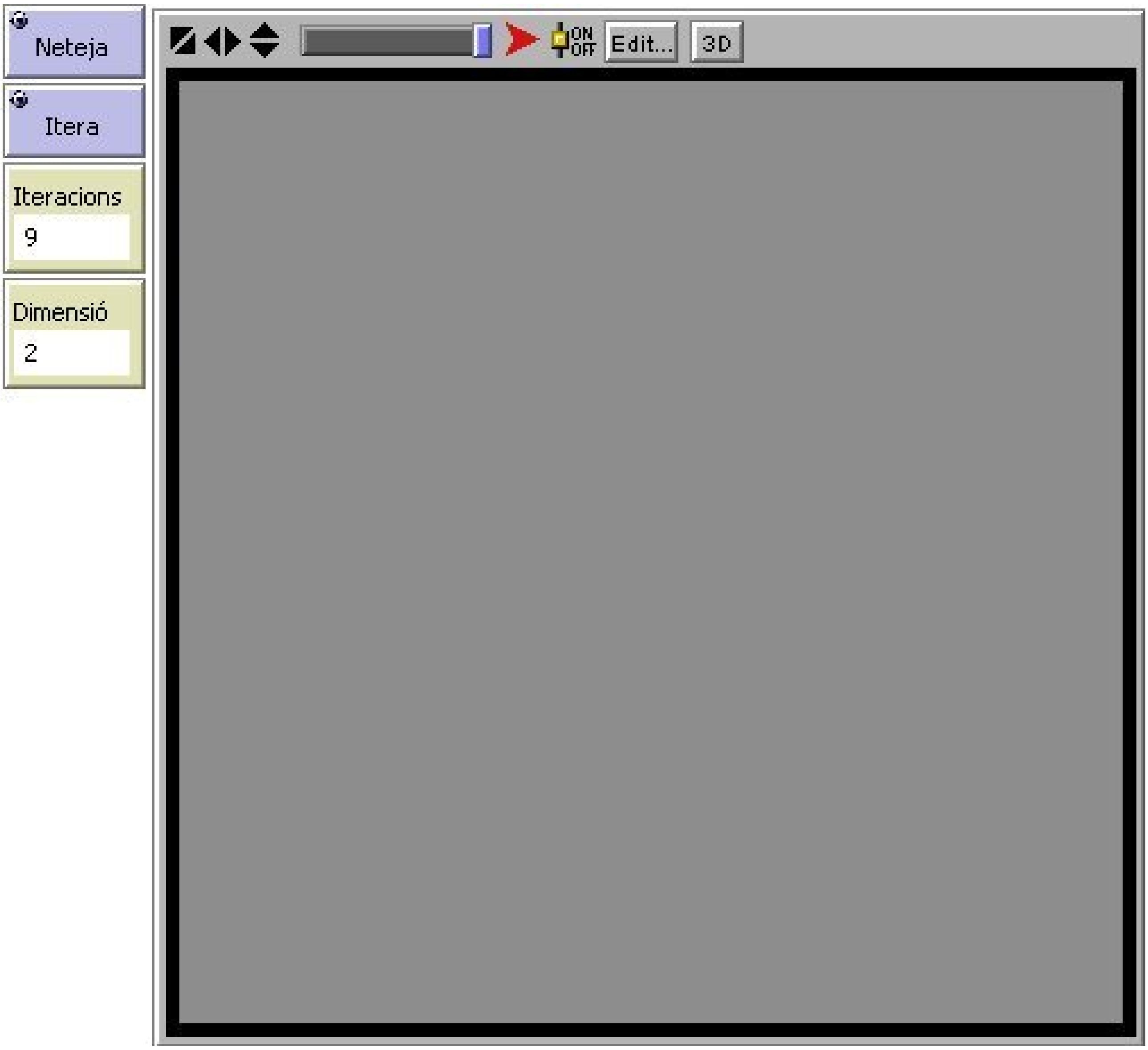
Setena iteració:



Vuitena iteració:



Novena iteració:



Aquí, la imatge ja no dóna per a més i nosaltres veiem un pla perfecte. Tot i així, s'hauria de iterar infinitament per a obtenir realment un pla iterant aquesta funció. Posem aquesta imatge per a que es pugui apreciar com la corba acaba omplint el pla tot i que no succeeix en la iteració 9.

Les instruccions que rep el programa per a crear aquestes imatges són:

```
globals [mida iteracions]

breed [ms m]
breed [primeres primera]
breed [ls l]

to neteja
        clear-all
        create-primeres 1
        ask primeres [
                 setxy
                         min-pxcor + max-pxcor / 2
                         min-pycor + max-pycor / 2
                 pen-down
                hide-turtle]
        set mida max-pxcor
        set iteracions 0
end

to itera
        clear-drawing
        ask turtles [   if breed = primeres [camim]
                        if breed = ms [camim]
                        if breed = ls [camil] ]
        ask turtles [
                        hide-turtle]
        set mida mida / 2
        set iteracions iteracions + 1
end

to camim
        hatch-ms 1 [
                pen-up
                fd mida / 4
                lt 90
                fd mida / 4
                lt 180
                pen-down]
        hatch-ls 1 [
```

```
        pen-up
        fd mida / 4
        lt 90
        fd mida / 4
        rt 90
        pen-down]
fd mida
hatch-ms 1 [
        pen-up
        lt 90
        fd mida / 4
        lt 90
        fd mida / 4
        lt 180
        pen-down]
rt 90
fd mida
hatch-ls 1 [
        pen-up
        lt 180
        fd mida / 4
        lt 90
        fd mida / 4
        rt 90
        pen-down]
hatch-ms 1 [
        pen-up
        lt 180
        fd mida / 4
        lt 90
        fd mida / 4
        lt 180
        pen-down]
rt 90
hatch-ls 1 [
        pen-up
        fd mida / 4
        lt 90
        fd mida / 4
        rt 90
```

```
        pen-down]
    fd mida
    hatch-ms 1 [
        pen-up
        fd mida / 4
        lt 90
        fd mida / 4
        lt 180
        pen-down]
    die
end


to camil
    hatch-ls 1 [
        pen-up
        fd mida / 4
        lt 90
        fd mida / 4
        rt 90
        pen-down]
    fd mida
    die
end
```

4.4 Simulació de l’estructura d’ un arbre.

Aquesta pràctica consisteix en crear una programa propi, amb l'interprete del llenguatge Logo, NetLogo, preparat per a representar, dins d'unes limitacions, una figura el més propera possible a l'estructura d'un arbre.
Per a fer aquesta pràctica, varem començar amb unes instruccions senzilles que produïen un senzill arbre de nombre de branques fixe i mides invariables. A poc a poc, vam anar-li afegint totes les millores o requisits que se'ns presentaven possibles.

El resultat final, és un programa que construeix un arbre del que en podem variar fàcilment:

1. La mida del tronc
2. El nombre d'iteracions (el nombre de vegades que es repeteix la separació en branques)
3. El nombre de branques
4. La relació de mida entre les primeres branques i el tronc
5. L'angle d'obertura total de les branques.

Aquestes variacions es poden dur a terme modificant el valor d'algunes variables modificables en la presentació del programa a través d'uns lliscadors:

Marcadors del programa que simula l'estructura d'un arbre

A la imatge anterior podem veure els marcadors, botons i lliscadors del programa.
Primer de tot tenim els ja familiars botons de neteja i preparació, i el d'aplicar l'operació. A sota podem veure els lliscadors cada un per a cada variable modificable i seguidament uns marcadors que expressen el nombre de branques de l'últim nivell (**branquetes**) i la dimensió de la figura.
(Veureu que els lliscadors tenen un nom que degut als requisits del programa no porta ni

accents ni caràcters especials ni espais).

Cada un dels lliscadors fa la seva funció dins la instrucció:

1. Número d'iteracions: Indica el nombre de vegades que es farà la separació en branques des del tronc fins les branques de l'ultim nivell o el nombre de nivells de branques sense contar el tronc.

2. Número de branques: Indica el nombre de branques de la figura (el nombre de branques creades en cada separació).

3. Mida del tronc: Indica la mida de la distància que es recorre abans de la primera separació en branques.

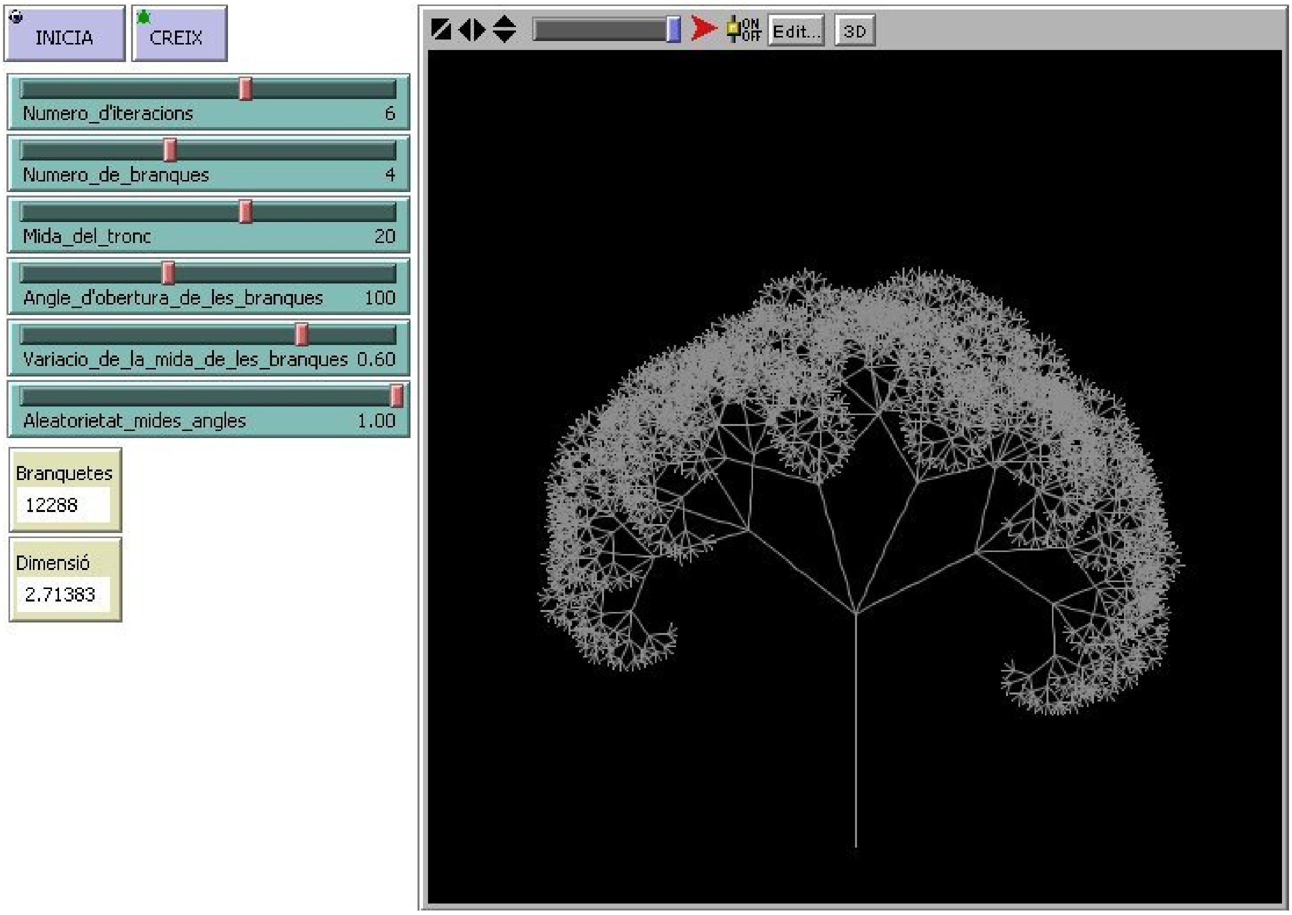
4. Angle d'obertura de les branques: Indica (en graus) l'angle d'obertura entre la primera branca i la última. Totes les altres es van col·locant ordenadament entre mig.

5. Variació de la mida de les branques: Indica el nombre per el que multiplicarem la mida inicial del tronc per tal d'obtenir la mida de les branques del primer nivell. I alhora, el nombre per el que multiplicarem la mida d'una branca per aconseguir la mida de les branques d'un nivell més alt.

6. Aleatorietat de mides i angles: Indica el grau d'aleatorietat, dins una límits, de la figura tant pel que fa a mides com pel que fa a angles. Si aquest lliscador marca 0, l'aleatorietat és nul·la i la figura surt perfectament simètrica però si la aleatorietat augmenta, els angles i les mides tenen un petit marge de modificació que canvia aleatòriament per cada mida, cada angle i que li dóna a la figura un aspecte més natural i no tant simètric.

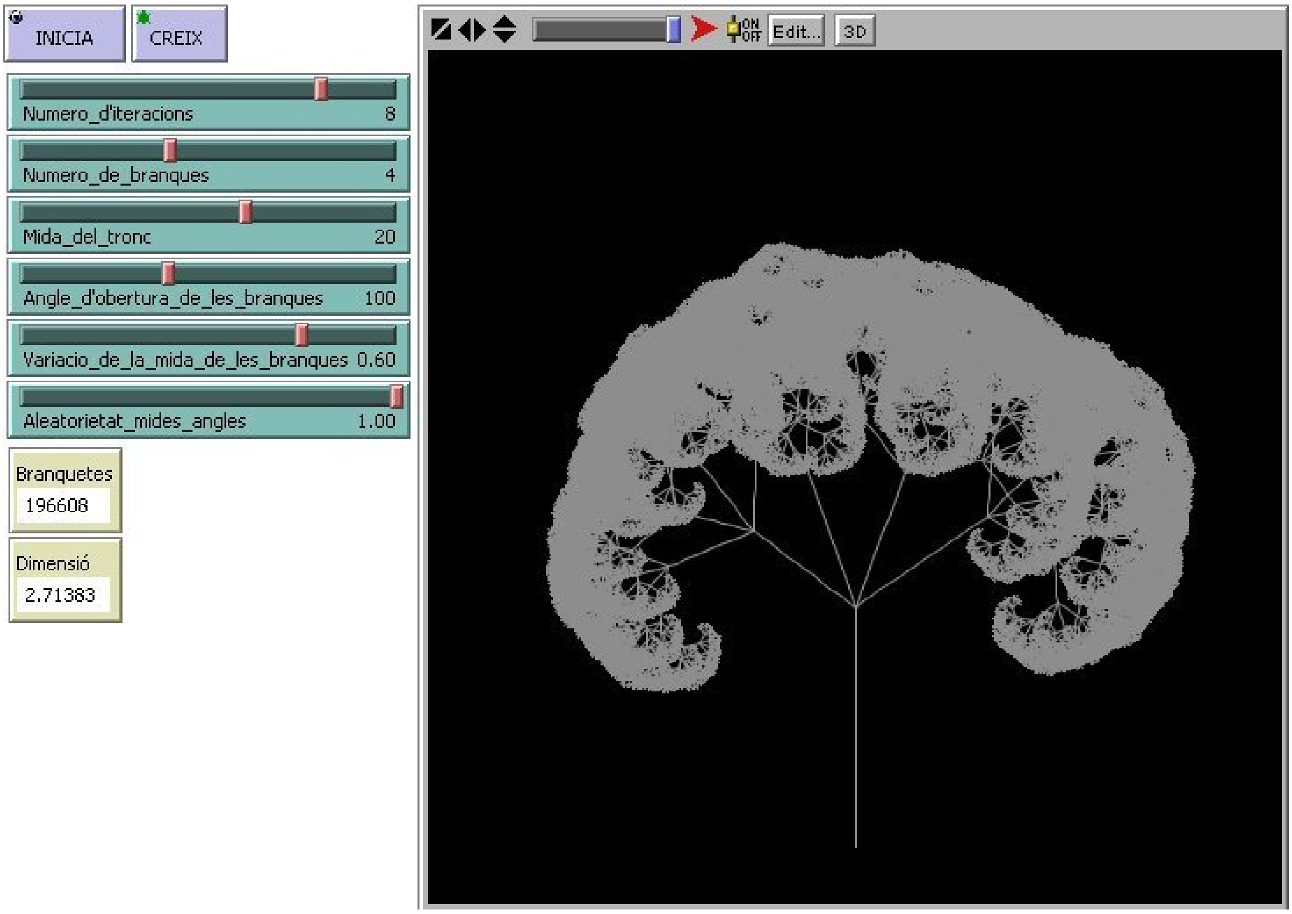
Podem comprovar el resultat de modificar cada un dels lliscadors:

Tenim la imatge mare, resultat dels valors representats en la imatge anterior:

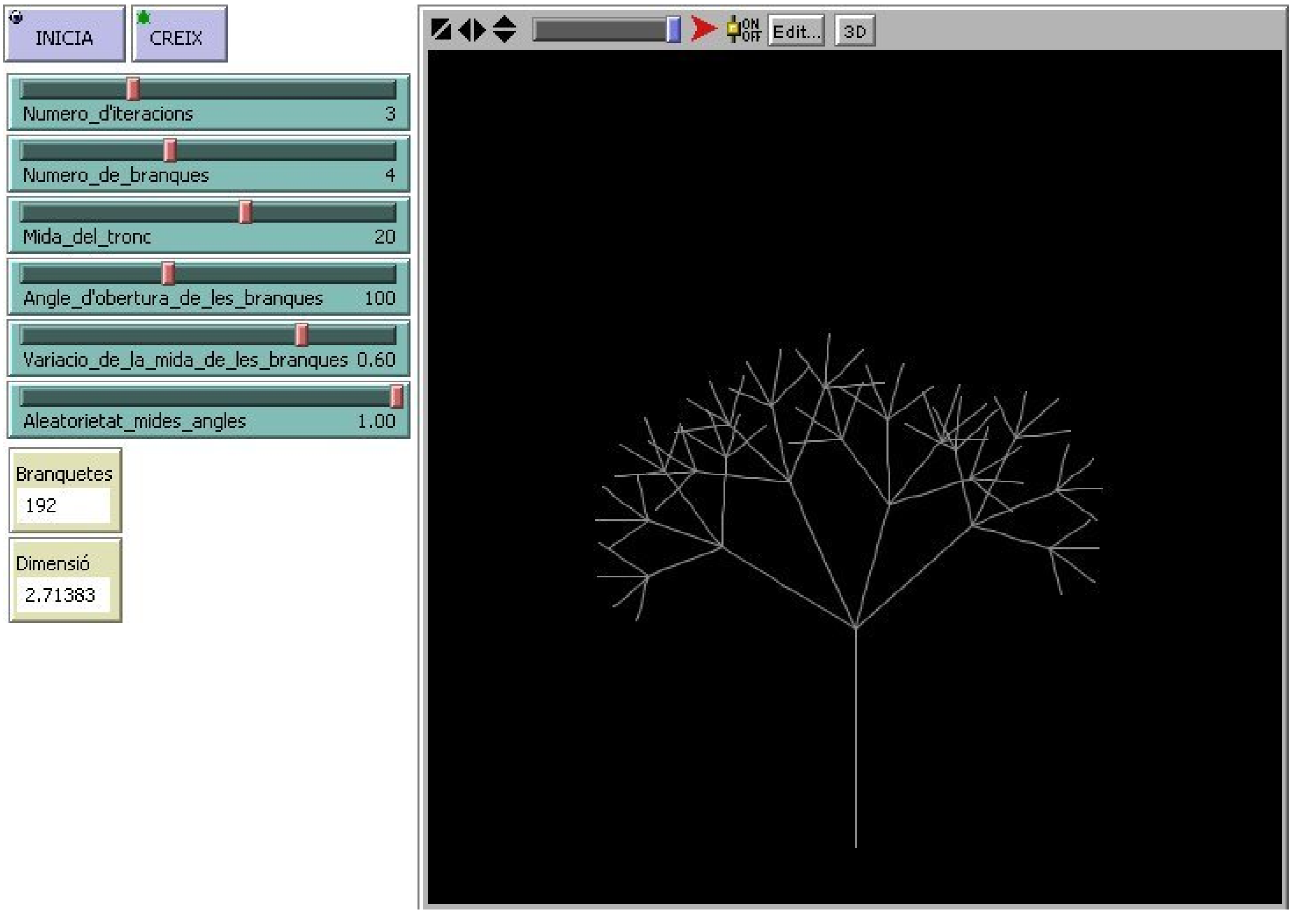


Podem modificar el lliscador de les iteracions i observar-ne el canvi:

Amb 8 iteracions:

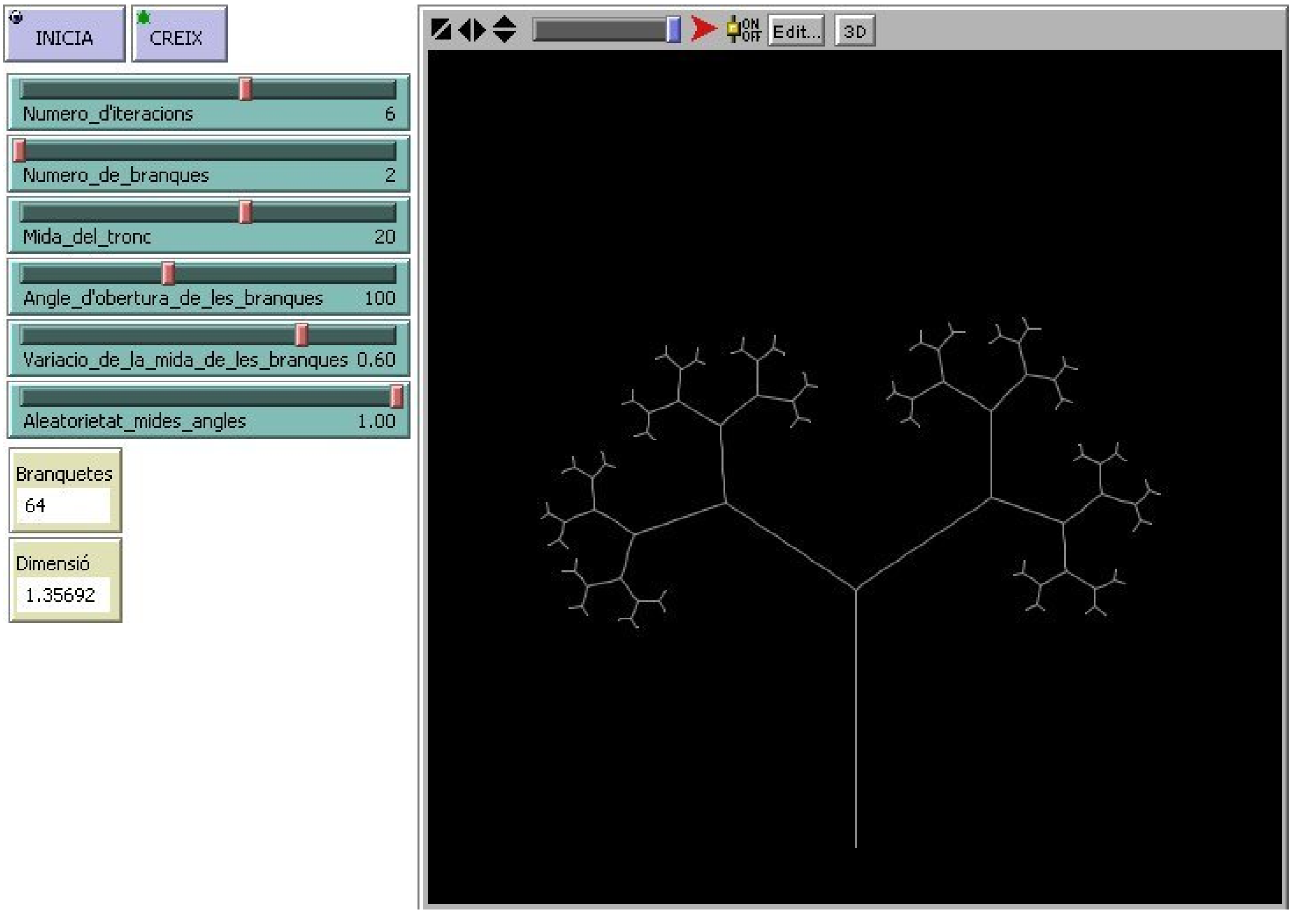


Amb 3 iteracions:

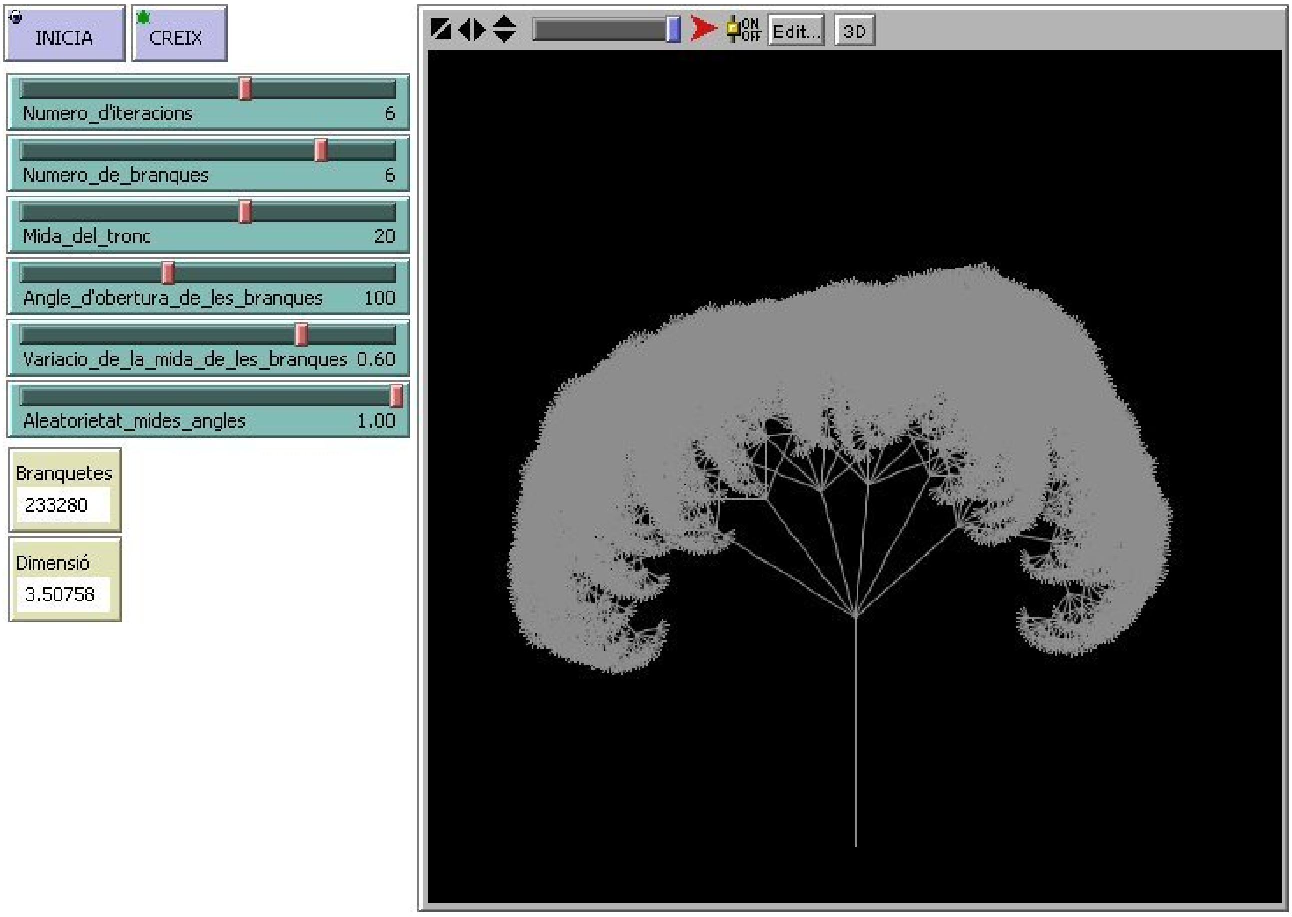


Tornem a posar el nombre d'iteracions a 6 com en la imatge mare. Si modifiquem el lliscador del nombre de branques observarem:

Amb 2 branques:

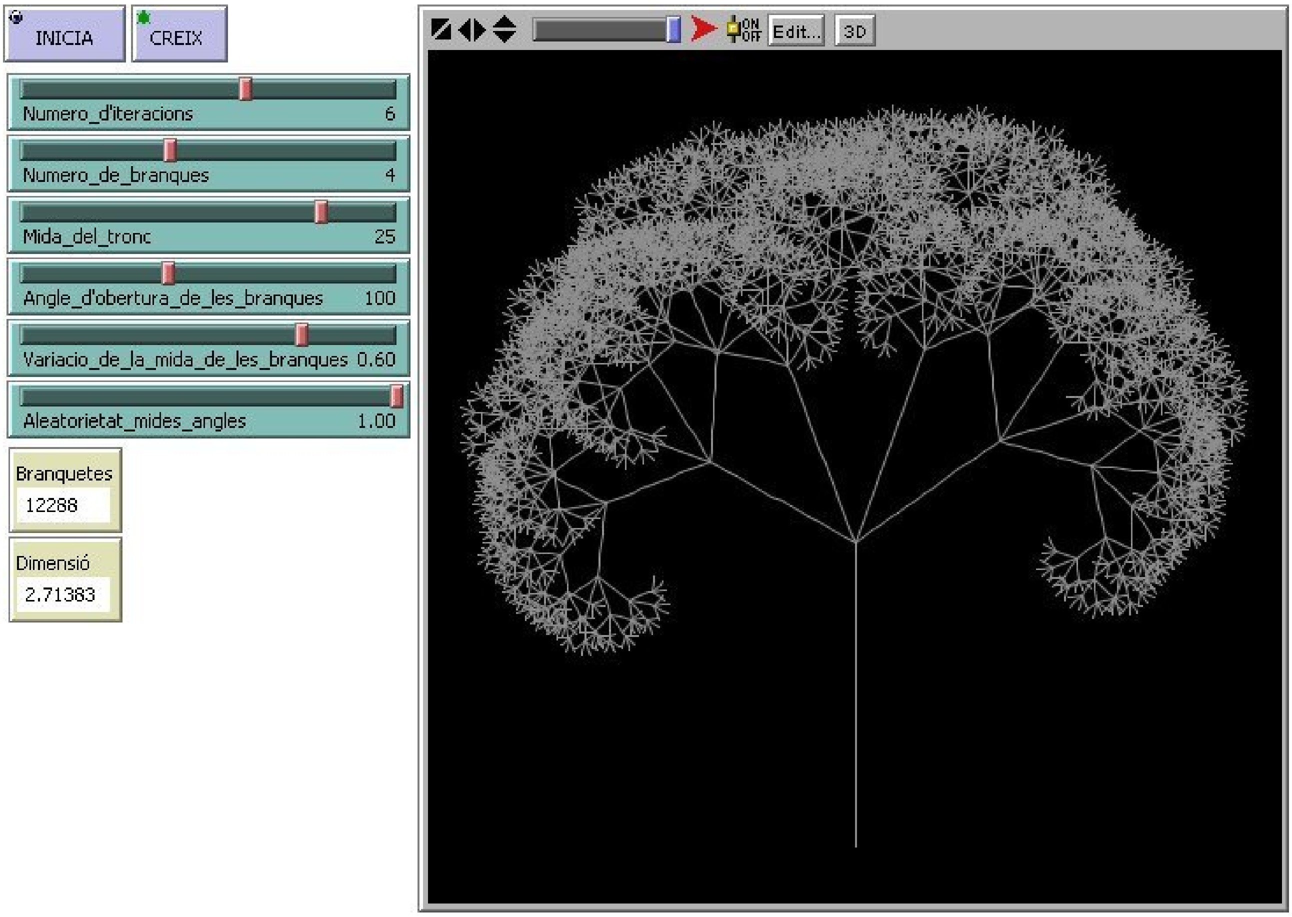


Amb 6 branques:

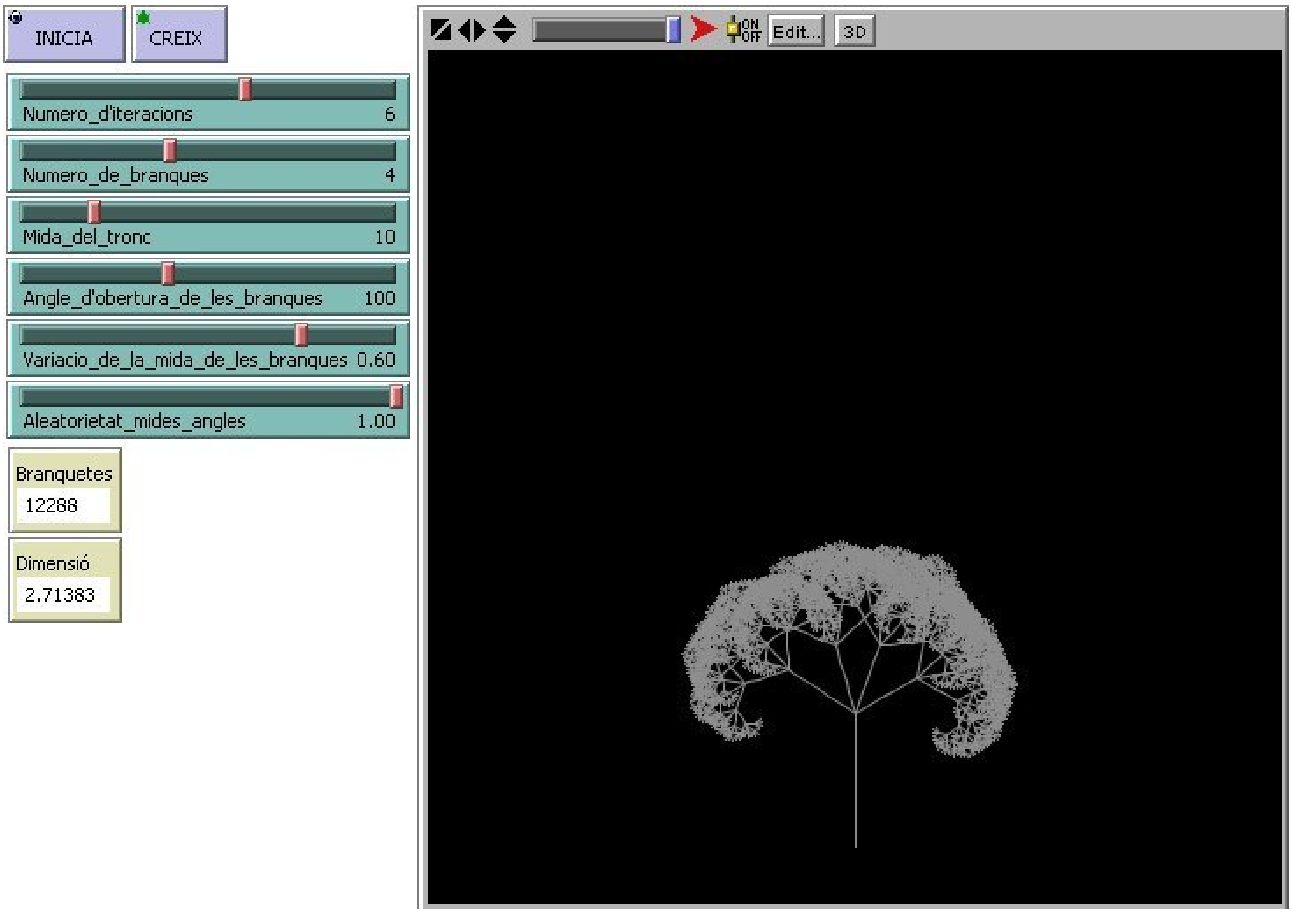


Tornem als valors inicials i observem el resultat si modifiquem la mida del tronc:

Com a mida inicial 25:

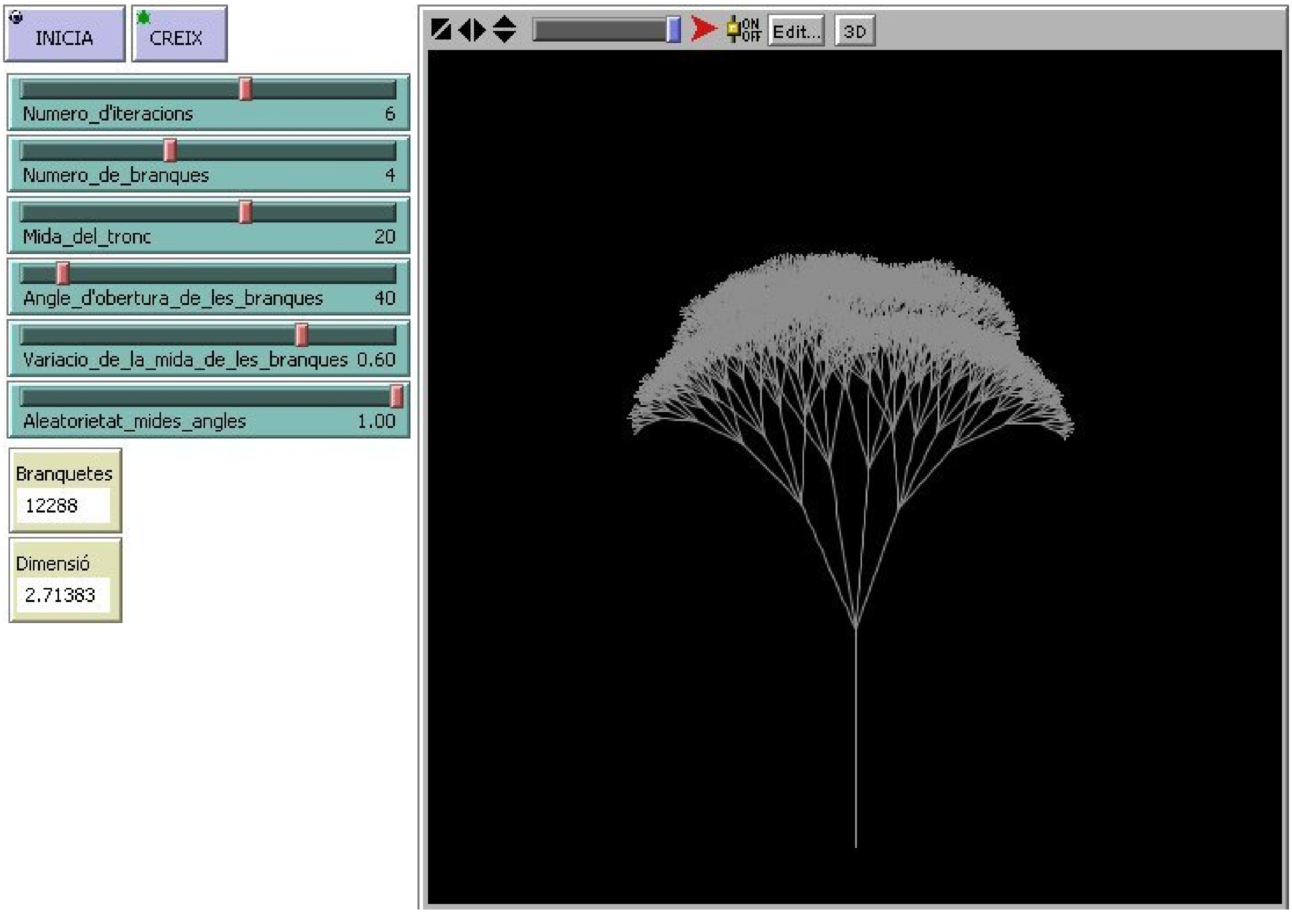


Com a mida inicial 10:

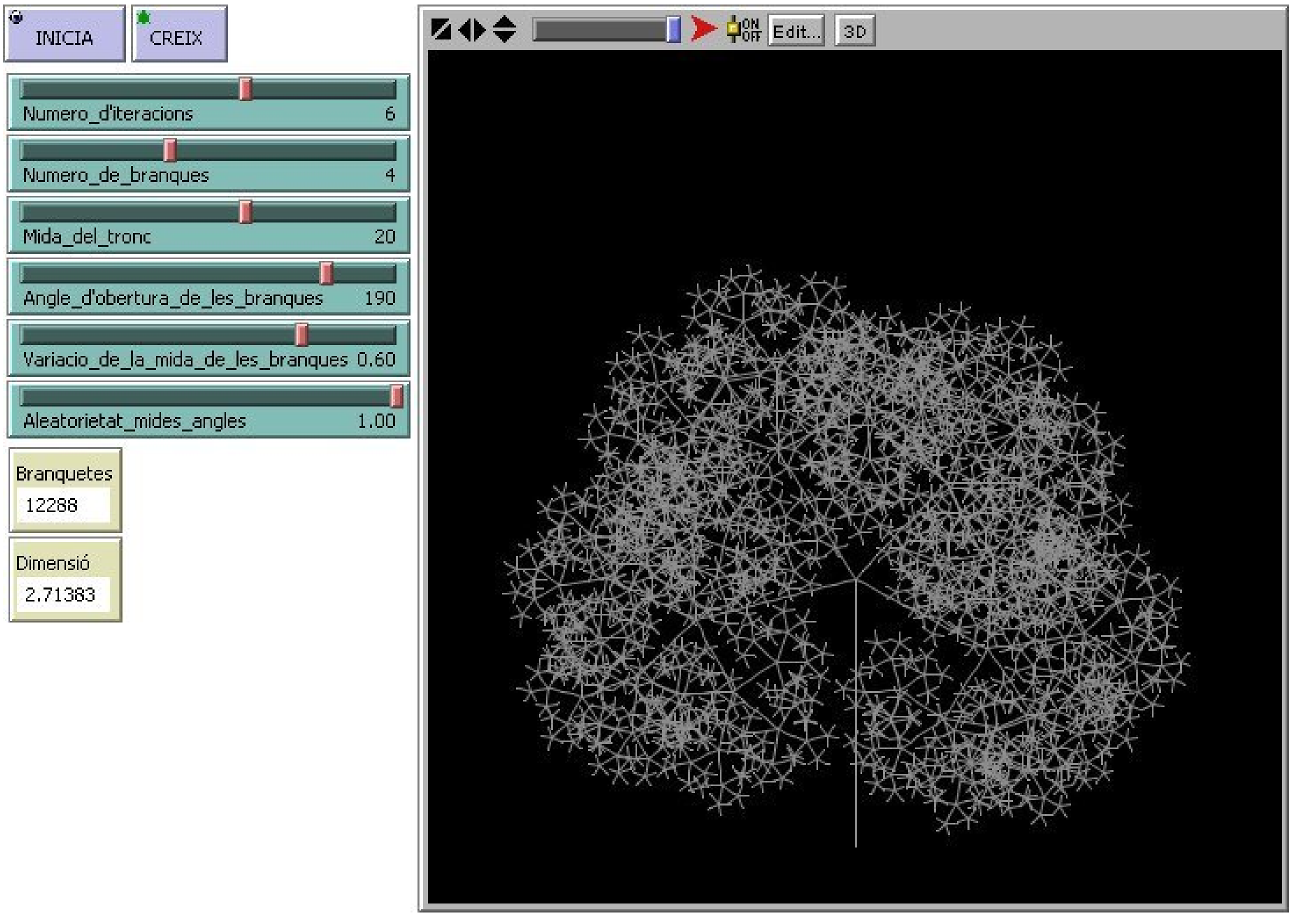


Modifiquem L'angle d'obertura enter branques:

Posem-hi 40 graus:

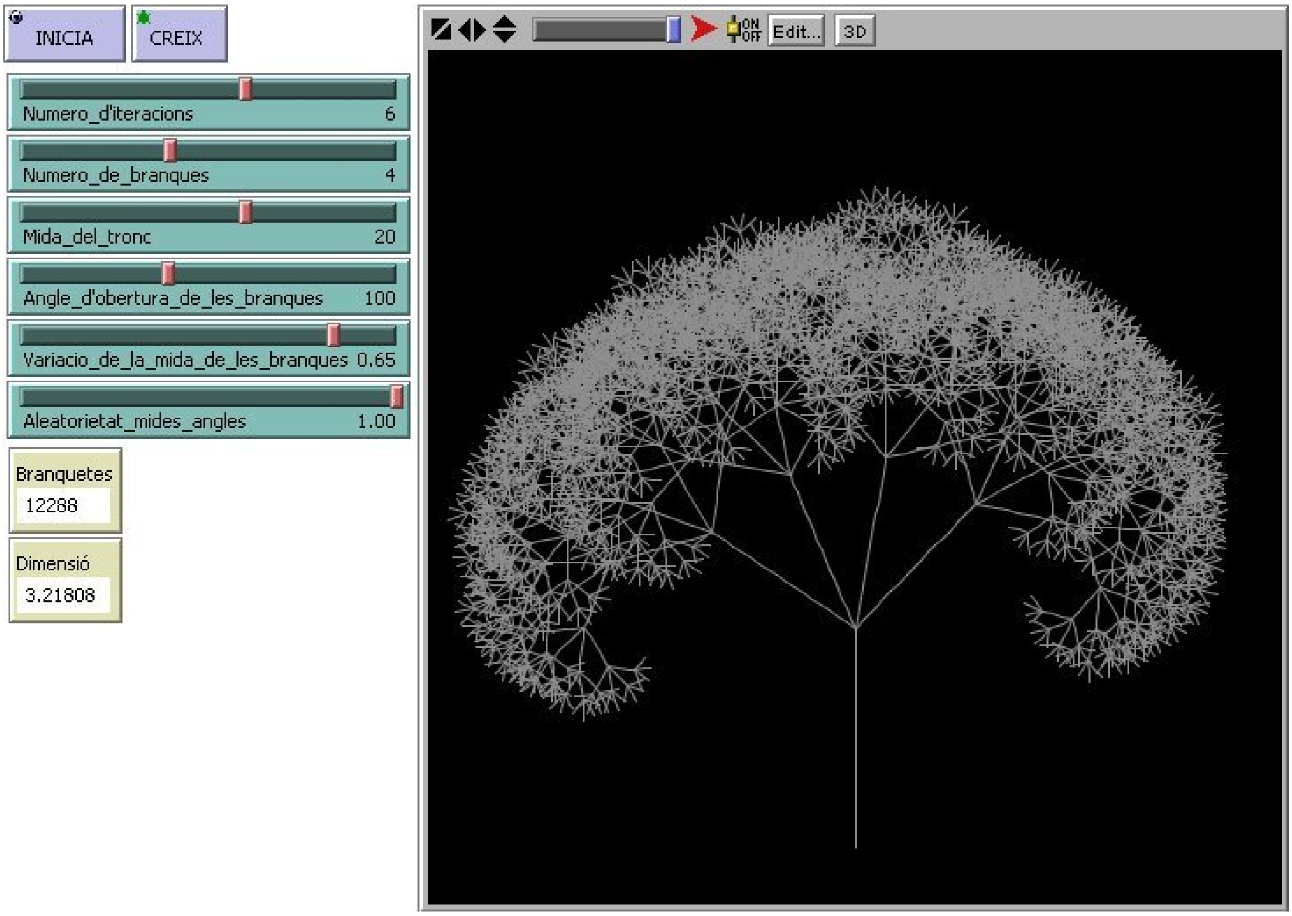


I 190 graus:

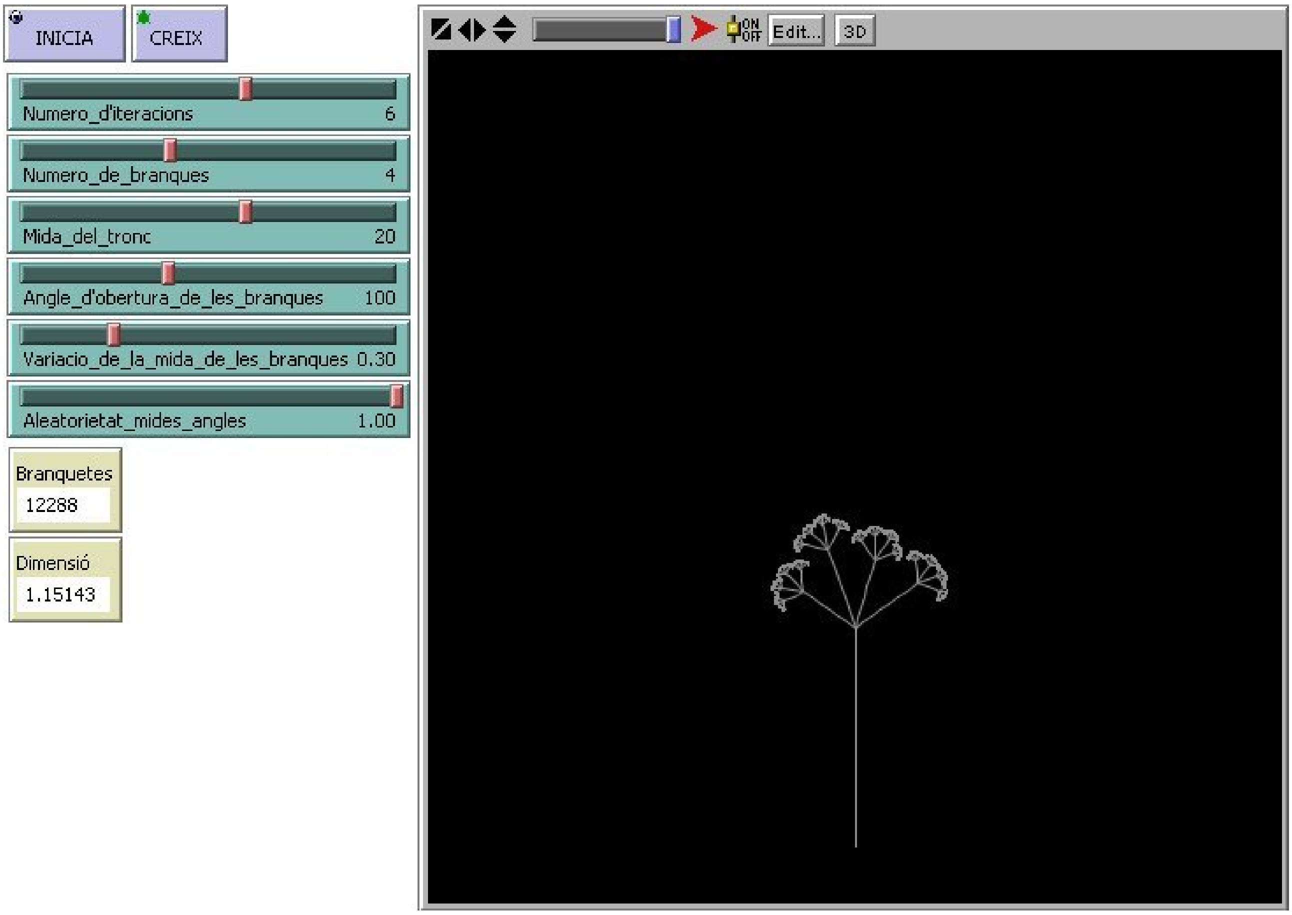


Si modifiquem la variació de la mida de les branques observem:

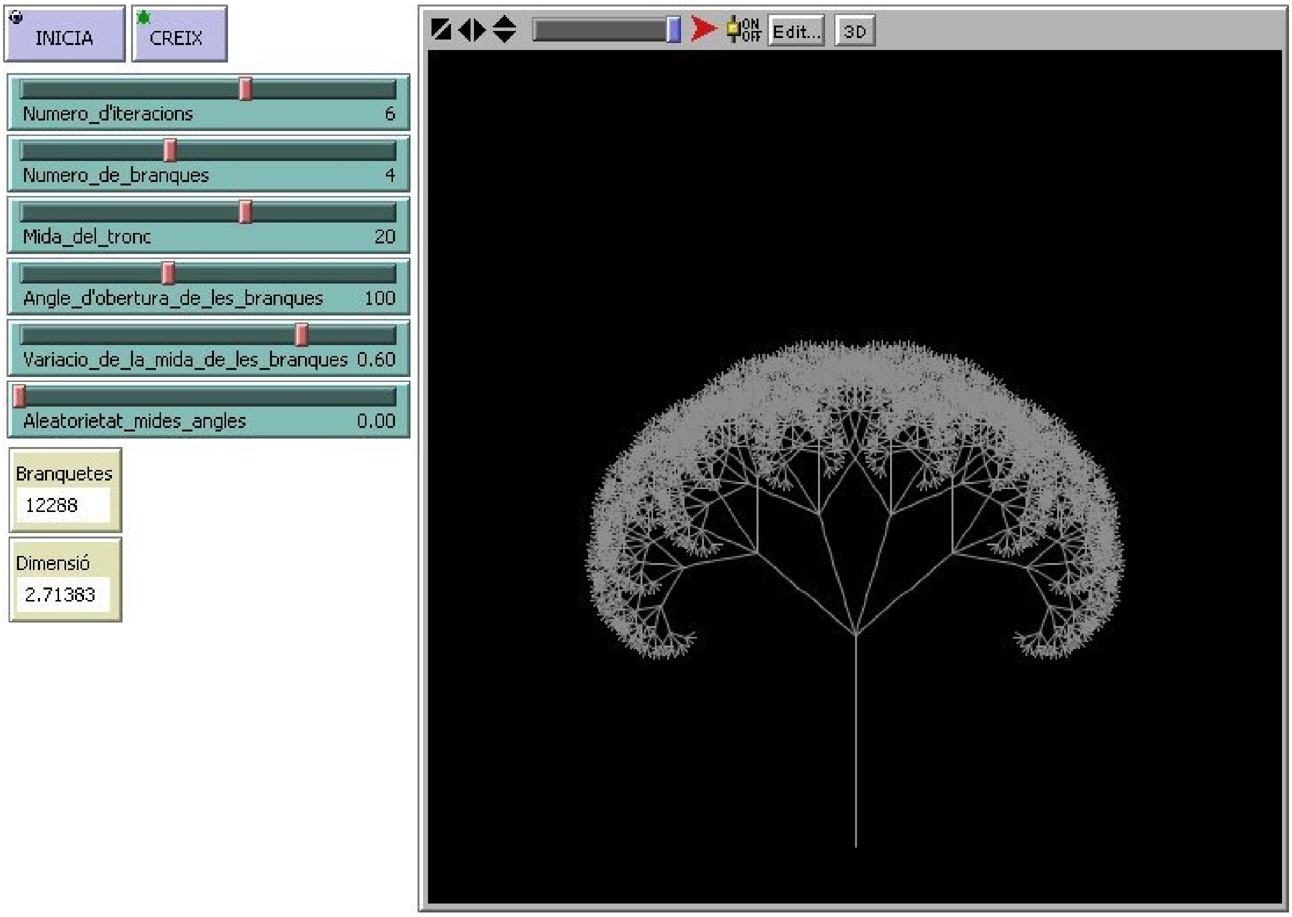
Amb el valor 0,65:



Amb el valor 0,30:



Per últim, modificarem la aleatorietat que hem tingut posada al màxim i que provoca irregularitats en les formes. La posarem al mínim, 0 de manera que no hi haurà gens d'aleatorietat, la figura serà totalment simètrica.



La dimensió en aquest programa es calcula segons el nombre de branques i la variació de la mida de les branques:

$$D=\frac{(\ln(\textit{Número de branques}))}{(\ln(\frac{1}{(\textit{Variació mida})}))}$$

La **N** és el nombre d'objectes dins la figura total sobre els quals també apliquem la iteració. (Número de branques)

La **S** és el quocient entre una mesura de la figura total i la mateixa mesura d'un objecte de dins la figura total sobre el que també apliquem la iteració. La relació entre aquestes dos mides és la variació de la mida de les branques. Quan la mida inicial és 1, la mida de les primeres branques és la variació de la mida de les branques. Així, podem posar S com a 1/Variació de la mida de les branques.

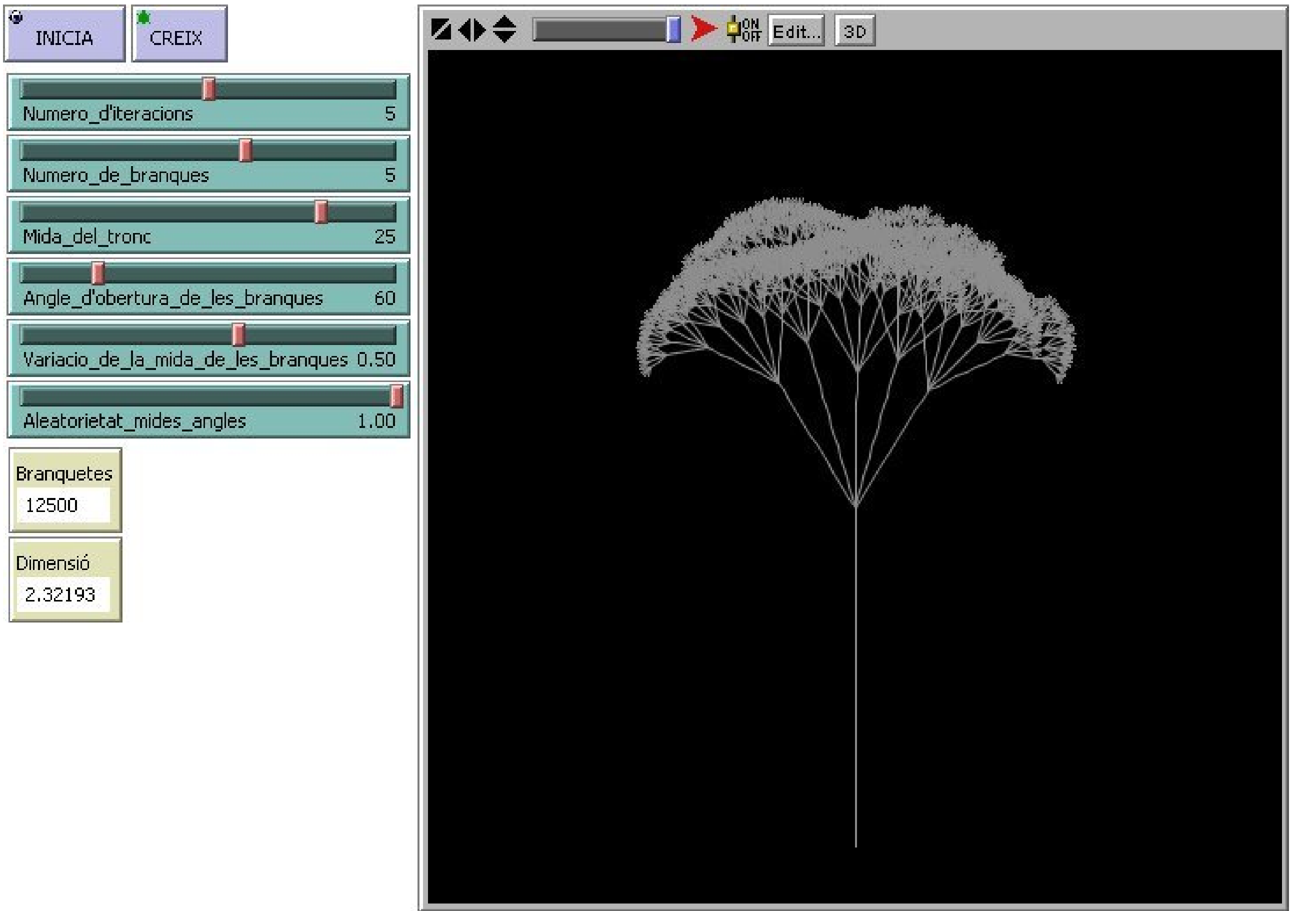
El mètode fet servir en aquest programa és diferent als que ja coneixem i consisteix en una

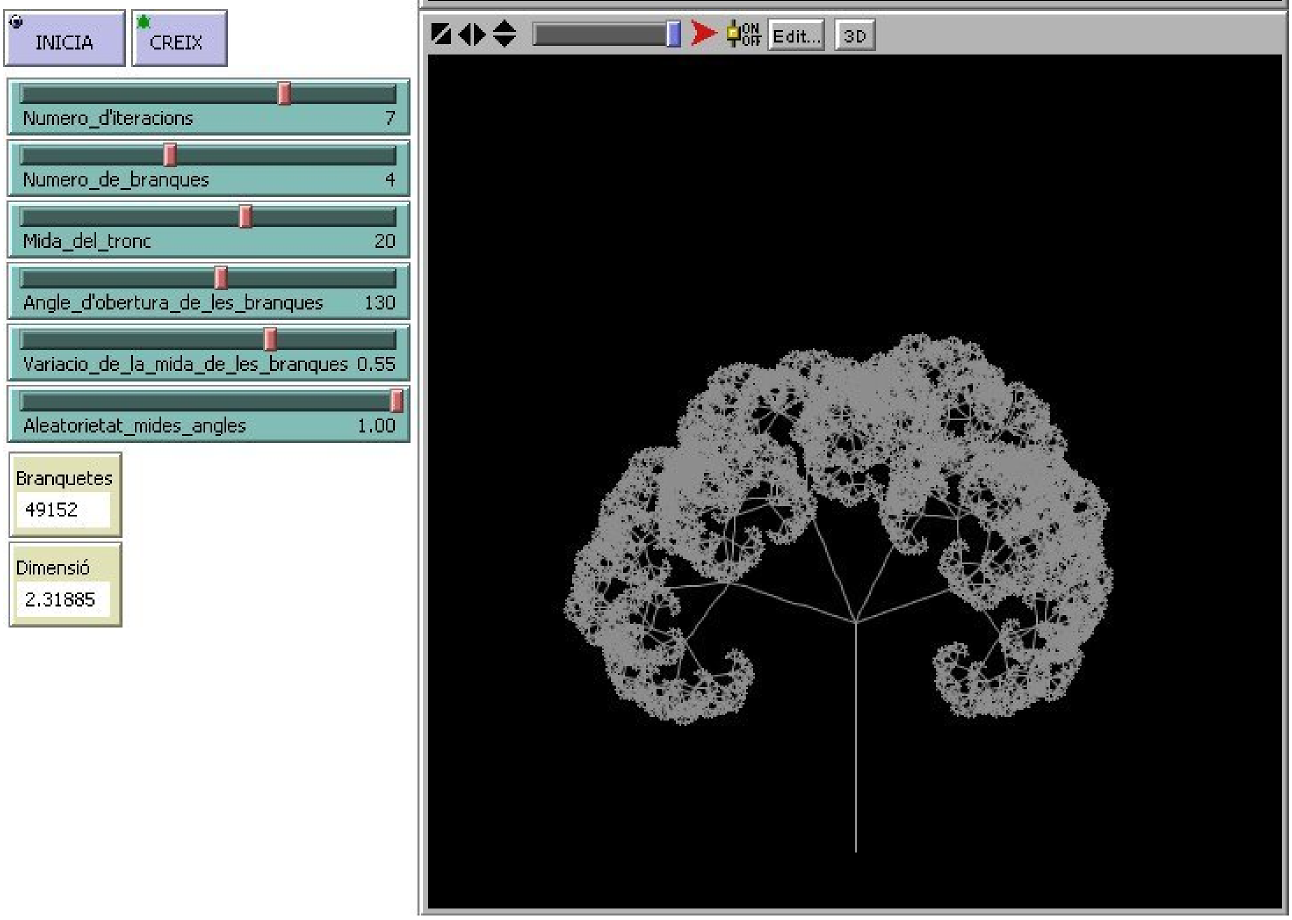
instrucció que es crida a ella mateixa. És a dir, la tortuga inicial, que es troba a l'inici inferior del tronc, avança la mida inicial i reparteix tortugues allà on toca fer un branca que fan el mateix procediment en una altre escala però dins mateix de la instrucció. Així, la figura no es separa en diferents iteracions sinó que les fabrica totes de cop, hi ha una sola imatge per cada valor. Hi ha un comptador que atura l'execució de la instrucció quan arriba a un límit, en el nostre cas, el nombre d'iteracions.
En el món de la programació, aquest mètode de auto executar-se, s'anomena recursivitat.

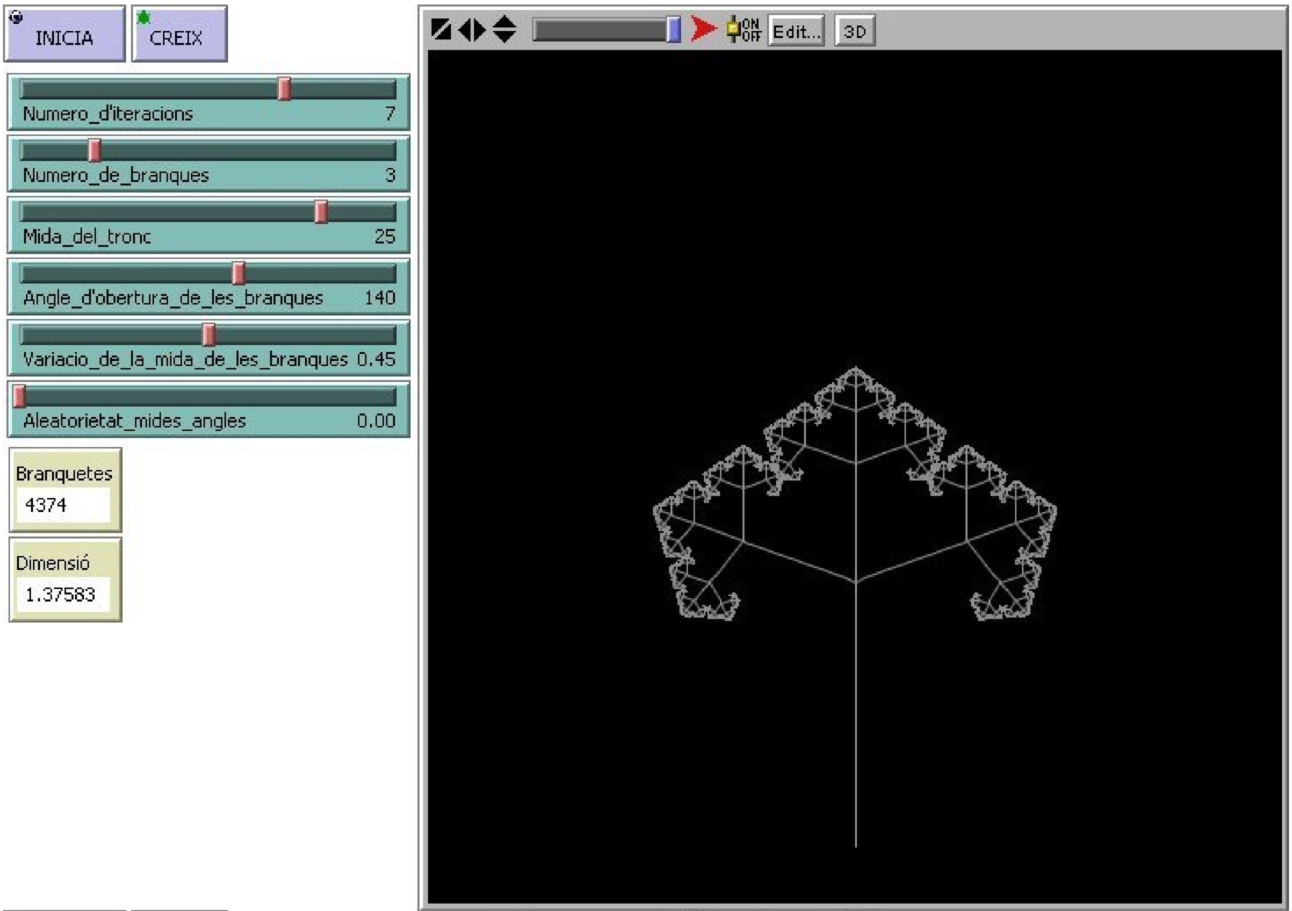
Les instruccions que rep el programa són:

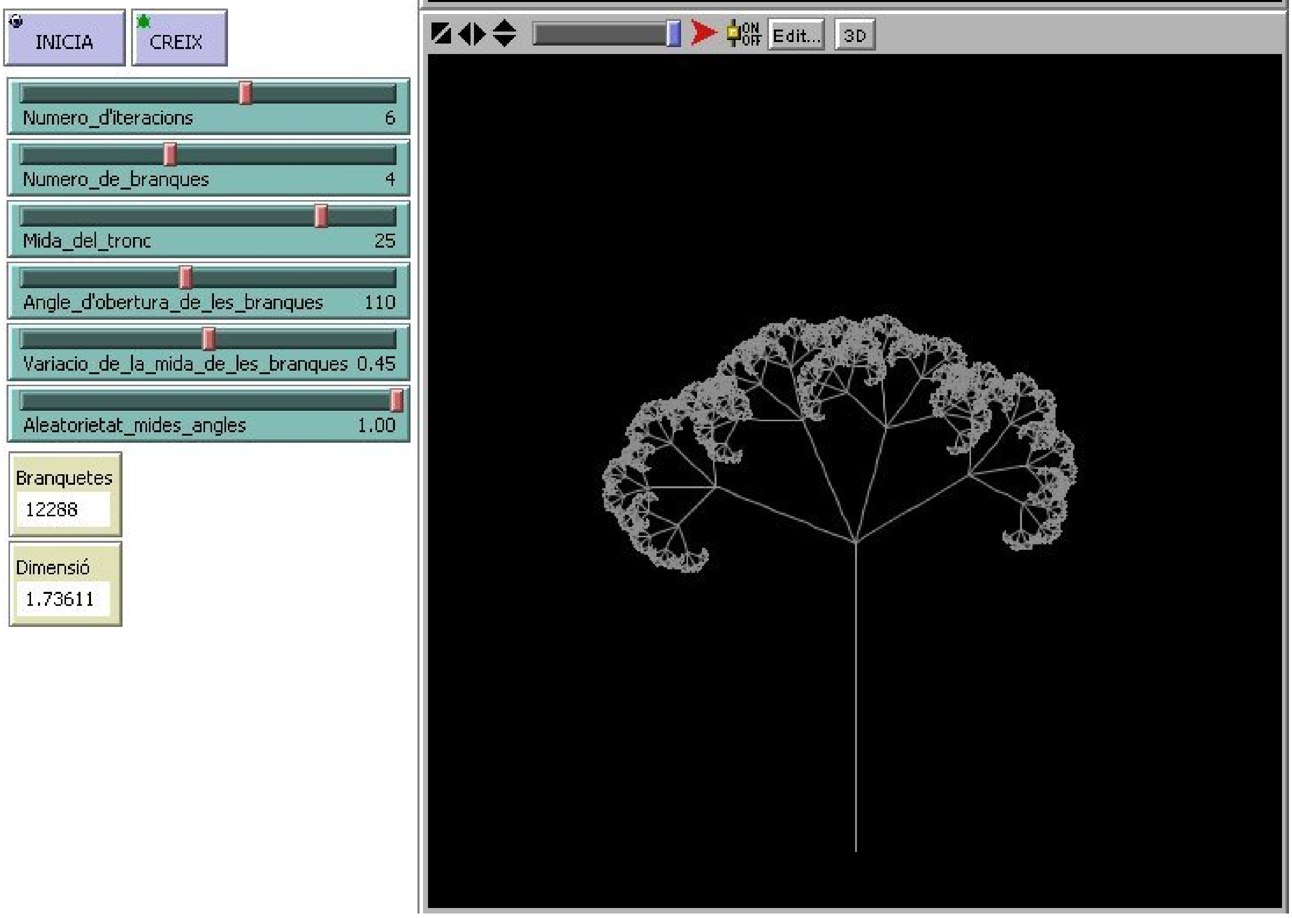
```
to INICIA
 clear-all
 create-turtles 1
 ask turtles [ setxy 0 min-pycor + 5
        hide-turtle]
end

to CREIX
 pen-down
 CAMI Mida_del_tronc 0
end

to CAMI [mida comptador]
 if comptador <= Numero_d'iteracions [
  fd mida * (1 + (random (Aleatorietat_mides_angles * 20)) / 60 )
  lt (Angle_d'obertura_de_les_branques / 2 )* (1 + (random (Aleatorietat_mides_angles *
20)) / 80 )
  repeat Numero_de_branques - 1 [
    hatch 1 [CAMI mida * Variacio_de_la_mida_de_les_branques comptador + 1]
    rt ( Angle_d'obertura_de_les_branques / ( Numero_de_branques - 1 ))* (1 + (random
(Aleatorietat_mides_angles * 20)) / 80 ) ]
  CAMI mida * Variacio_de_la_mida_de_les_branques comptador + 1
  die]
end
```

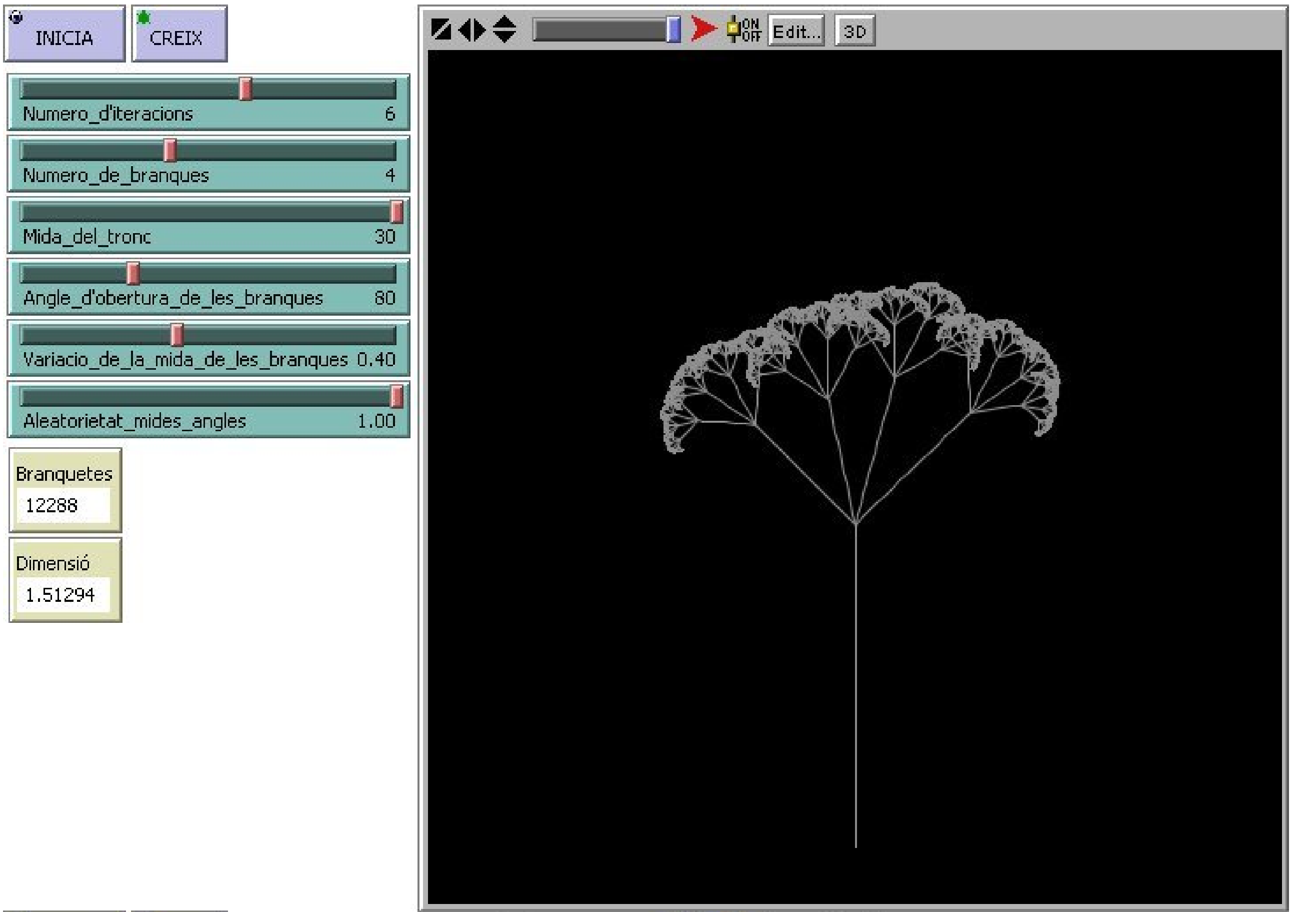
A continuació hi ha alguns exemples d'imatges de les que poden ser creades pel programa:

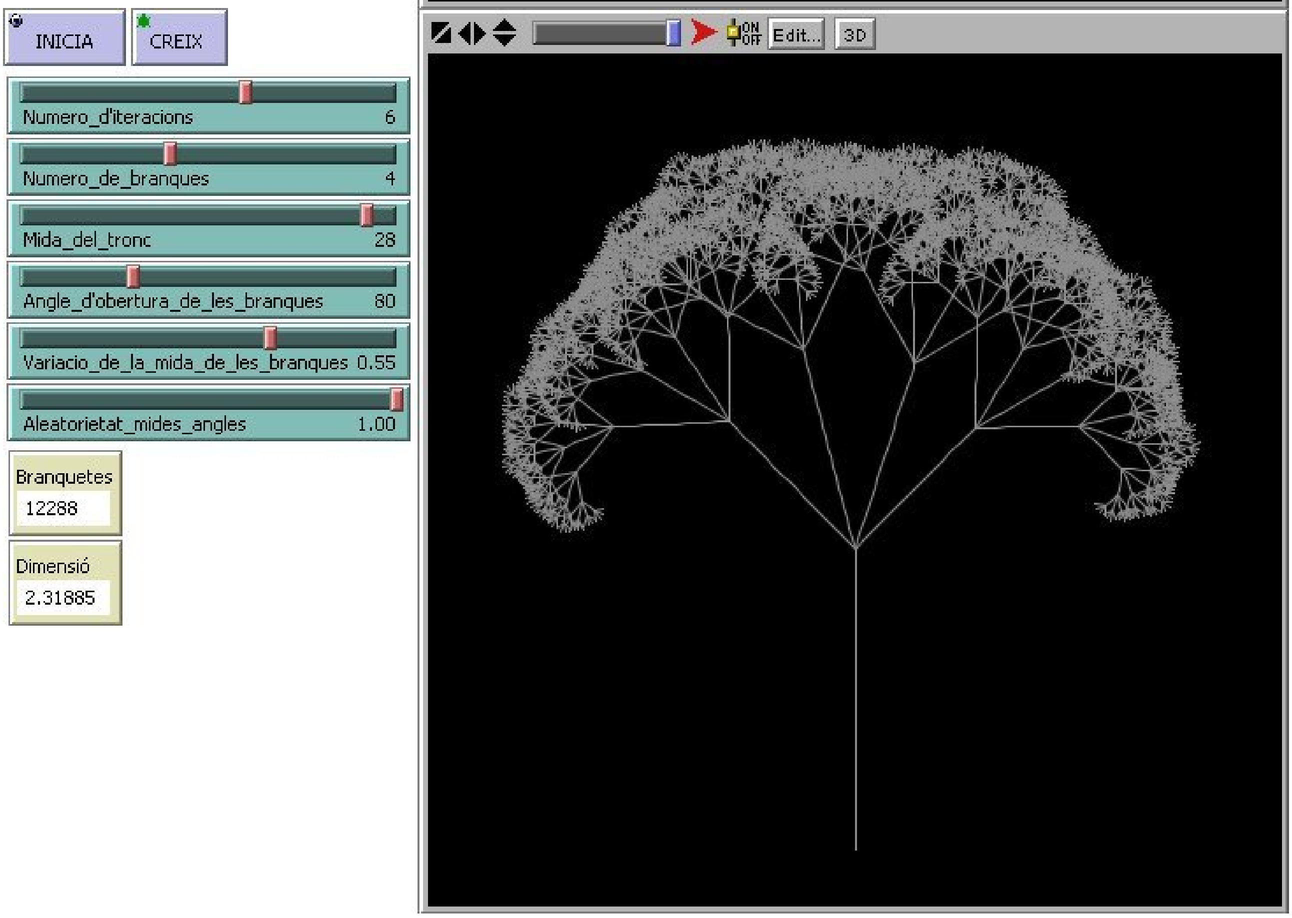

INICIA
CREIX
Numero_d'iteracions 5
Numero_de_branques 5
Mida_del_tronc 25
Angle_d'obertura_de_les_branques 60
Variacio_de_la_mida_de_les_branques 0.50
Aleatorietat_mides_angles 1.00
Branquetes
12500
Dimensió
2.32193
Edit...
3D

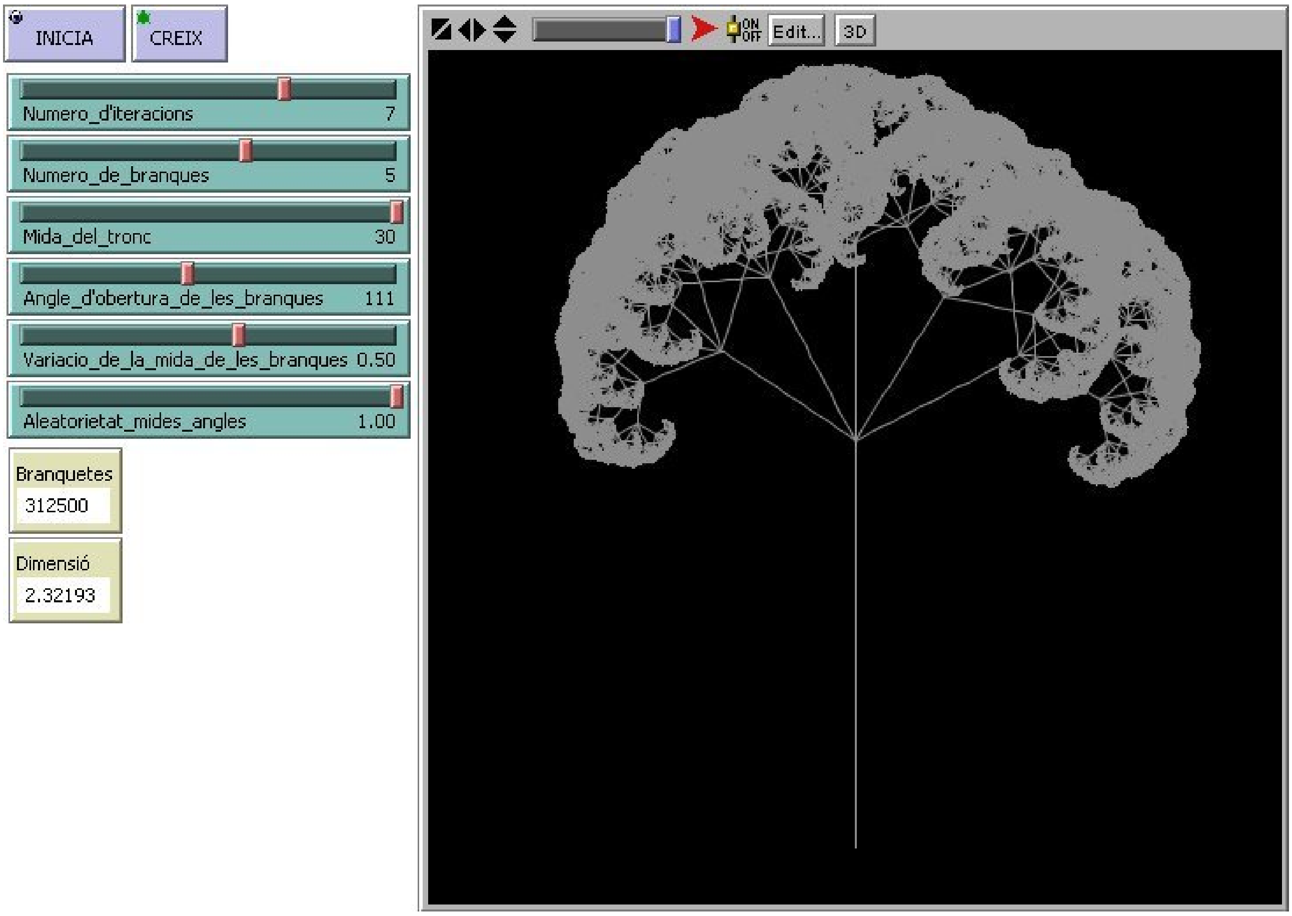

INICIA
CREIX
Numero_d'iteracions 7
Numero_de_branques 4
Mida_del_tronc 20
Angle_d'obertura_de_les_branques 130
Variacio_de_la_mida_de_les_branques 0.55
Aleatorietat_mides_angles 1.00
Branquetes
49152
Dimensió
2.31885
Edit...
3D


INICIA
CREIX
Numero_d'iteracions 7
Numero_de_branques 3
Mida_del_tronc 25
Angle_d'obertura_de_les_branques 140
Variacio_de_la_mida_de_les_branques 0.45
Aleatorietat_mides_angles 0.00
Branquetes
4374
Dimensió
1.37583
Edit...
3D


INICIA
CREIX
Numero_d'iteracions 6
Numero_de_branques 4
Mida_del_tronc 25
Angle_d'obertura_de_les_branques 110
Variacio_de_la_mida_de_les_branques 0.45
Aleatorietat_mides_angles 1.00
Branquetes
12288
Dimensió
1.73611
Edit...
3D


INICIA
CREIX
Numero_d'iteracions 6
Numero_de_branques 4
Mida_del_tronc 30
Angle_d'obertura_de_les_branques 80
Variacio_de_la_mida_de_les_branques 0.40
Aleatorietat_mides_angles 1.00
Branquetes
12288
Dimensió
1.51294
Edit...
3D


INICIA
CREIX
Numero_d'iteracions 6
Numero_de_branques 4
Mida_del_tronc 28
Angle_d'obertura_de_les_branques 80
Variacio_de_la_mida_de_les_branques 0.55
Aleatorietat_mides_angles 1.00
Branquetes
12288
Dimensió
2.31885
Edit...
3D



4.5 Simulació de l'estructura d'una fulla de falguera

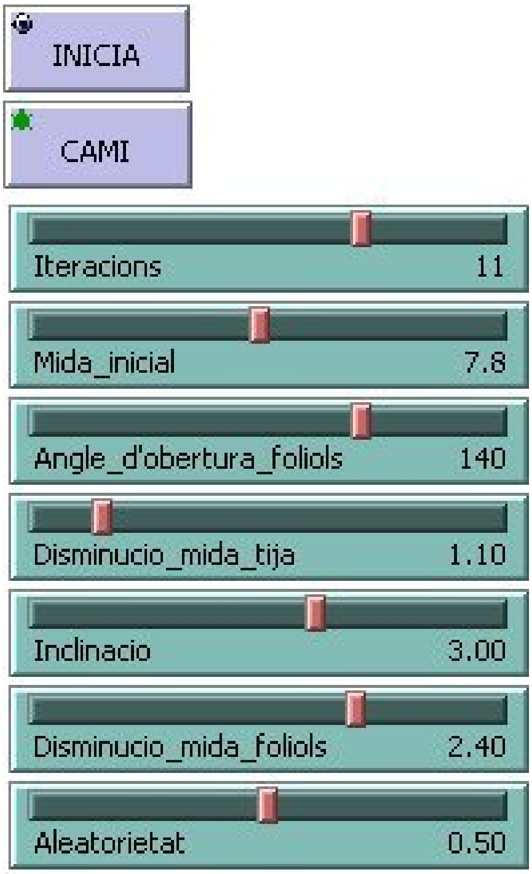
Aquesta pràctica consisteix, com l'anterior, en intentar reproduir amb el NetLogo, interpret del llenguatge Logo, una estructura natural. Aquest cop, intentar imitar l'estructura d'una fulla de falguera.

Podem observar molt fàcilment l'estructura fractal d'una falguera:

Aquesta pràctica crea un eix principal inclinat (tija) als laterals del qual s'hi formen foliols semblants a la imatge total (auto similitud).

En aquest programa, la tortuga, fa un viatge per a tota la figura dibuixant-la tota ella sola. Comença des de l'inici de la tija, avança la mida inicial, gira a l'esquerra, i fa el primer foliol, gira cap a la dreta, i fa el segon foliol, torna a girar cap a l'esquerra, i acaba de fer tota la tija repetint l'operació cada vegada que avança la separació entre una parella de foliols i la següent. Després de tot el procés, torna enrere i acaba el recorregut al punt on l'ha començat.
De fet, utilitza el mateix procediment per a fer tots els foliols i alhora, tots els laterals dels foliols encara que amb mides diferents i això li dóna la propietat fractal de l'auto similitud a la figura final.

Com en l'anterior, aquest programa també inclou lliscadors que serveixen per variar diferents característiques de la imatge. Podem modificar:



1. Les iteracions: Indiquen el nombre de ramificacions.

2. Mida inicial: La mida del tros que avancem abans de la primera ramificació.

3. Angle d'obertura dels foliols: Indica l'angle de separació entre foliols.

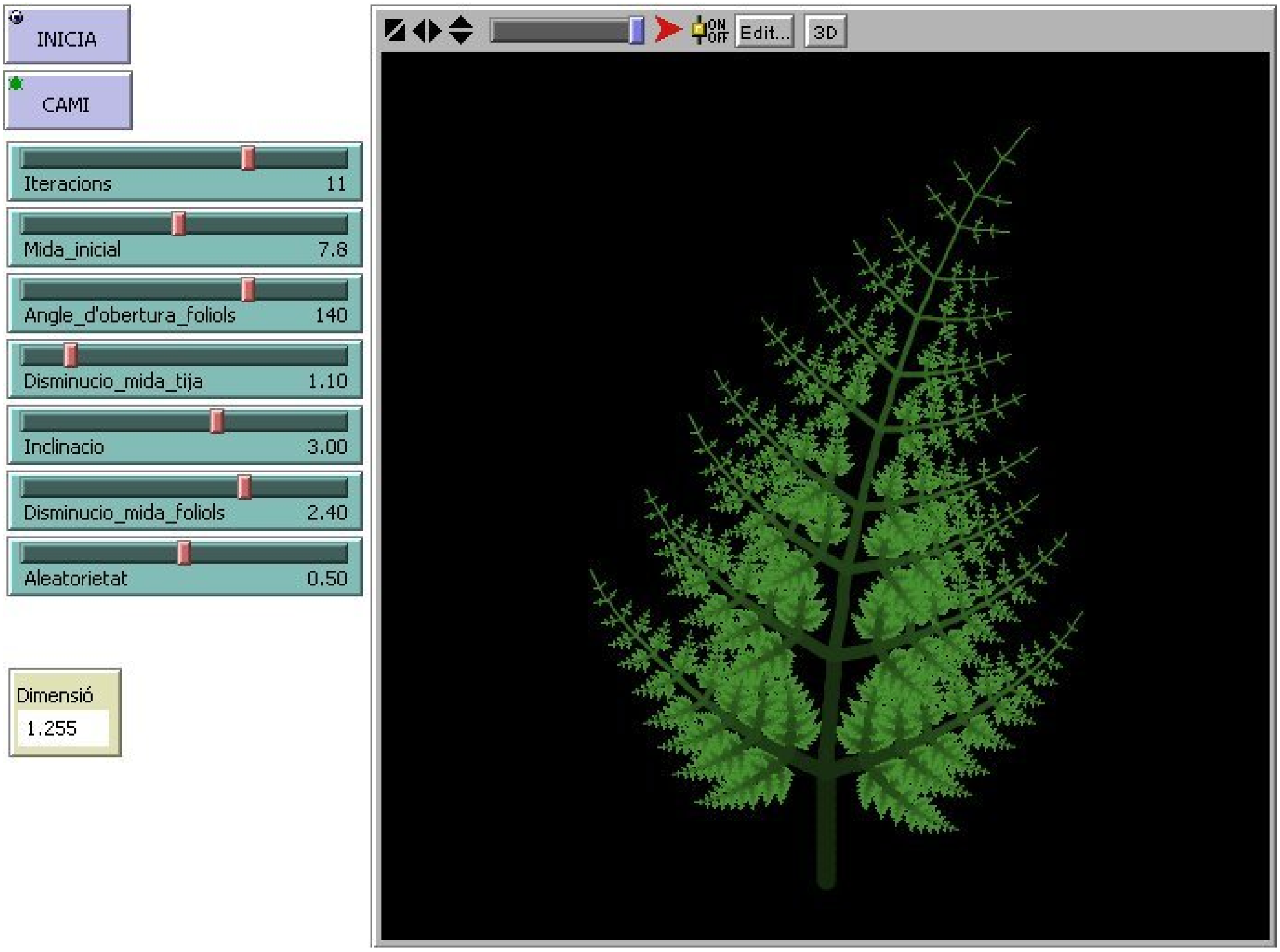
4. Disminució de la tija: Indica el nombre per el que dividirem a la mida de la tija de l'anterior nivell per a obtenir la mida de la tija.

5. Inclinació: Són els graus d'inclinació de la tija cada vegada que fa una ramificació.

6. Disminució de la mida dels foliols: És el nombre per el qual dividiràs la mida de la tija per a aconseguir la mida de la primera part de tija dels foliols de la següent ramificació.

7. Aleatorietat: Provoca una alteració de les mides i dels angles. Si el valor és 0 l'aleatorietat és nul·la.

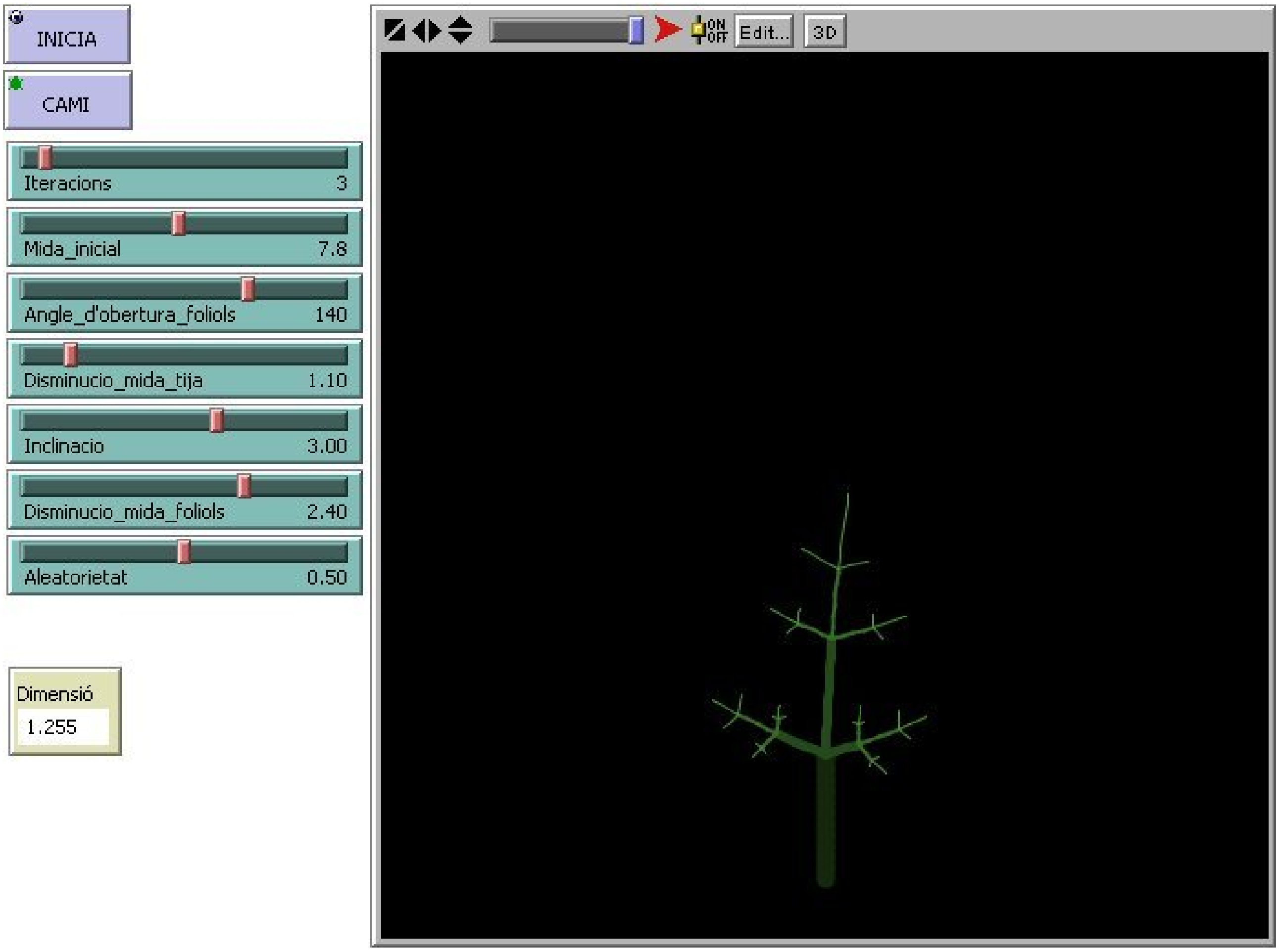
La imatge corresponent a aquests valors és:



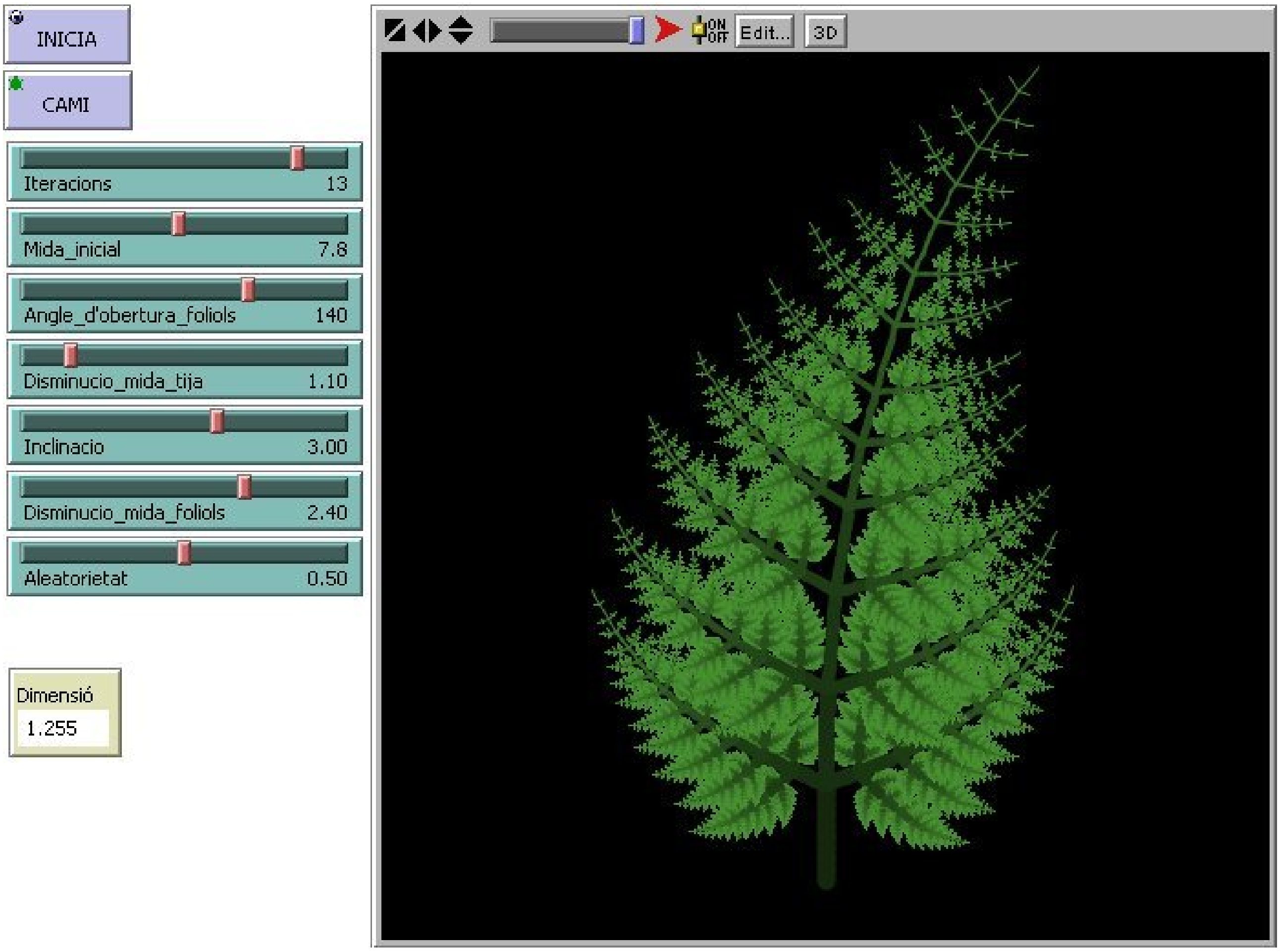
Tenim una imatge i podem comprovar com, variant les diferents variables, varia la imatge.

Variem el nombre d'iteracions:

El comptador a 3 iteracions:



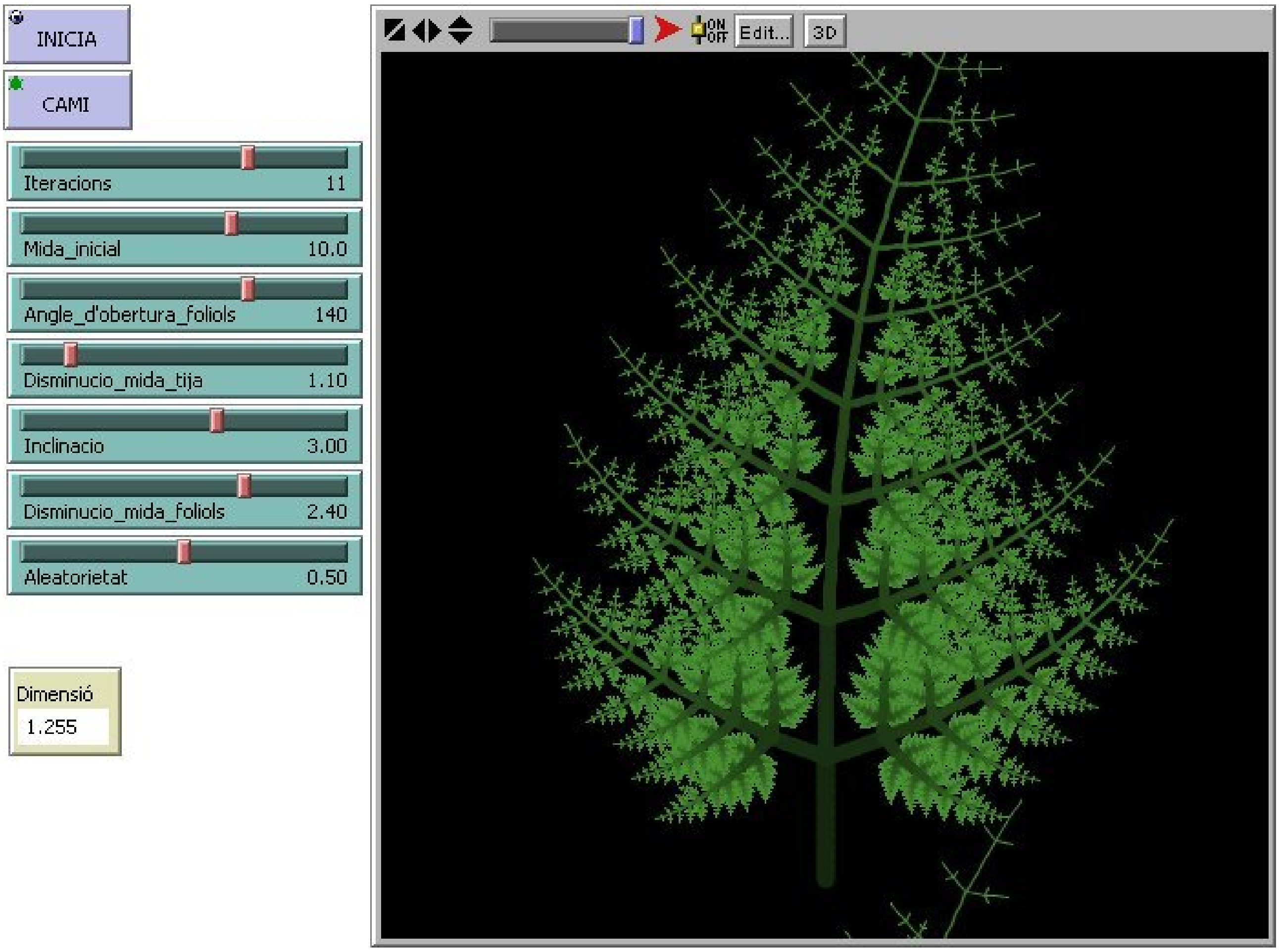
El comptador a 13 iteracions:



Variem la mida inicial:

Quan la mida és igual a 2,5 obtenim:

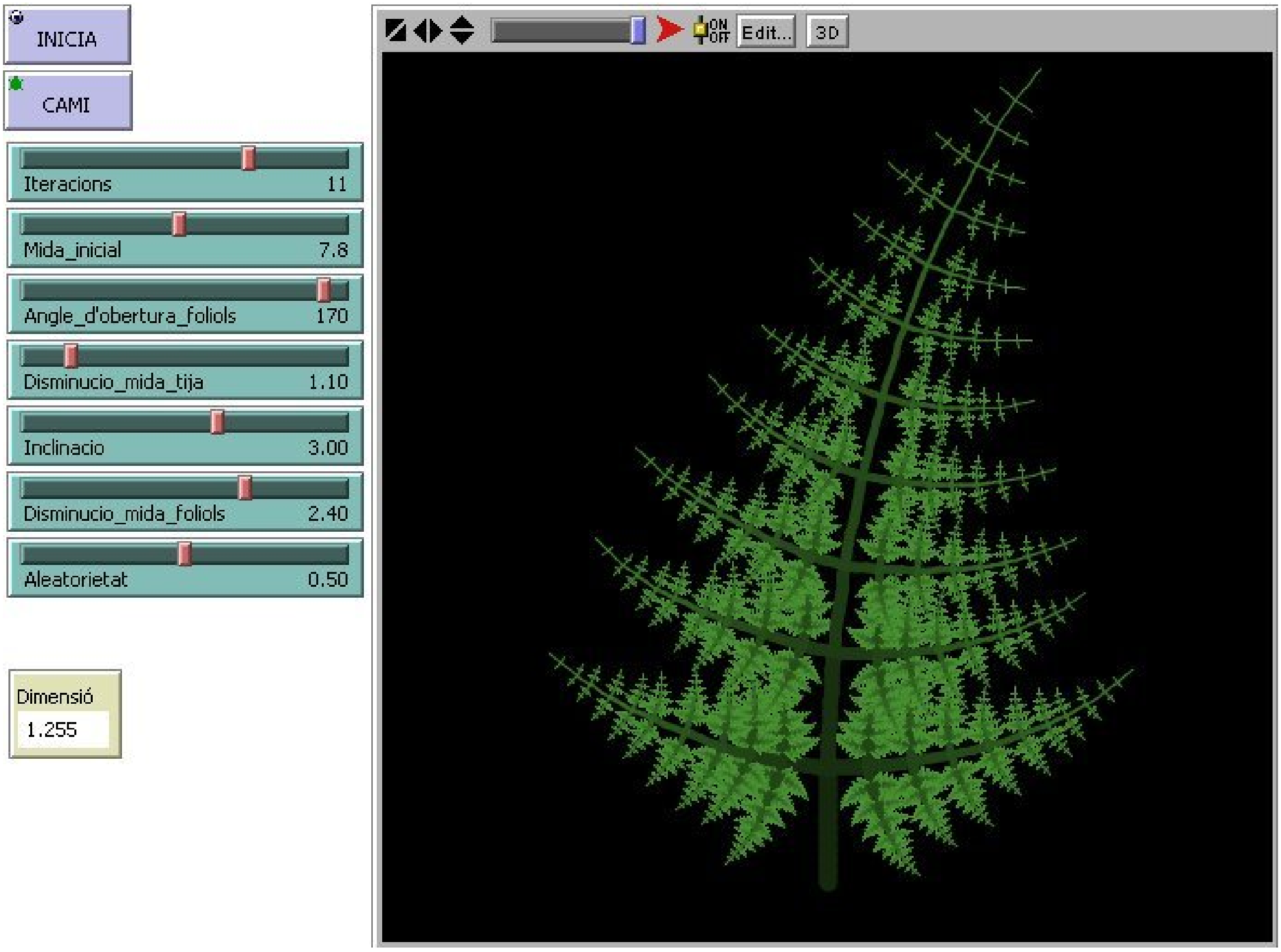
Quan la mida és igual a 10 obtenim:



Variem l'angle d'obertura dels foliols:

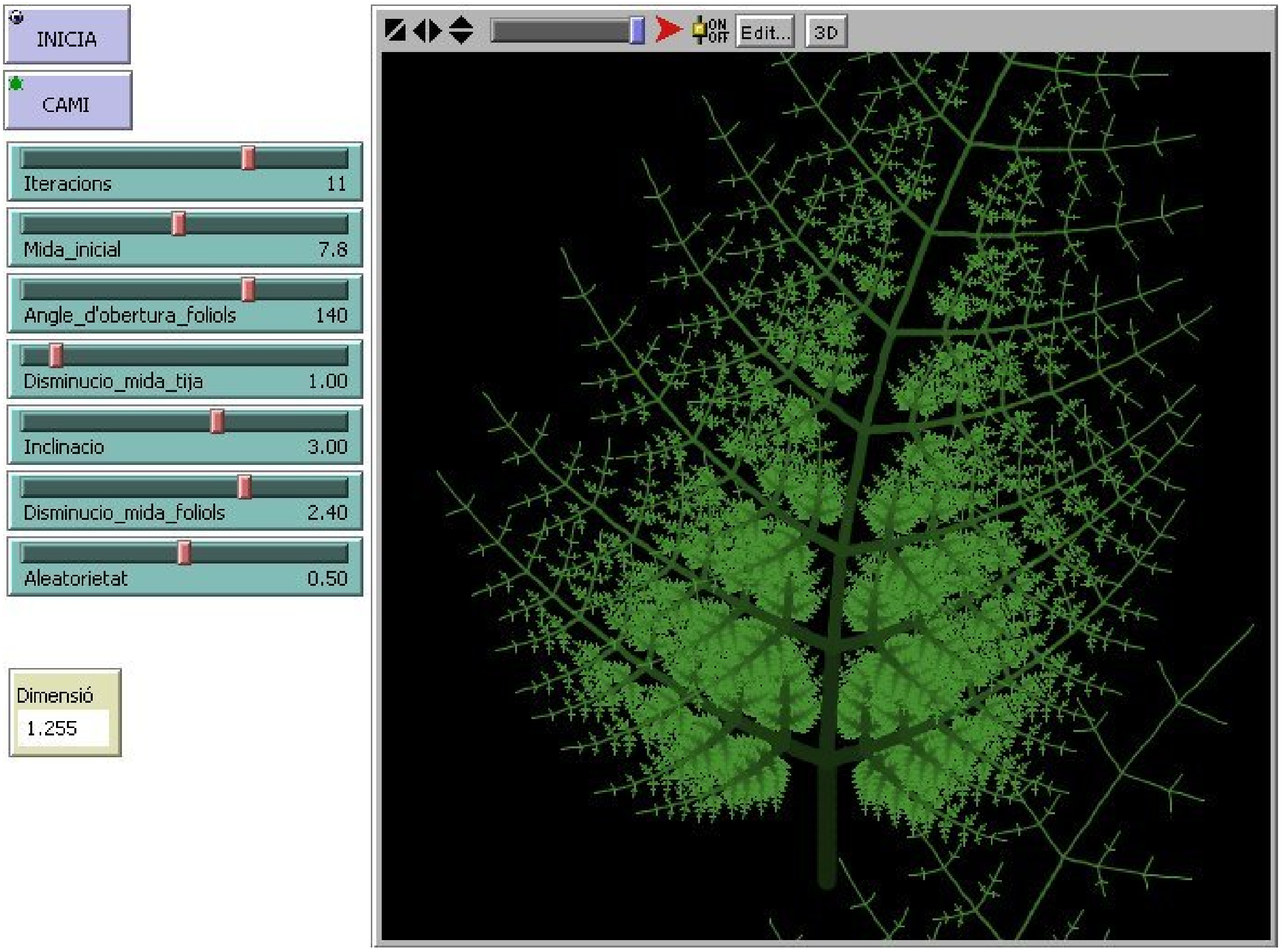
Si l'angle val 55º:

Si l'angle val 170º:

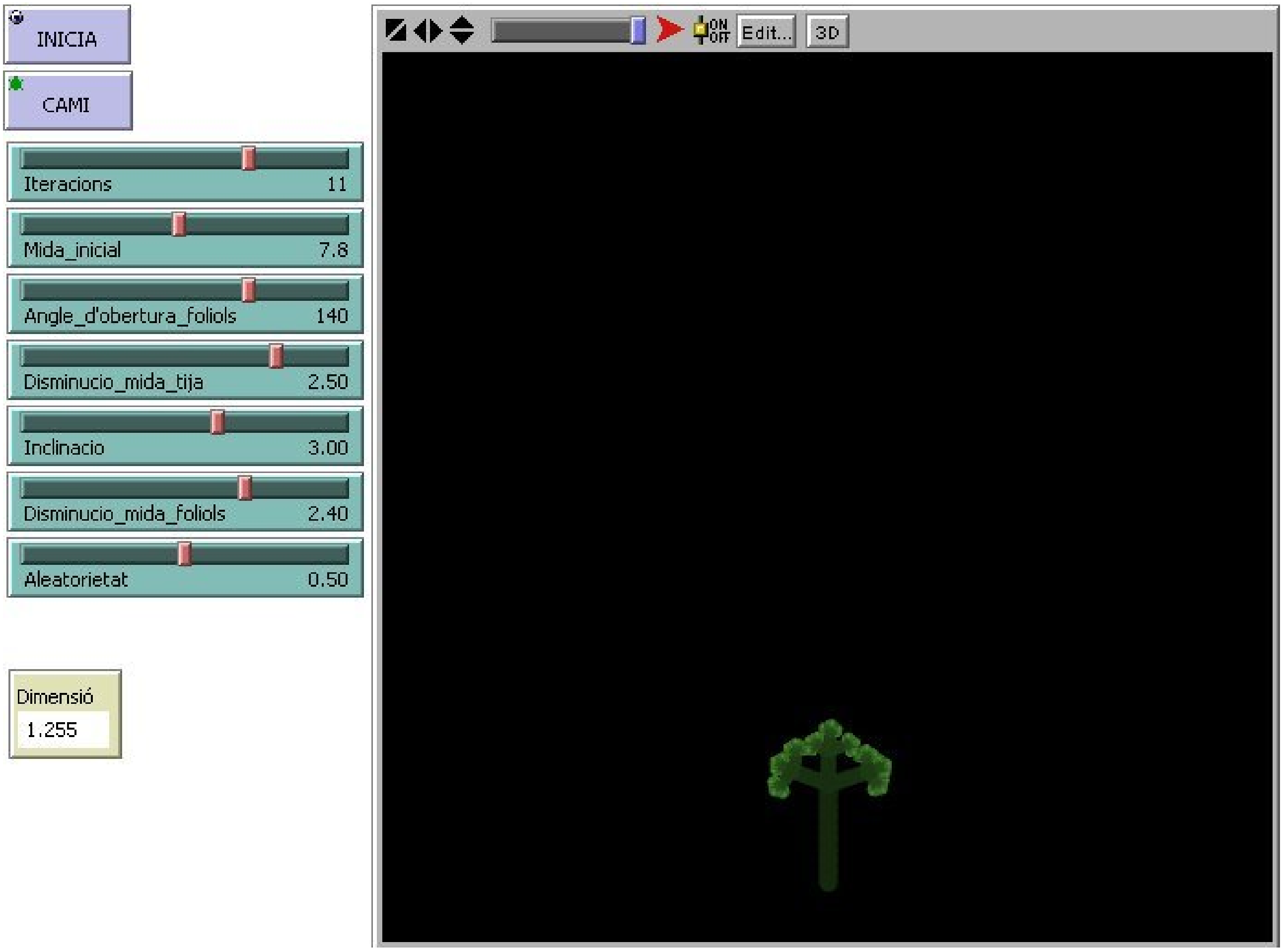


Variem la disminució de la tija:

Amb el valor 1 obtenim:

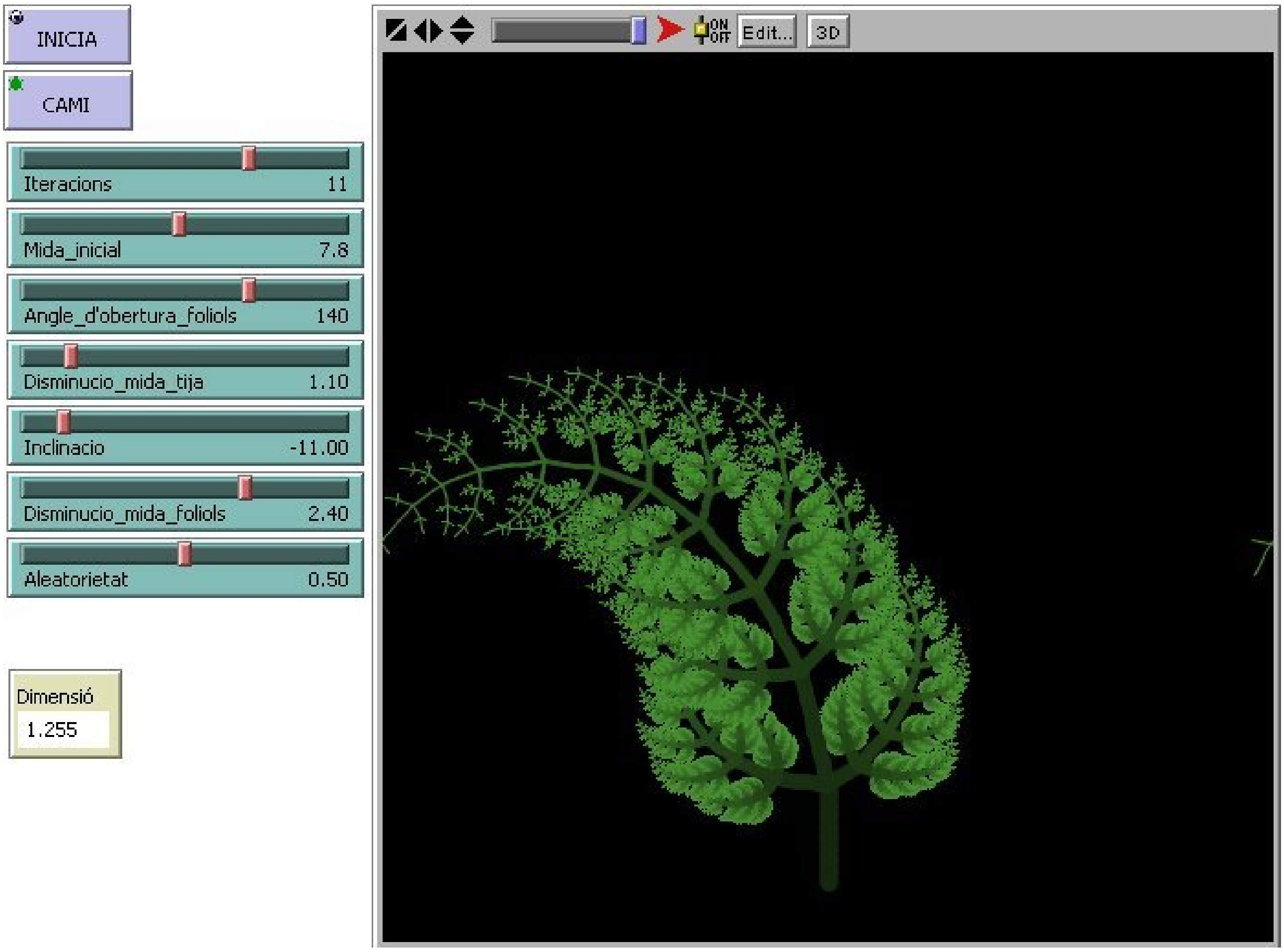


Amb el valor 2,5 obtenim:



Si variem la inclinació:

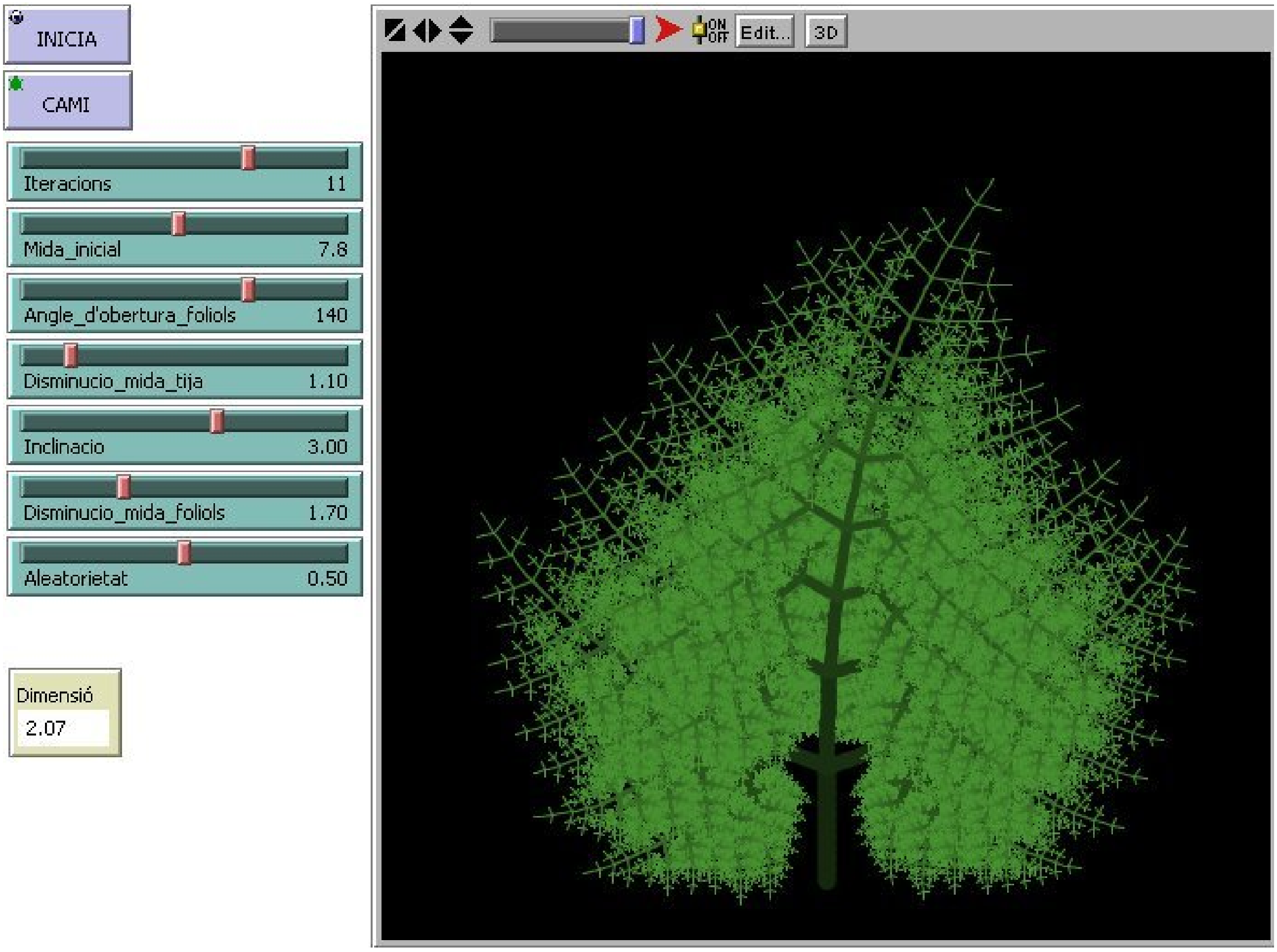
Quant la inclinació val -11:



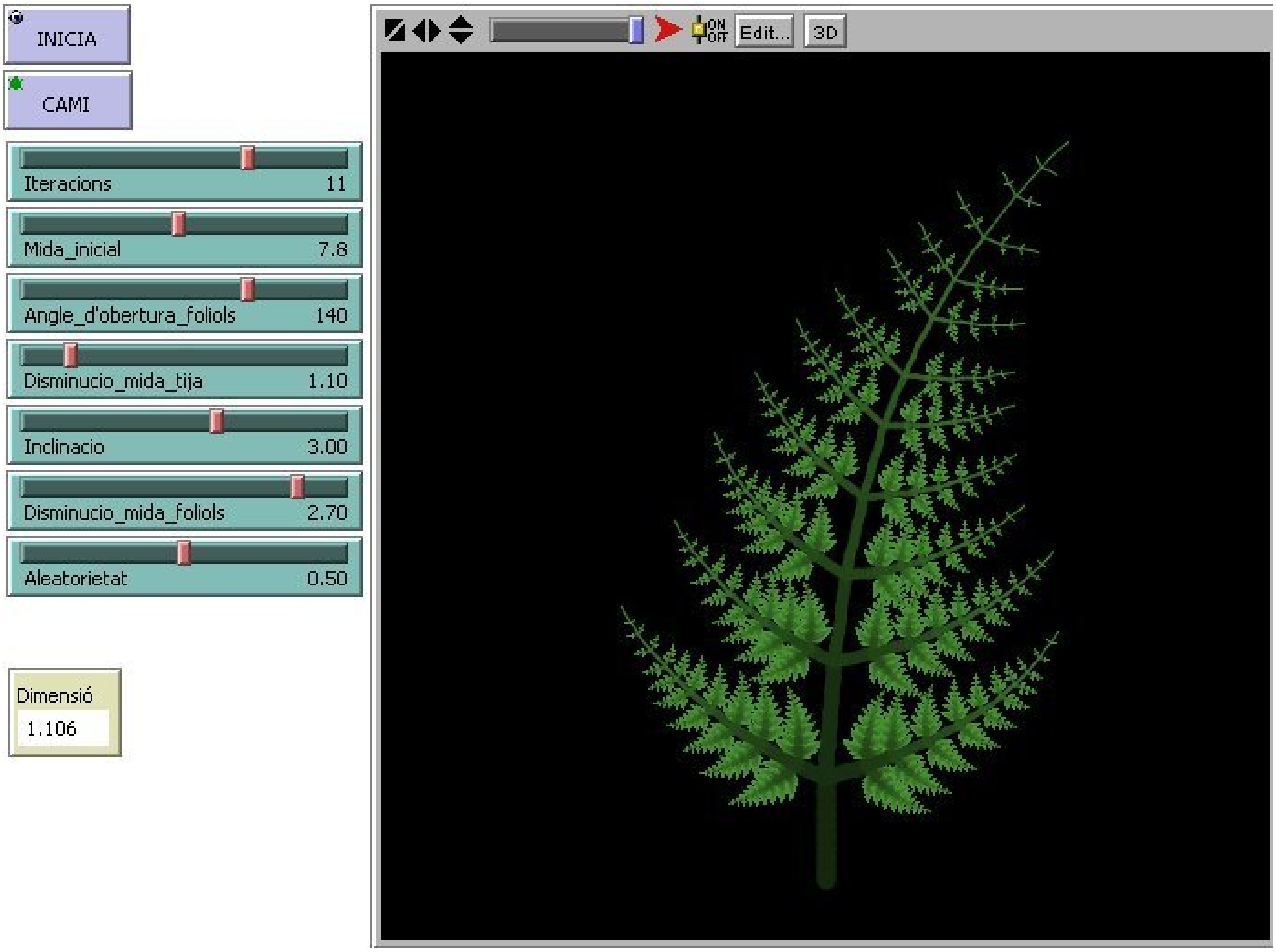
Quant la inclinació val 8,5:

Variem la disminució de la mida dels foliols:

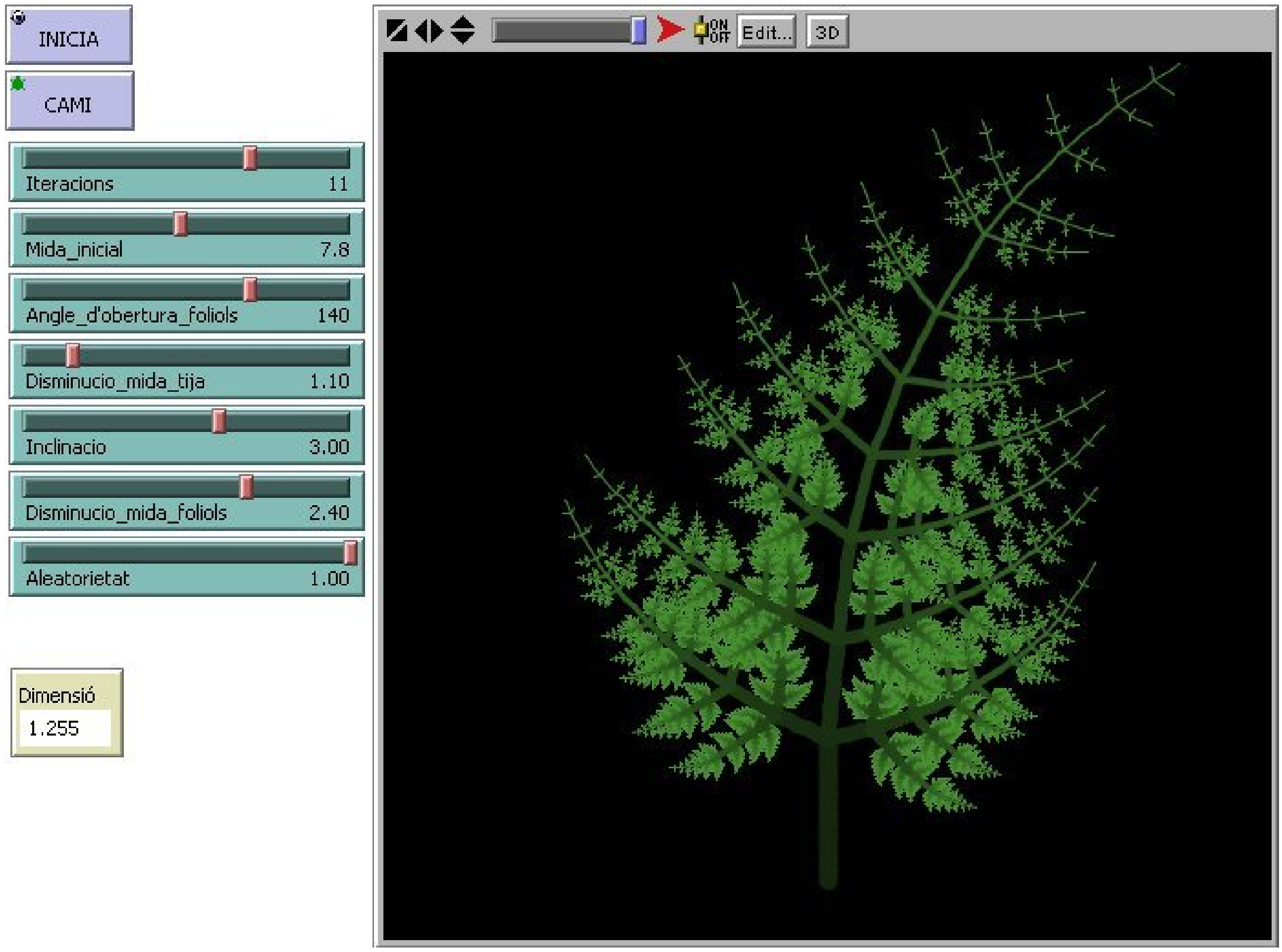
Amb un valor igual a 1,7 obtenim:



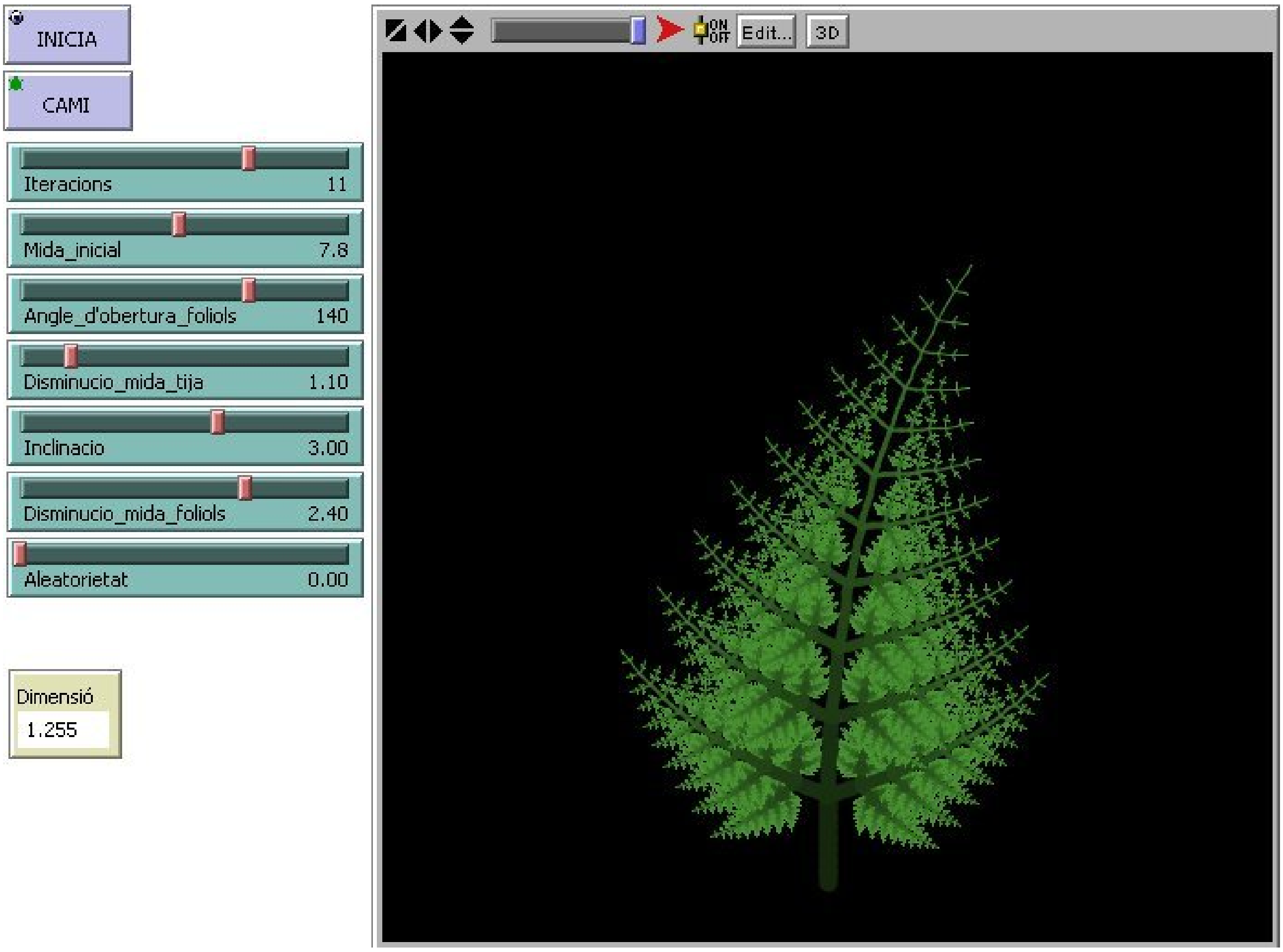
Amb un valor igual a 2,7 obtenim:



Si augmentem al màxim la aleatorietat obtenim:



Si anul·lem l'aleatorietat obtindrem una figura perfectament simètrica respecte l'eix que marca la tija.



La dimensió en aquest programa, depèn de la disminució de la mida dels foliols ja que el nombre de branques a les ramificacions és fixe i les repliques de l'original són els foliols.

$$D=\frac{(\ln 3)}{(\ln(disminució\ mida\ foliols))}$$

N = 3 (2 foliols, una tija principal: tres reproduccions de la imatge grossa)

S = És el nombre per el qual dividim la mida per a obtenir la següent.

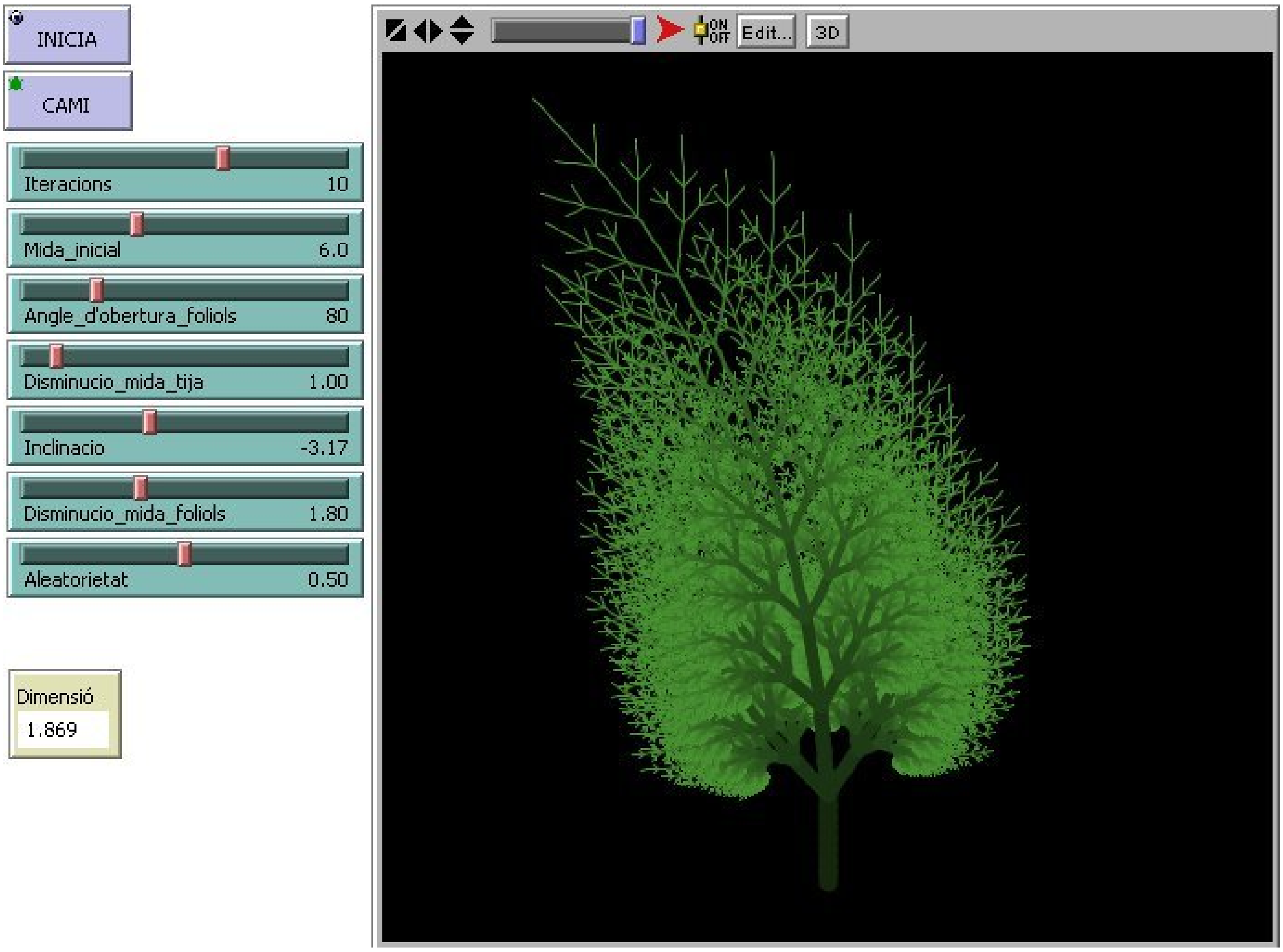
Podem comprovar els diferents valors que ens ha donat la dimensió al canviar la disminució de la mida dels foliols.

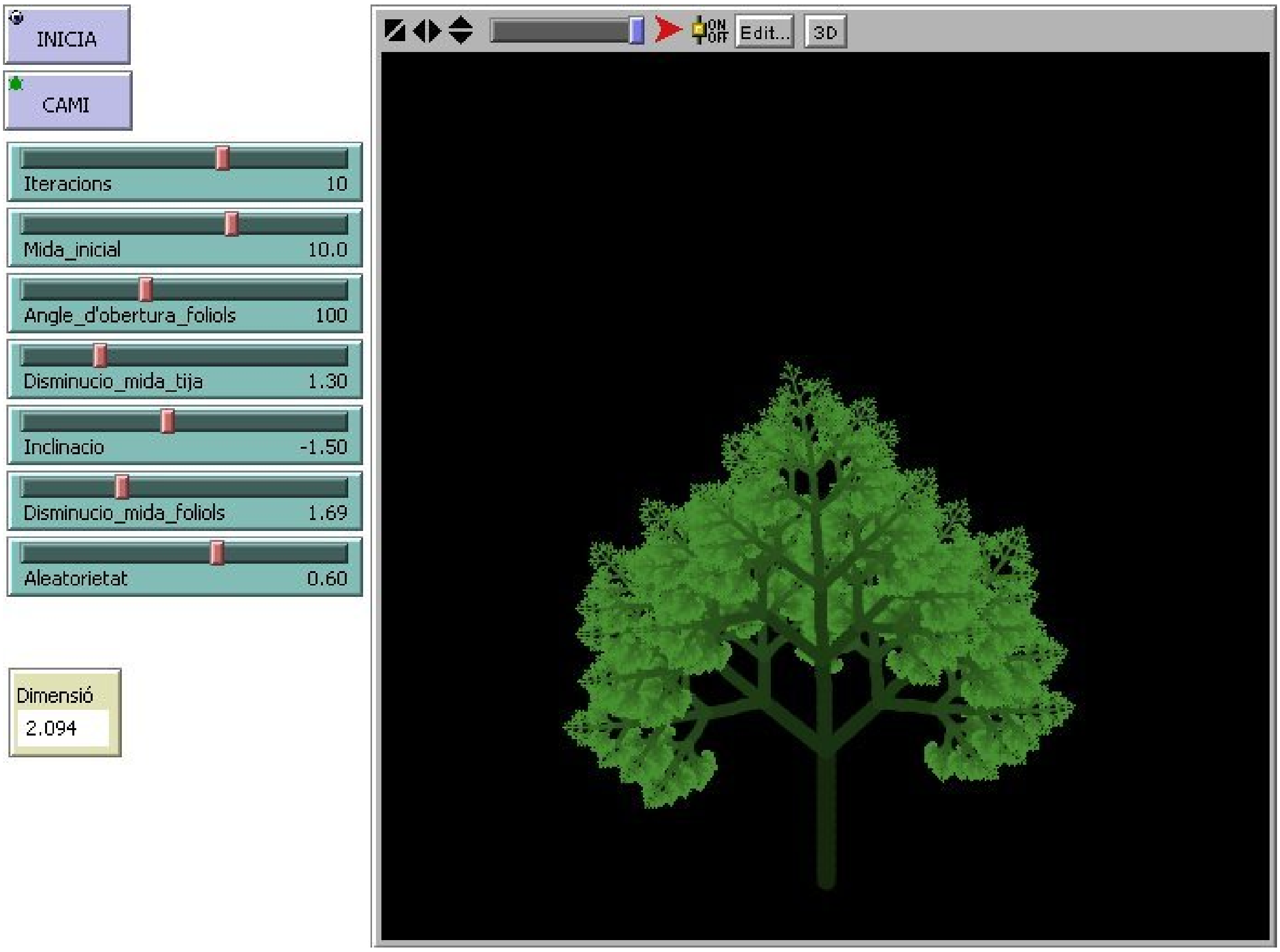
Es tracta de valors superiors a 1 i, la majoria, menors a 2. Normal degut a les característiques de l'estructura de la falguera dibuixada sobre una superfície plana i a base de segments.

Les instruccions corresponents a aquest programa són:

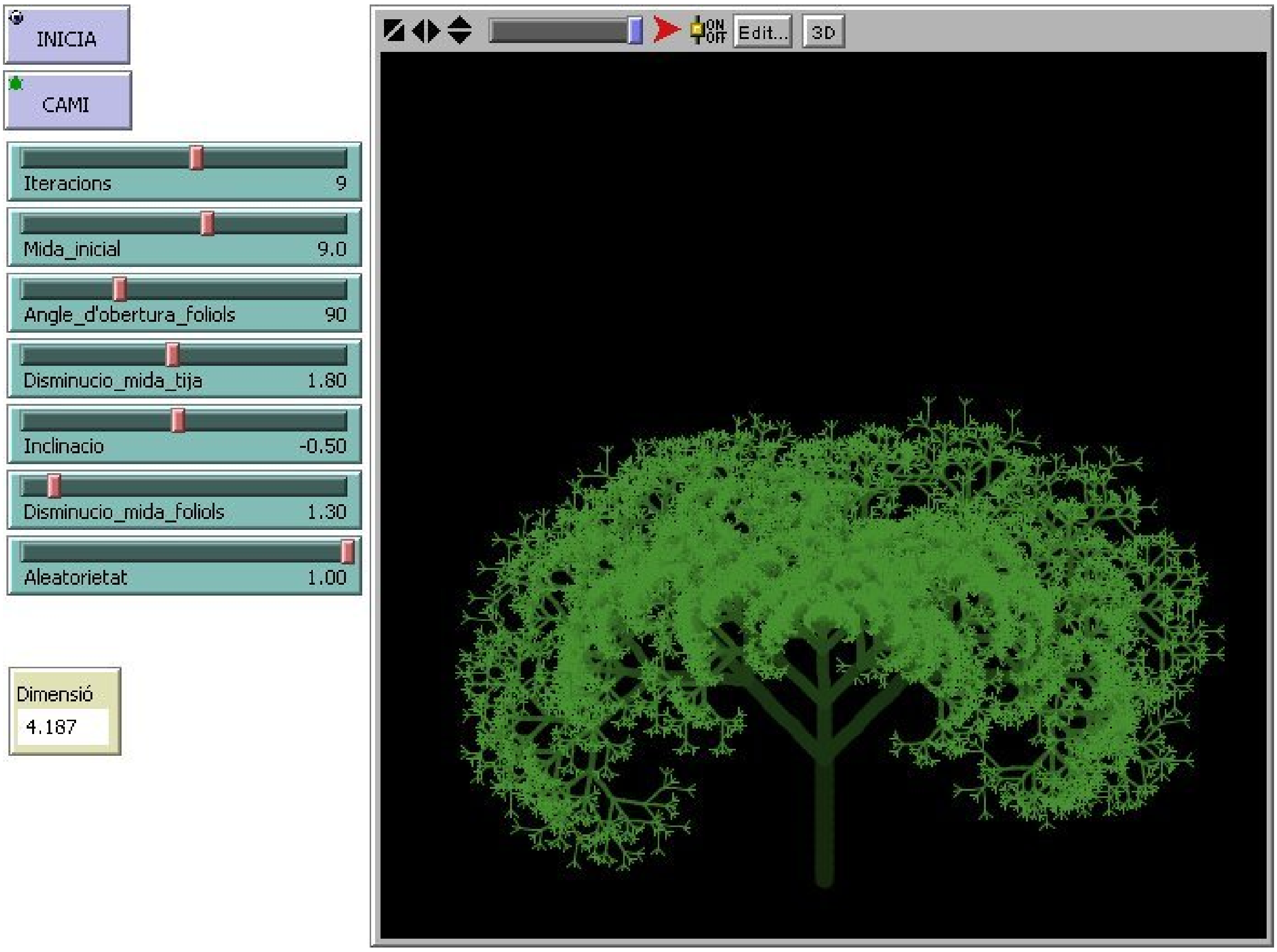
```
to INICIA
    ca
    crt 1
    ask turtles [
        setxy 0 -35
        hide-turtle
        pen-down  ]
end

to CAMI [mida comptador angle]
    if comptador <= Iteracions [
        set pen-size (1 + 9 * (( comptador - Iteracions ) ^ 2)  / Iteracions ^ 2 )
        set color 51 + comptador * 3 / Iteracions
        let variaciomida aleatorietat * random mida
        let variacioangles aleatorietat * ( random ( angle + 6 ) - 3 )
        fd mida + variaciomida
        lt (- angle - variacioangles + angle_d'obertura_foliols / 2)
        CAMI mida / disminucio_mida_foliols comptador + 1  abs angle
        lt (- angle_d'obertura_foliols)
        CAMI mida / disminucio_mida_foliols (comptador + 1) (- abs angle)
        lt angle_d'obertura_foliols / 2
        CAMI mida / disminucio_mida_tija comptador + 1  angle
        lt  angle + variacioangles
        set color 52 + comptador * 4 / Iteracions
        pen-up
        back mida + variaciomida
        pen-down ]
end
```

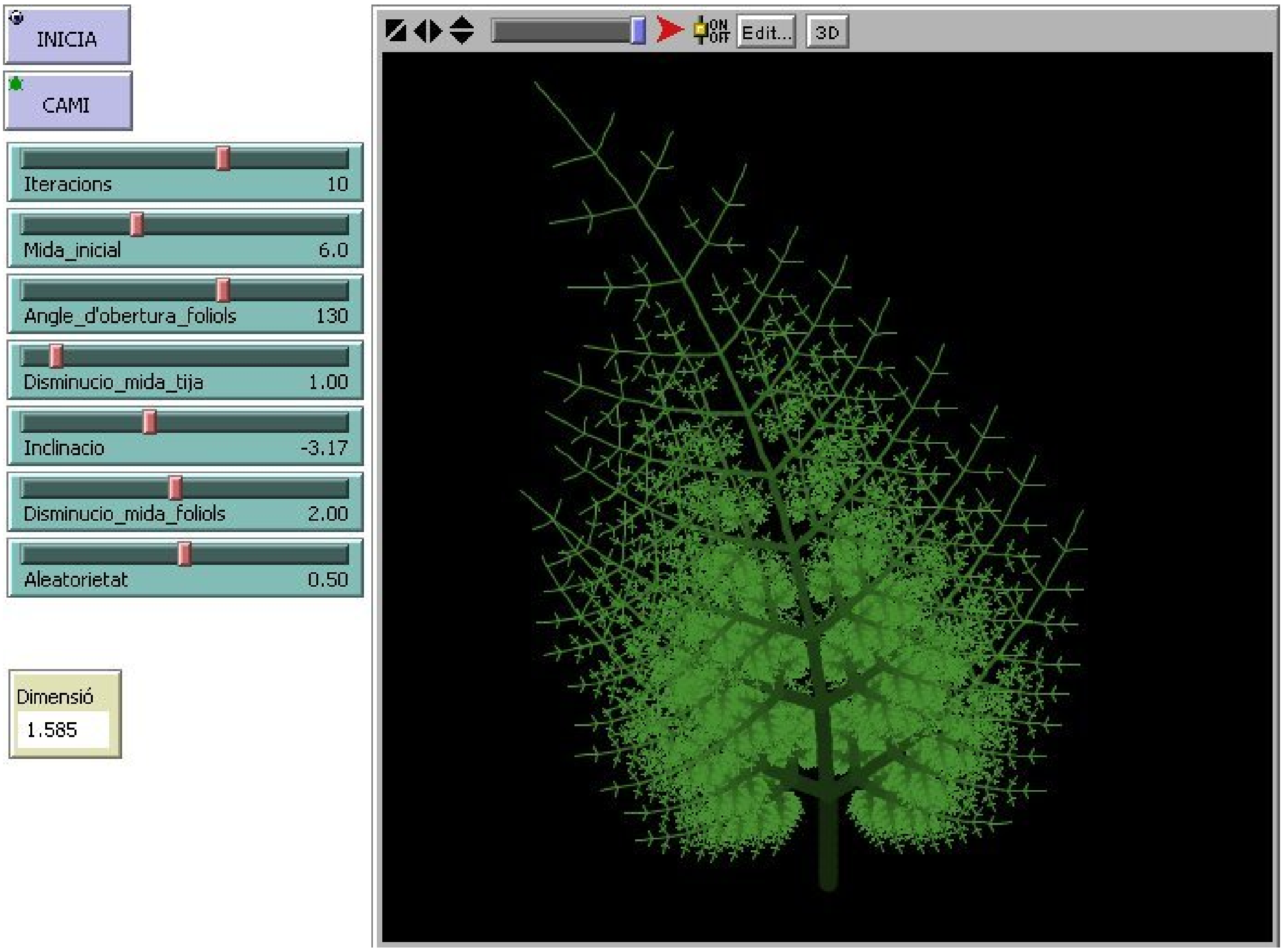
A continuació hi podem veure algunes imatges més creades pel programa.

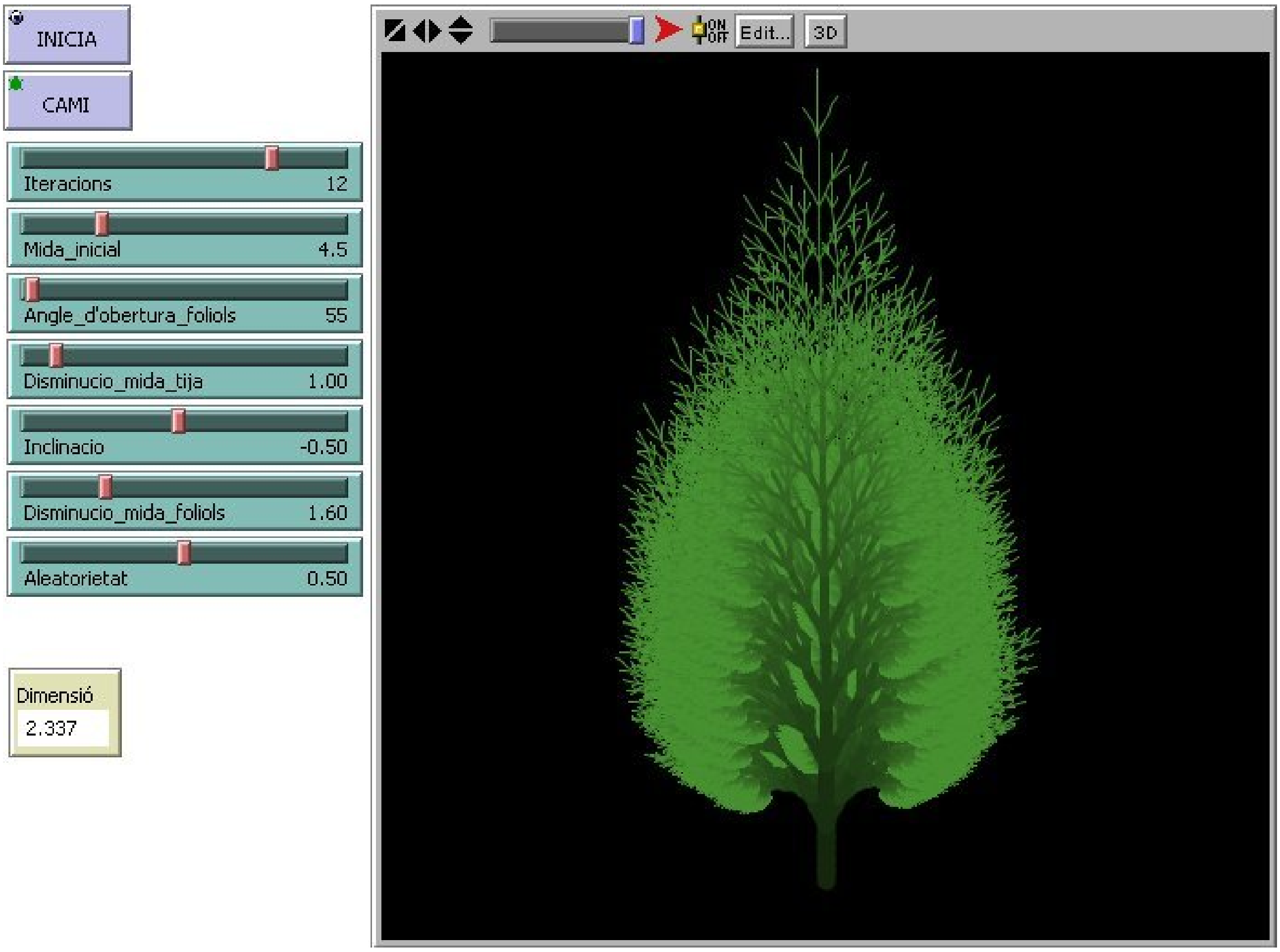

INICIA
CAMI
Iteracions 10
Mida_inicial 6.0
Angle_d'obertura_foliols 80
Disminucio_mida_tija 1.00
Inclinacio -3.17
Disminucio_mida_foliols 1.80
Aleatorietat 0.50
Dimensió
1.869
Edit...
3D


INICIA
CAMI
Iteracions 10
Mida_inicial 10.0
Angle_d'obertura_foliols 100
Disminucio_mida_tija 1.30
Inclinacio -1.50
Disminucio_mida_foliols 1.69
Aleatorietat 0.60
Dimensió
2.094
Edit...
3D


INICIA
CAMI
Iteracions 9
Mida_inicial 9.0
Angle_d'obertura_foliols 105
Disminucio_mida_tija 1.35
Inclinacio -0.50
Disminucio_mida_foliols 1.40
Aleatorietat 1.00
Dimensió
3.265
Edit...
3D


INICIA
CAMI
Iteracions 9
Mida_inicial 9.0
Angle_d'obertura_foliols 90
Disminucio_mida_tija 1.80
Inclinacio -0.50
Disminucio_mida_foliols 1.30
Aleatorietat 1.00
Dimensió
4.187
Edit...
3D


INICIA
CAMI
Iteracions 10
Mida_inicial 6.0
Angle_d'obertura_foliols 130
Disminucio_mida_tija 1.00
Inclinacio -3.17
Disminucio_mida_foliols 2.00
Aleatorietat 0.50
Dimensió
1.585
Edit...
3D


INICIA
CAMI
Iteracions 12
Mida_inicial 4.5
Angle_d'obertura_foliols 55
Disminucio_mida_tija 1.00
Inclinacio -0.50
Disminucio_mida_foliols 1.60
Aleatorietat 0.50
Dimensió
2.337
Edit...
3D

Conclusions

La interpretació de les formes naturals és el que en un principi va donar sentit, raó de ser als fractals, però derivat d'aquesta font d'estudi s'ha creat tot un món força més complicat que continua tenint aplicacions pràctiques.

Els fractals són un tema apassionant que m'ha portat a conèixer lligams entre objectes quotidians i un concepte abans desconegut per a mi. Les matemàtiques, després de tot, no han sigut un pes soportar en aquest treball sinó que, al revés, al tenir una vesant pràctica molt directa, es feien més lleugeres, tot i que difícils d'explicar sobre el paper.

Un cop acabat el treball tinc la sensació de deixar moltes portes inexplorades del món dels fractals per les que no he pogut aprofundir degut a la complexitat de les matemàtiques que comportaven.

He descobert que el programa intèrpret del llenguatge Logo, NetLogo és una eina molt útil per a crear els teus propis programes , inclou moltes capacitats i no es restringeix al camp de les matemàtiques, sinó que també es pot utilitzar per a moltes altres coses com la creació de gràfics, estadístiques o simulacions de ciències socials i experimentals. Permet tenir una llibertat i una flexibilitat que d'altres programes no permeten.

Poca gent ha sentit a parlar dels fractals i encara més poca n'entén el concepte. Jo només he pogut tocar allò que el meu nivell em permetia. He estat treballant i buscant en el tema però tot i així no se si de veritat puc dir que entenc en la seva totalitat l'ampli concepte fractal. Així i tot, m'agradaria que, com a resultat d'aquest treball, més gent s'interessi pels fractals i en conegui els principals trets.

Tinc la sensació, que, després de tot, he arribat a encertar en l'acotació del treball i he arribat a trobar una pràctica que m'ha portat a la creació dels meus propis fractals.

Fer el treball de recerca és una experiència molt positiva ja que ha estat la primera vegada que he fet un treball d'aquesta magnitud. Fer-lo t'ensenya que a més del treball pròpiament dit has de tenir en compte molts aspectes com l'organització, l'estructura, la redacció, el format... , aspectes externs al tema del treball que no arribes a valorar com cal fins que et veus obligada a utilitzar-los.

Bibliografia

A. TOMALIA Donald. *Moléculas dendrímeras* (1995). *Investigación y Ciencia*. Número: 226. Juliol 1995.

BOADA Marc. *Digitaciones viscosas: generación de fractales en un fluido* (2004). *Investigación y Ciencia*. Taller y laboratorio. Septembre 2004.

D. LANDY Stephen. *Cartografía del universo* (1990). *Investigación y Ciencia*. Número: 275. Agost 1990.

DEWDNEY A. K. *Belleza y profundidad: el conjunto de Mandelbrot y una hueste de primos suyos, de apellido Julia* (1988). *Investigación y Ciencia*. Número: 88. Gener 1988.

DEWDNEY A. K. *Montañas fractales, plantas graftales y múltiples ordinográficos en Pixar* (1987). *Investigación y Ciencia*. Número: 125. Febrer 1987.

DEWDNEY A. K. *Paseos aleatorios conducentes hacia muchedumbres fractales* (1989). *Investigación y Ciencia*. Número: 149. Febrer 1989.

DEWDNEY A. K. *Un microscopio computarizado escudriña el objeto más complejo de la matemática* (1985). *Investigación y Ciencia*. Número: 109. Octubre 1985.

JÜRGENS Hartmut; PEITGEN Heinz-Otto; SAUPE Dietmar. *El lenguage de los fractales* (1990). *Investigación y Ciencia*. Número: 169. Octubre 1990.

M. SANDER Leonard. *Crecimiento fractal* (1987). *Investigación y Ciencia*. Número: 126. Març 1987.

MANDELBROT Benoît (1987). *Los objetos fractales*. Tusquets editores. 6ª edició. Barcelona. 2006. ISBN: 84-7223-458-4.

MUSER George. *Comunicaciones inalámbricas, Fractles útiles* (1999). *Investigación y Ciencia*. Número: 276. Septembre 1999.

SHARON Eran; MARDEN Michael; L. SWINNEY Harry. *Flores y hojas onduladas* (2005). *Investigación y Ciencia*. Número: 344. Maig 2005.

Recursos informàtics:

AFONSO Hugo. *Geometría Fractal: Una breve introducción*. [en línia]. Planeta matemático. Desembre 2005. <http://www.planetamatematico.com/index.php?option=com_content&task=view&id=16&Itemid=5>. [Consulta: 5/10/2007]

BAGAN E.; MENDEZ A. Notes d'ampliació de càlcul en variable complexa (1r esborrany). [en línia]. <http://personal.ifae.es/bagan/apunts_ACVC-1.pdf>. [Consulta: 5/10/2007]

BOURKE Paul. *Fractals, Chaos*. [en línia]. University of Western Australia. <http://local.wasp.uwa.edu.au/~pbourke/fractals/>. [Consulta: 5/10/2007]

CONNORS Mary Ann. *Exploring Fractals*. [en línia]. Department of Mathematics and Statistics;University of Massachusetts Amherst. 2007 <http://www.math.umass.edu/~mconnors/fractal/fractal.html>. [Consulta: 5/10/2007]

DENVIR James. *The Mandelbrot/Julia Set Applet*. [en línia]. <http://math.bu.edu/DYSYS/applets/JuliaIteration.html>. [Consulta: 5/10/2007]

E. JOYCE David. *Complex Numbers*. [en línia]. 1999. Department of Mathematics and computer Sciencs. Clark University. <http://www.clarku.edu/~djoyce/complex/>. [Consulta: 5/10/2007]

E. JOYCE David. *Julia and Mandelbrot Sets*. [en línia]. Department of Mathematics and Computer Science; Clark University. 2003. <http://aleph0.clarku.edu/~djoyce/julia/>. [Consulta: 5/10/2007]
ELERT Glenn. *The chaos hypertestbook*. [en línia]. <http://hypertextbook.com/chaos/>.

[Consulta: 5/10/2007]

FAGELLA Núria. El conjunt de Mandelbrot i altres plans de bifurcació. [en línia]. Butlletí de la Societat Catalana de Matemàtiques. Març 1999. <http://www.raco.cat/index.php/ButlletiSCM/article/viewFile/17616/17457>. [Consulta: 5/10/2007]

FIOL MORA Miguel Angel . *Els Nombres Complexos*. [en línia]. Departament de Matemàtica Aplicada IV. Universitat Politècnica de Catalunya. <http://www-ma4.upc.edu/~fiol/matel/complexos.pdf>. [Consulta: 5/10/2007]

FRAME Michael; MANDELBROT Benoit;NEGER Nial. *Fractal Geometry*. [en línia]. Yale University. Octubre 2007. <http://classes.yale.edu/fractals/>. [Consulta: 5/10/2007]

GHEE BEOM KIM. *Fractals*. [en línia]. 2006. Copyright. <http://geometricarts.googlepages.com/fractals>. [Consulta: 5/10/2007]

KANAMARU Takashi; T. THOMPSON J. Michael. *Julia set applet* . [en línia]. Setembre 1997. <http://brain.cc.kogakuin.ac.jp/~kanamaru/Chaos/e/Julia/>. [Consulta: 5/10/2007]

LAMBERTSON Lori. Fractales en su aula. [en línia]. Exploratorium Teacher Institute San Francisco, EUA. Fundación CIENTEC. 2001. <http://www.cientec.or.cr/matematica/fractales.html> [Consulta: 5/10/2007]

LANIUS Cynthia. *Fractals*. [en línia]. 2007. <http://math.rice.edu/~lanius/frac/>. [Consulta: 5/10/2007]

MANDELBROT Benoît. *How long is the coast of britain?*. [en línia]. Science. Número: 156. Març 1967. <http://www.math.yale.edu/mandelbrot/web_pdfs/howLongIsTheCoastOfBritain.pdf>. [Consulta: 5/10/2007]
OLIVELLA Roger. *Insula smaragdina / Visual poetry*. [en línia]. Desembre 2004.

<http://www.rogerolivella.net/insula/cat/descripcio.htm#quesonfract>. [Consulta: 5/10/2007]

OSINGA Hinke. *One-Dimensional Dynamical Systems*. [en línia]. May 1998. <http://www.geom.uiuc.edu/~math5337/ds/>. [Consulta: 5/10/2007]

PACREU Mireia . *Té dimensió fractal la Costa Brava?* [en línia]. Gener 2002. <http://www.xtec.es/ieslabisbal/fractals/intro.htm>. [Consulta: 5/10/2007]

SYSIFUS. [en línia]. Març 2000. <http://www.arrakis.es/%7Esysifus/intro.html>. [Consulta: 5/10/2007]

TUCEK Jim. *Fractals*. [en línia]. Agost 2004. <http://www.jracademy.com/~jtucek/math/fractals.html>. [Consulta: 5/10/2007]

WEISSTEIN Eric. *Fractal*. [en línia]. Wolfram Research. Octubre 2007. <http://mathworld.wolfram.com/Fractal.html>. [Consulta: 5/10/2007]

WIKIMEDIA. *Fractal*. [en línia]. Setembre 2007. <http://commons.wikimedia.org/wiki/Fractals>. [Consulta: 5/10/2007]

WIKIPEDIA. *Fractal*. [en línia]. Octubre 2007. <http://en.wikipedia.org/wiki/Fractal>. [Consulta: 5/10/2007]

YANGO. *Fractales: una nueva geometría*. [en línia]. Març 2005. <http://usuarios.lycos.es/sisar/fractales/indice.php>. [Consulta: 5/10/2007]