

Sistemes Iteratius de Funcions: un programari

L'objectiu d'aquest taller és introduir un programari (*software*) per generar fractals mitjançant **Sistemes Iteratius de Funcions** (IFS), complementant la conferència realitzada per la professora Núria Fagella. Aquests fractals van ser concebuts per John Hutchinson l'any 1981 i van ser popularitzats per Michael Barnsley amb el llibre *Fractals Everywhere*, encara que alguns exemples són molt anteriors, com el conjunt de Cantor (1881) o el triangle de Sierpinski (1915).

Emprarem els aplicatius (*applets*) que trobem a la web d'un curs sobre fractals de la Universitat de Yale, <http://classes.yale.edu/fractals/>, on a més hi ha molta informació sobre aquests objectes misteriosos, amb multitud d'aplicacions. Un dels professors és Benoit Mandelbrot, introductor dels fractals amb llibres com *La Geometria Fractal de la Naturaleza* (Metatemas 49, Tusquets Editores). El software el trobem a <http://classes.yale.edu/Fractals/Software/Software.html>.

Al final d'aquestes notes repassem breument alguns conceptes matemàtics, recordant les definicions i notacions sobre **afinitats** del pla que s'utilitzen en aquest programari.

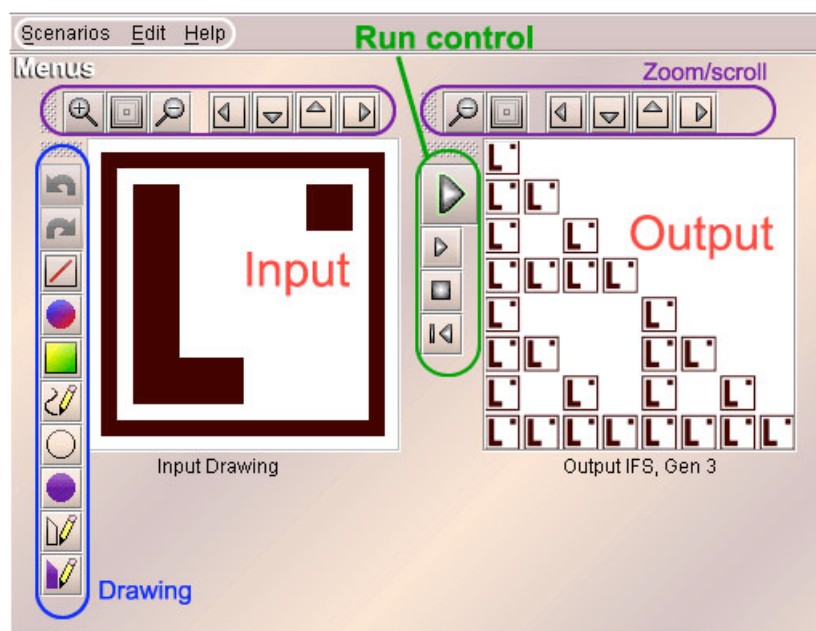


Àlex Haro
Departament de Matemàtica Aplicada i
Anàlisi

Atractors Fractals de Sistemes Iteratius de Funcions

El software **Deterministic IFS** que es troba a <http://classes.yale.edu/Fractals/Software/detifs.html> permet obtenir l'atractor d'un IFS donat mitjançant afinitats contractives. L'aplicatiu ha estat programat per Michael Frame i Ginger Booth.


El panel de control de l'aplicatiu és com el de la següent figura, on apareixen diverses barres de botons (que aquí estan engrandides), amb un **àrea de dibuix** (esquerra) i un **àrea de generació** (dreta).



A l'àrea de dibuix s'ha d'introduir l'**iniciador**, a partir del qual es genera el fractal per aplicació successiva de les **regles** generadores. A l'àrea de generació apareixen les generacions del dibuix original.

La barra vertical central és per generar el fractal, i els botons són, respectivament, per *córrer/fer un pas/parar/tornar al principi*. Així, en prémer el triangle gran



es succeeixen diverses generacions (per defecte, 4), però podem fer més prement el triangle petit . També podem fer zooms i desplaçaments usant els botons de les barres horitzontals. Els de la barra vertical de l'àrea de dibuix són precisament per manipular-lo, i són respectivament, per *desfer/refer/esborrar el dibuix, seleccionar color/fons i pintar a mà/cercles/cercles plens/polígons/polígons plens*. Per defecte, l'iniciador és

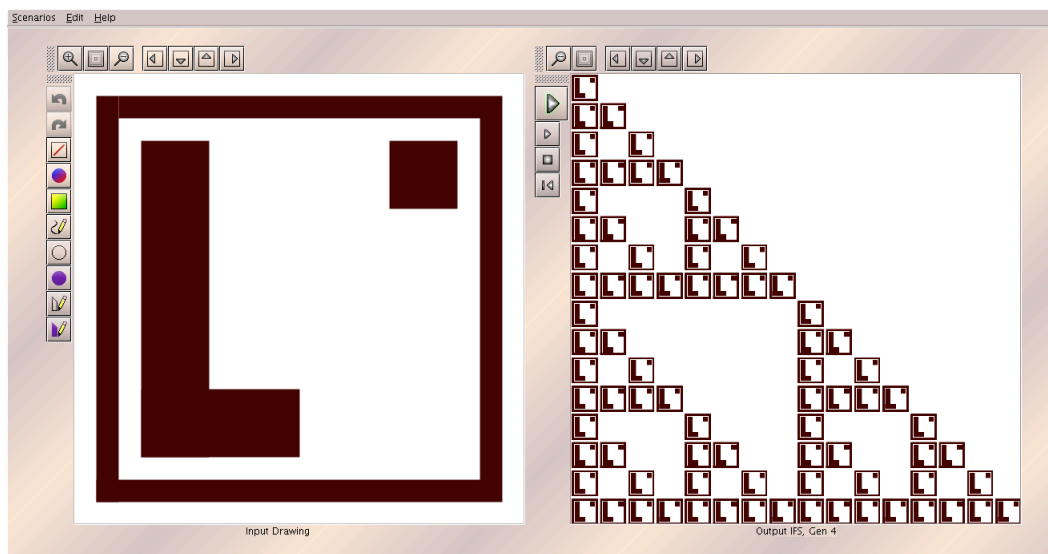



Però podem dibuixar-ne de nous!

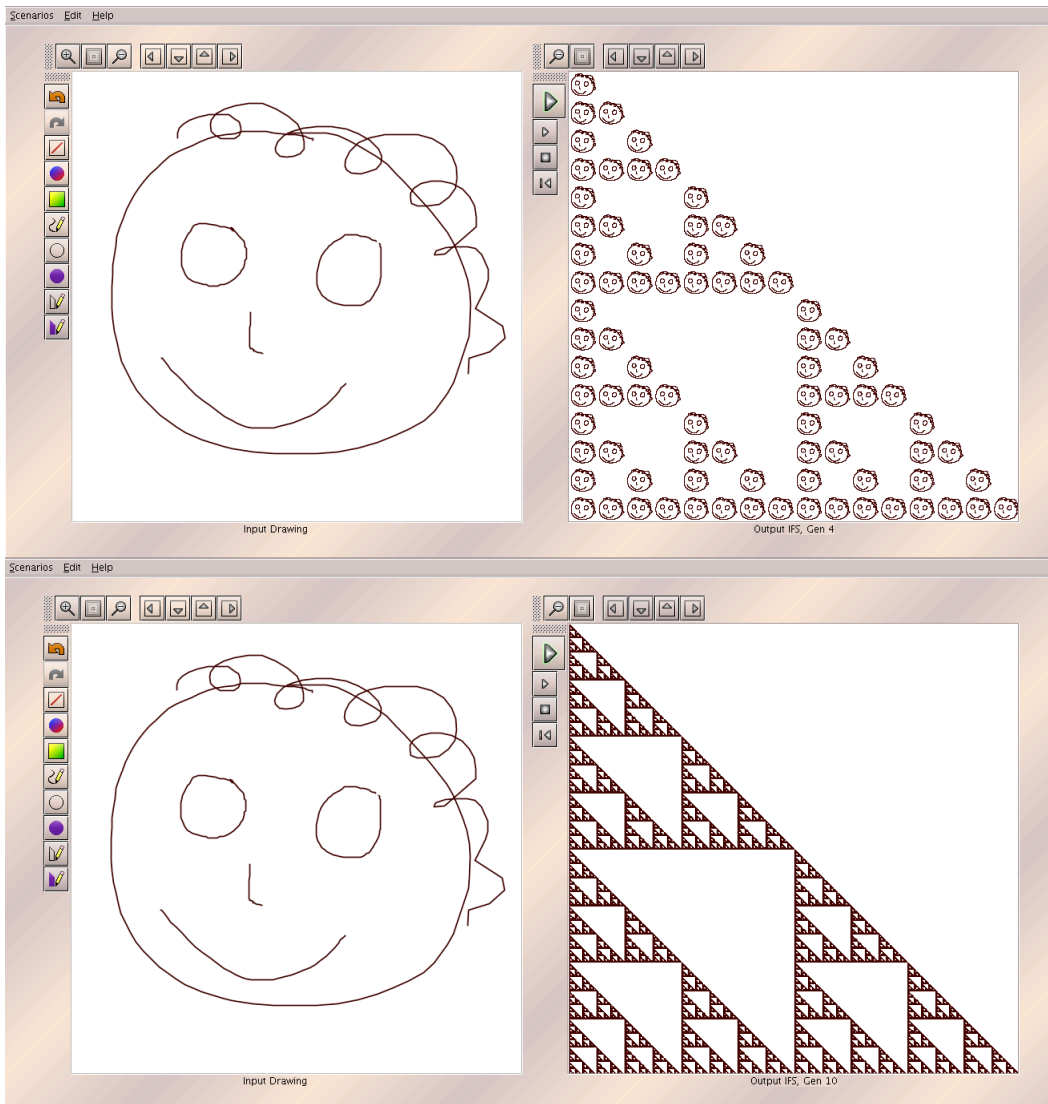
Les regles generadores del fractal es poden veure, modificar i crear en el menú **Edit**. Hi ha uns quants de preestablerts a **Scenarios**. Clicant la tecla de **Help** trobarem més informació sobre el programa.

Exemple 1: El triangle de Sierpinski

Quan es clica al link **Deterministic IFS** s'activa l'aplicatiu **detifs.html** amb l'exemple del triangle de Sierpinski.



Prement , esborrarem el dibuix inicial, i podem pintar el nostre propi iniciador amb el mouse. En la figura següent apareix a l'esquerra un iniciador dibuixat a mà, i a la dreta apareix la quarta generació. Veiem després que, si continuem iterant el procediment, obtenim el triangle de Sierpinski.



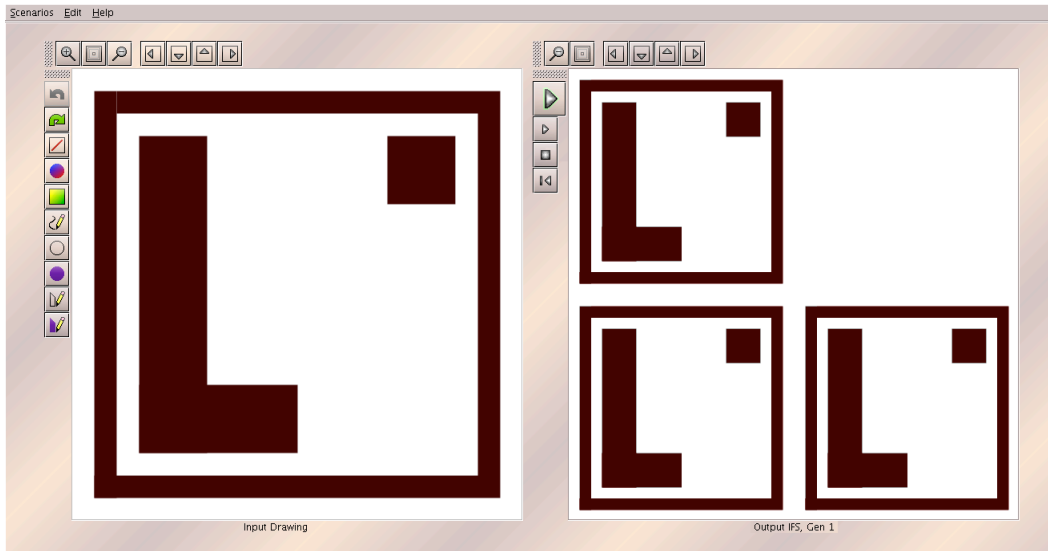
Emfatitzem que, afortunadament (almenys per qui escriu aquestes línies), el dibuix final és independent de l'iniciador. El procés genera una successió de dibuixos que convergeix cap a l'**atractor** del IFS.

Q-1: Dibuixa el teu propi iniciador. Comprova que convergeix cap al mateix triangle de Sierpinski!

Podem consultar la llista d'afinitats del generador del fractal al menú **Edit**. En aquest cas vénen donades per la taula següent, en la que es considera que el triangle està enmarcat en el quadrat unitat $[0,1] \times [0,1]$.

afinitat	r	s	θ	φ	e	f
1	0.5	0.5	0	0	0	0
2	0.5	0.5	0	0	0.5	0
3	0.5	0.5	0	0	0	0.5

Podem veure l'efecte de cadascuna de les afinitats fent només una generació. Inclús deixant només una afinitat podem veure quin efecte fa. Això és útil a l'hora de generar nous fractals, o per buscar les regles d'un fractal donat. En aquest cas és molt senzill, i veiem que la primera generació del dibuix original és



Q-2: A quina afinitat correspon cadascuna de les transformacions?

Q-3: Afegeix a la llista d'afinitats la que correspon al “subquadrat” que falta (el superior a la dreta). Quin és l'atractor del IFS? Sembla que no és gaire fractal...

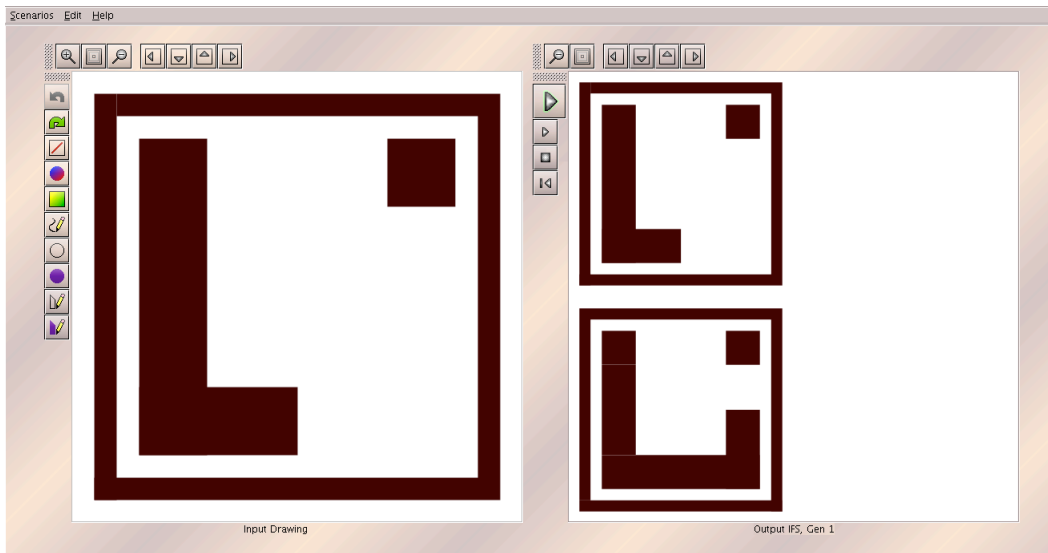
Q-4: Modificar els factors de contracció agafant per exemple tots iguals a 0.45, 0.4, 0.35, 0.3 ... Què observes? Pots continuar experimentant escollint altres factors, que inclús poden ser diferents entre ells!

Exemple 2: Un fill del Triangle de Sierpinski

Si volem, per exemple, que el quadrat de la dreta apareixi rotat 90 graus en el sentit antihorari, haurem de modificar l'afinitat 2 de la taula. Provem!

afinitat	r	s	θ	φ	e	f
1	0.5	0.5	0	0	0	0
2	0.5	0.5	90	90	0.5	0
3	0.5	0.5	0	0	0	0.5

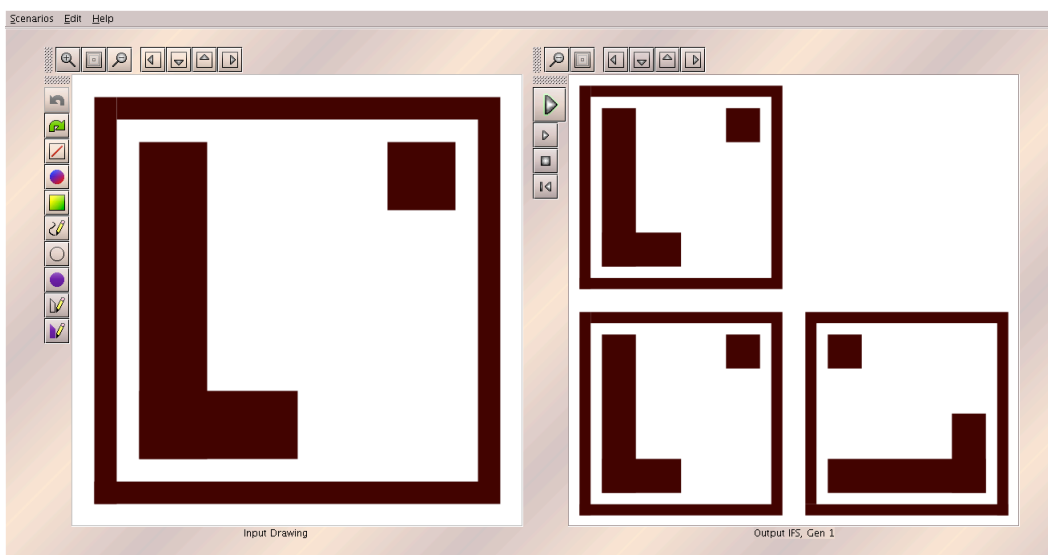
En fer la primera generació d'aquest nou fractal obtenim ...



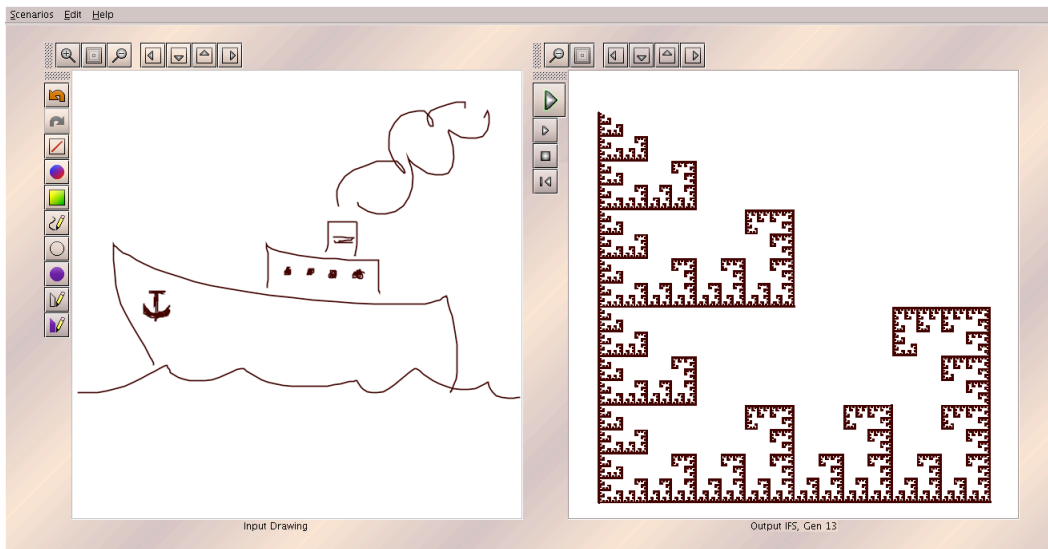
Sembla ser que no ho hem fet massa bé ... El problema és que per construir l'afinitat 2 hem de fer: l'escalament de factors $r = s = 0.5$, la rotació rígida de 90 graus ... i la translació adequada! També hem de canviar e i f , doncs!

Pel que veiem, hem d'afegir una translació de vector $(0.5,0)$, que afegida a l'anterior dóna $(1,0)$. Tornem-hi, que no ha estat res!

afinitat	r	s	θ	φ	e	f
1	0.5	0.5	0	0	0	0
2	0.5	0.5	90	90	1	0
3	0.5	0.5	0	0	0	0.5

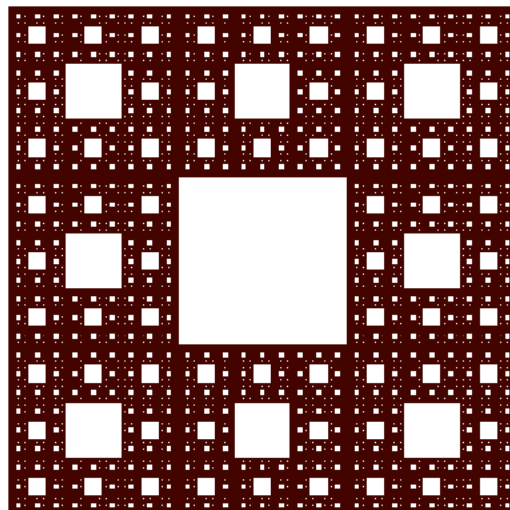


Ara sí que ho hem fet bé! Si continuem iterant, encara que sigui amb un altre iniciador, obtenim l'atractor d'aquest nou IFS.



Exemple 3: La catifa de Sierpinski

En Sierpinski també va dissenyar una catifa fractal.



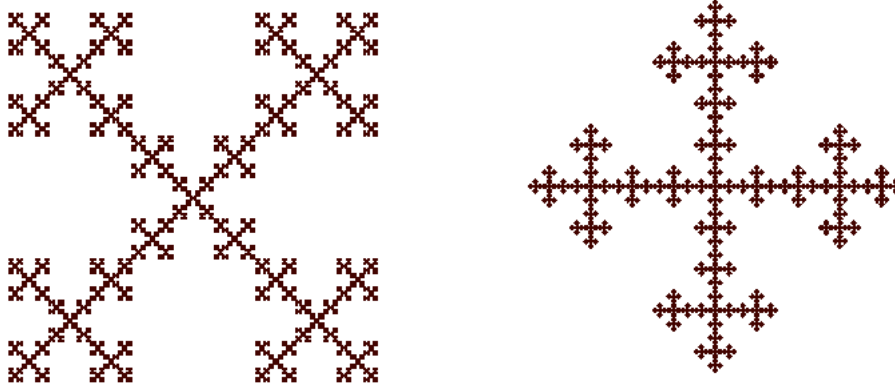
La podem generar amb el programa, escollint l'opció **Sierpinski carpet** al menú **Scenarios**. Aquest fractal està generat per 8 contraccions amb $r = s = 1/3$.

Q-6: Si suposem que està enmarcat al quadrat unitat, sabeu trobar la llista d'afinitats? Podem comprovar la llista d'afinitats a **Affine Transforms** del menú **Edit**.

Q-7: Experimenta de nou canviant els factors de contracció, tal com vas fer amb el Triangle de Sierpinski.

Exemple 4: els fractals 'x' i '+'

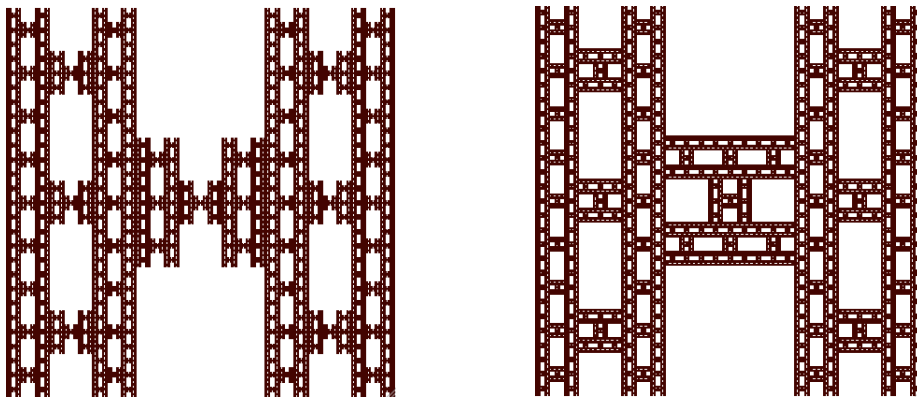
Modificant adequadament les dades corresponents a la catifa de Sierpinski, podem aconseguir les següents creus fractals.



Q-8: Quàntes contraccions necessitem en aquests dos fractals? Modifica les dades de la catifa de Sierpinski per generar-los.

Exemple 5: els fractals 'H'

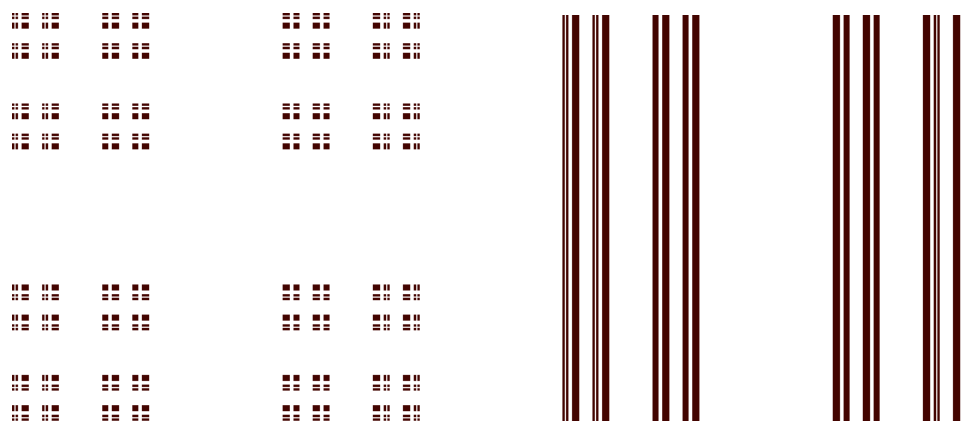
També podem generar fàcilment els següents fractals 'H', modificant la catifa de Sierpinski:



Q-9: Genera els dos fractals. Observa que al de la dreta, que sembla més maco, s'ha de fer alguna rotació.

Exemple 6: La pols de Cantor

En la figura següent, el fractal de l'esquerra rep el nom de pols de Cantor, i el “codi de barres” és un ternari de Cantor “multiplicat” per un interval.



Q-10: Els dos fractals es poden generar amb 4 afinitats. Fes-ho!

Exemple 7: La falguera

Entre altres exemples que pots trobar al menú **Scenaris** de l'aplicatiu, podem trobar la famosa **falguera**, sota el nom de **Barnsley's Fern**. És un exemple de **fractal naturalista**.



La llista d'afinitats del IFS de la falguera és

afinitat	r	s	θ	φ	e	f
1	0	0.16	0	0	0	0
2	0.85	0.85	-2.5	-2.5	0	1.6
3	0.3	0.34	49	49	0	1.6
4	0.3	0.37	-50	-50	0	0.44

És un fet realment notable que amb aquests 24 nombres es pugui generar un objecte (aparentment) tan complex.

En intentar generar la falguera amb l'aplicatiu, hi ha un petit problema. Sembla que al programa “no li agrada” que hi hagi factors de contracció que siguin 0, i el factor r de la primera afinitat ho és! Podem resoldre aquest entrebanc agafant $r=0.00001$ (per exemple). Aquest factor és important per generar el tronc i les branques de la falguera, que són “molt primets”.

Q-11: Modificant una mica els coeficients de la falguera pots moure-la una miqueta, com si la estigués movent el vent. Per començar, pots jugar amb els coeficients rotacionals de l'afinitat 2. Un projecte interessant seria fer una petita pel·lícula.

Q-12: Com engreixaries el tronc de la falguera? Prova-ho!

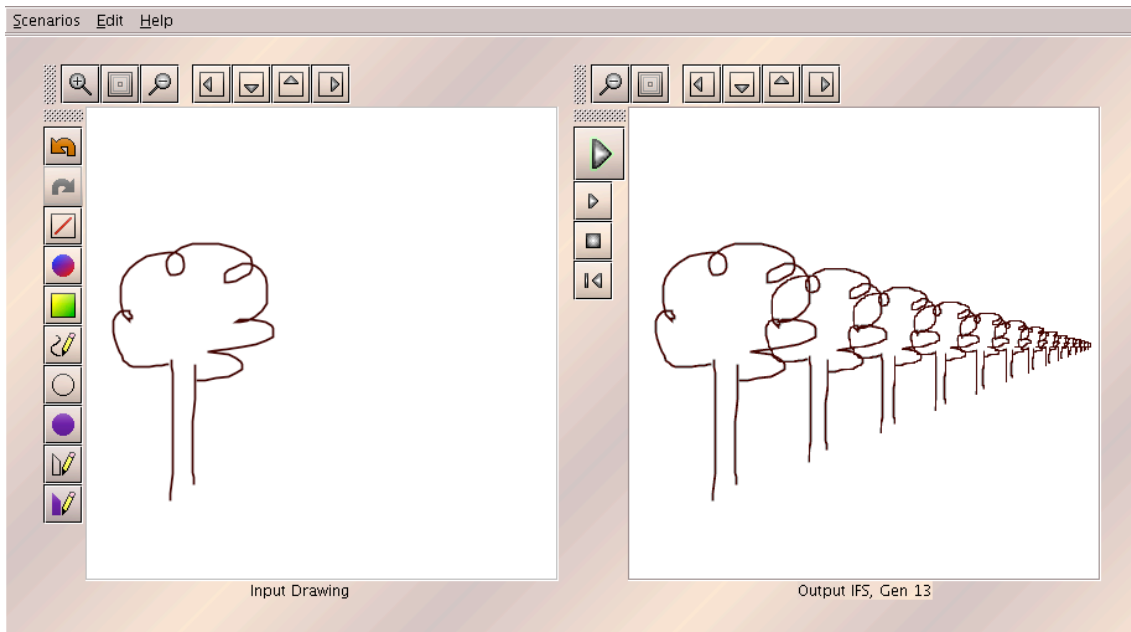
IFS amb condensació

Podem aconseguir que el l'iniciador es repeteixi infinites vegades en l'atractor del IFS, tot escollint la primera afinitat que sigui l'aplicació indentitat (és a dir, $r = s = 1$, $\theta = \varphi = 0$, $e = f = 0$). Encara que aquesta aplicació no sigui contractiva, si la resta ho són, també convergirem cap a un atractor. En aquest cas, es diu que el IFS és **amb condensació**, i el **conjunt de condensació** és el motiu inicial que es repeteix.

Per exemple, amb la següent llista d'afinitats

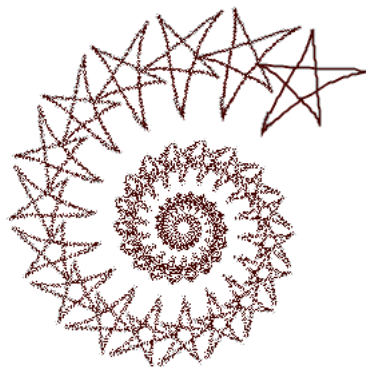
afinitat	r	s	θ	φ	e	f
1	1	1	0	0	0	0
2	0.75	0.75	0	0	0.25	0

i amb l'iniciador que surt a l'esquerra de la figura de sota, convergim cap a l'atractor de la dreta.



Q-13: Prova altres iniciadors (o conjunts de condensació), a veure cap a on convergeixen! Observa que en aquest cas l'atractor **sí que depen** de l'iniciador. Per què?

Q-14: Dissenya un IFS amb condensació en què l'atractor sigui (més o menys), el de la figura següent:



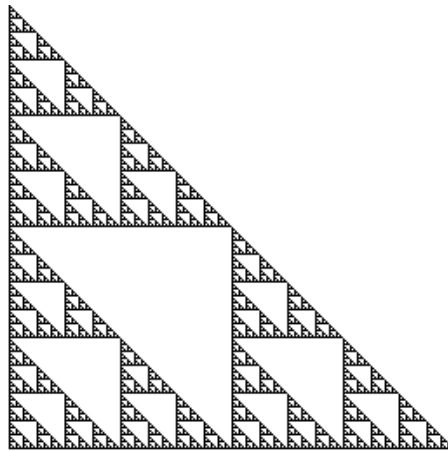
Q-15: Amb el truc d'usar un conjunt de condensació adequat, genera el conjunt "codi de barres" de l'exemple 6, usant només 3 afinitats (incloent la indentitat). Sent una mica més astut, fins i tot es pot fer amb dues afinitats!

Q-16: Pots generar la pols de Cantor de manera similar?

La família Sierpinski

Observant el triangle de Sierpinski, veiem que verifica les següents propietats **topològiques**¹:

- és **connex**, ja que no el podem separar en diferents components;
- és **arc-connex**, ja que per qualsevol parell de punts del fractal hi ha un camí (dins el fractal) que va d'un punt a l'altre;
- conté una infinitat de **loops** (o camins tancats);
- és **múltiplement connex**, perquè és connex i hi ha loops encerclant forats del conjunt;
- no té cap **branch point** (o punt de ramificació), que és un punt que si el traiem, desconnecta el conjunt.

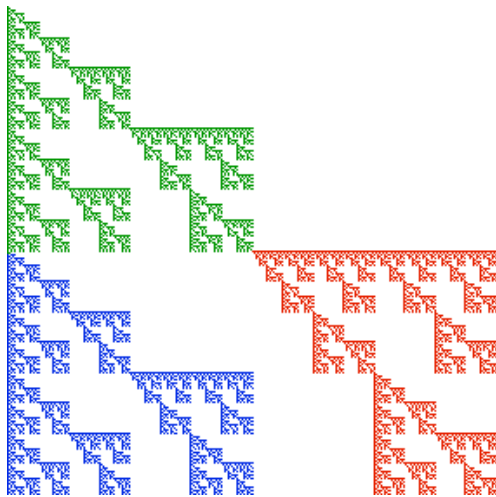


El triangle de Sierpinski es pot descomposar en tres còpies escalades amb un factor 0.5. Si permetem també que les còpies estiguin rotades o reflectides, inclús mantenint els factors d'escala igual a 0.5 i que les tres imatges del quadrat $[0,1] \times [0,1]$ siguin igualment $[0,0.5] \times [0,0.5]$, $[0.5,1] \times [0,0.5]$ i $[0,0.5] \times [0.5,1]$, podem generar tota una família de fractals amb diverses propietats topològiques!

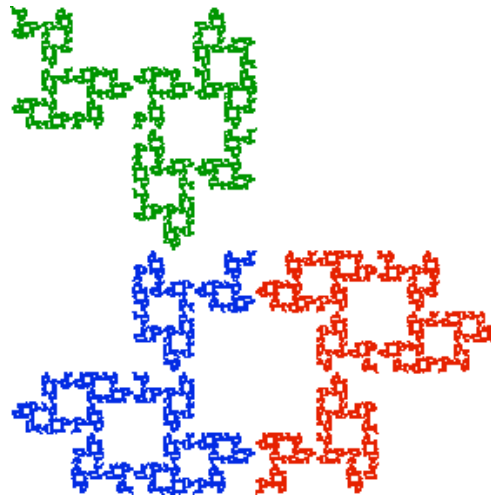
Quatre membres de la família Sierpinski

Q-17: En la següent figura hi ha quatre fractals de la família Sierpinski, que heu de generar. S'han de fer algunes rotacions de 90 graus i reflexions. Els colors (en la còpia, les tonalitats) no formen part del fractal, i són per visualitzar les diferents afinitats. Més endavant farem el primer exemple.

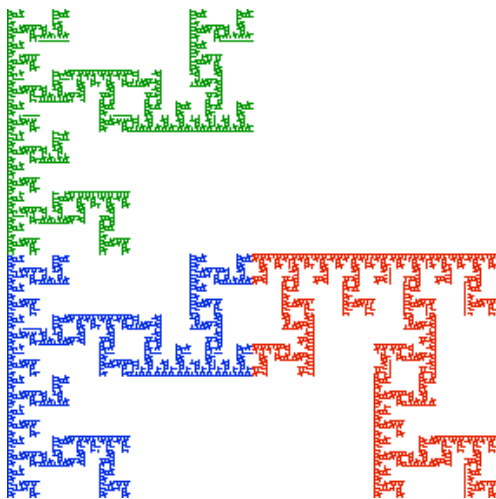
¹ La **Topologia** és la part de les Matemàtiques que estudia les propietats dels objectes geomètrics que no canvien en deformar-los contínuament.



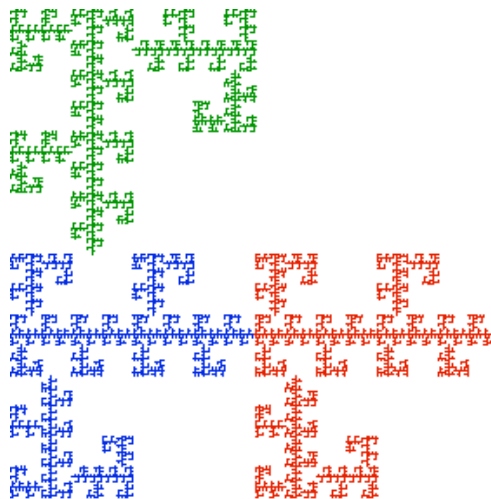
Dendrita



Cantorià



Fractal multiplement connex



Fractal híbrid

- La **dendrita** és un conjunt connex, no conté loops i té infinits punts de ramificació.
- El **cantorià** és un conjunt **totalment disconnex** i sense cap punt isolat. Així, observem que cap punt d'un color està en contacte amb cap punt d'un altre color, i la autosimilitud porta aquesta disconnexió a tots els nivells.
- En el fractal **multiplement connex**, la còpia blava connecta amb la vermella i la verda, i de fet en el primer cas ho fa en infinits punts. A més, hi ha infinits “loops” entre les còpies blava i vermella, que encerclen forats del conjunt.
- El fractal **híbrid** no és connex perquè la còpia verda no toca cap de les altres dues còpies. Però les còpies blava i vermella sí que connecten en un punt. Observem que també conté un segment (de $(0,1/3)$ a $(1,1/3)$), i de fet conté infinits segments (per què?). Així doncs, aquest fractal no és cantorià.

La dendrita

En aquesta secció veurem com generar la dendrita i (esperem) pot ser d'utilitat per generar la resta de fractals de la família Sierpinski.

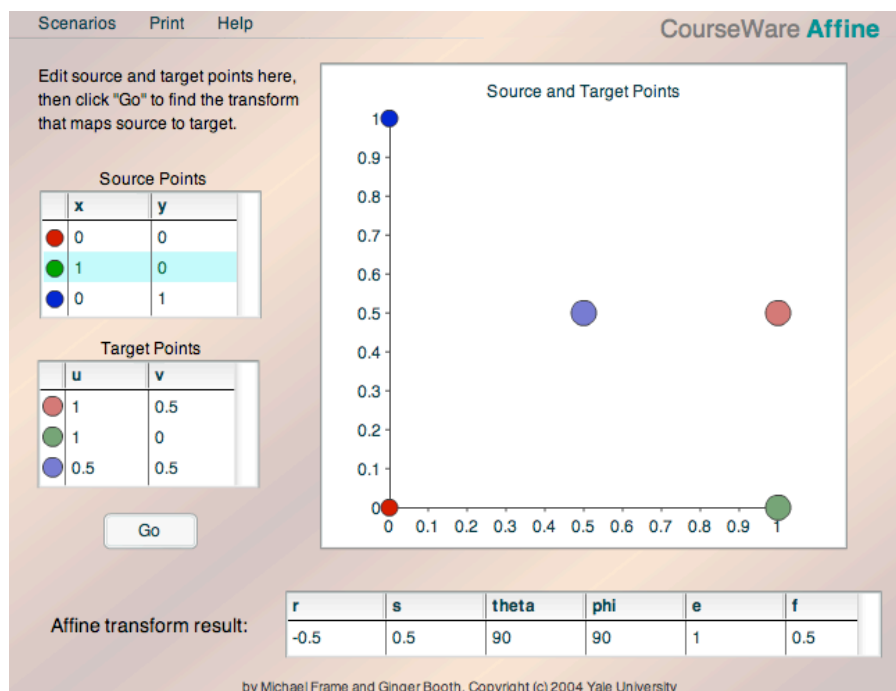
Recordem primer que està feta de tres còpies de l'original, escalades per un factor 0.5, però en diferents orientacions i posicions.

Observant el dibuix veiem que, respecte a la taula que genera el Triangle de Sierpinski, només hem de modificar la segona afinitat (corresponent al color vermell). Notem que la component vermella està enquadrada a $[0.5, 1] \times [0, 0.5]$, però no només és una còpia escalada del fractal, sinó que també hem de fer una rotació d'una reflexió!

Podem jugar a trobar aquesta afinitat anant modificant-la a la taula i provant amb la primera generació, tal com vam fer al primer exemple. Com pot ser una mica complicat, si no ens surt podem usar el software de <http://classes.yale.edu/Fractals/Software/affine/affine.html>, per determinar l'afinitat indicant a on "van a parar" tres punts del pla. En aquest cas, la segona afinitat ...

- Envia el punt (0,0) al (1,0.5)
- Envia el punt (1,0) al (1,0)
- Envia el punt (0,1) al (0.5,0.5)

Posant les dades al applet, obtenim ...



Així, doncs, una taula de la dendrita és

afinitat	r	s	θ	φ	e	f
1	0.5	0.5	0	0	0	0
2	-0.5	0.5	90	90	1	0.5
3	0.5	0.5	0	0	0	0.5

La següent taula també ho és!

afinitat	r	s	θ	φ	e	f
1	0.5	0.5	0	0	0	0
2	0.5	0.5	-90	90	1	0.5
3	0.5	0.5	0	0	0	0.5

La genealogia de la família Sierpinski

Per acabar, us proposem el següent ...

Projecte: Computar **tots** els membres de la família Sierpinski, i estudiar les seves propietats topològiques.

Treballar en equip pot anar bé ... perquè, quants membres té la família Sierpinski?

Apèndix: Afinitats

Per generar fractals, utilitzarem unes aplicacions del pla anomenades **afinitats**. Una afinitat es pot descriure com una aplicació de la forma

$$\begin{aligned}u &= a x + b y + e, \\v &= c x + d y + f,\end{aligned}$$

o, usant notació matricial,

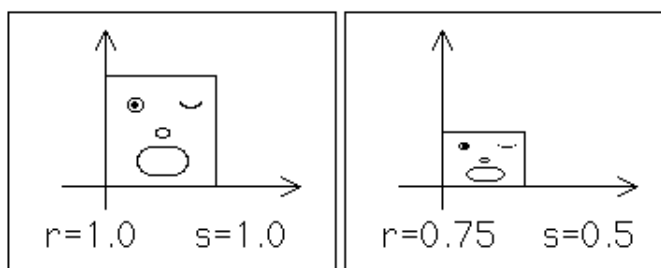
$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

que ens diu que el punt (x,y) “va a parar” o “és aplicat” al punt (u,v) . Els coeficients a,b,c,d,e,f determinen l’afinitat. Podem descomposar qualsevol afinitat del pla en una composició d’un **escalament**, una **rotació** i una **translació** (en aquest ordre). Passem ara a descriure cadascuna d’aquestes afinitats simples.

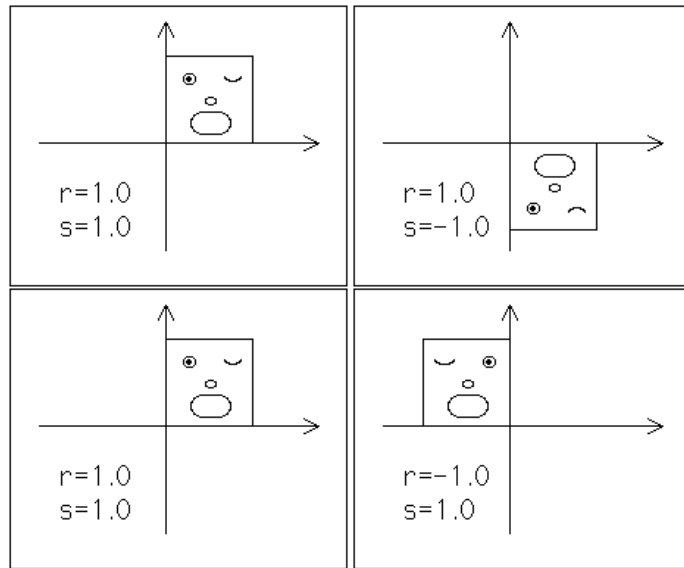
Un **escalament** està descrit per un factor d’escala r respecte a l’eix x i un factor d’escala s respecte a l’eix y . Les seves equacions són, doncs,

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} r & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

de manera que la seva acció és, per exemple:



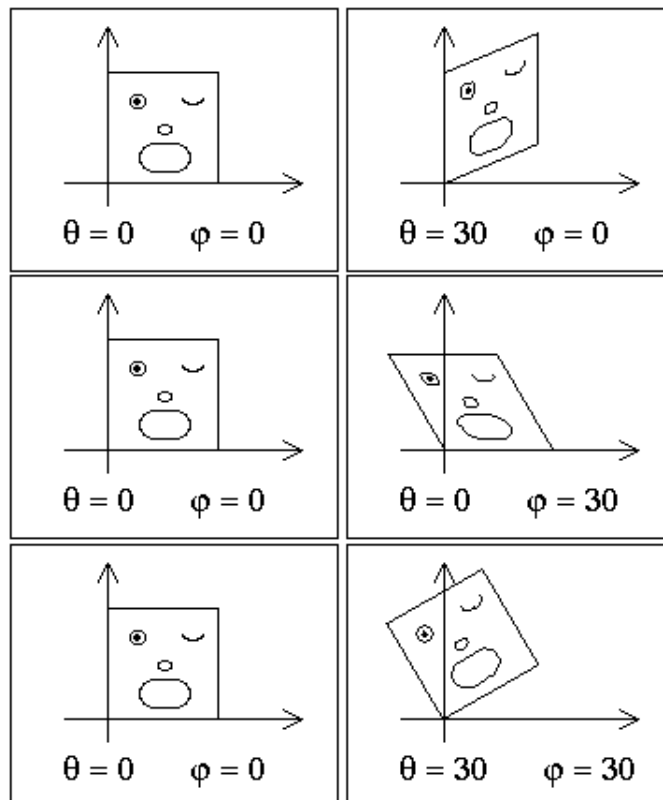
Els factors d’escala poder ser negatius, inclouen les **reflexions**. Per exemple, si $r = -1$ i $s = +1$ tenim una reflexió respecte a l’eix y , i si $r = +1$ i $s = -1$ tenim una reflexió respecte a l’eix x . Què obtenim si $r = s = -1$?



Una **rotació** respecte l'origen està descrita per dos angles. (Mesurarem els angles en graus). L'angle θ dóna la rotació de les rectes horitzontals i l'angle φ la de les rectes verticals. Si $\theta = \varphi$ obtenim una **rotació rígida**. Les seves equacions són

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \varphi \\ \sin \theta & \cos \varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

de forma que, per exemple,

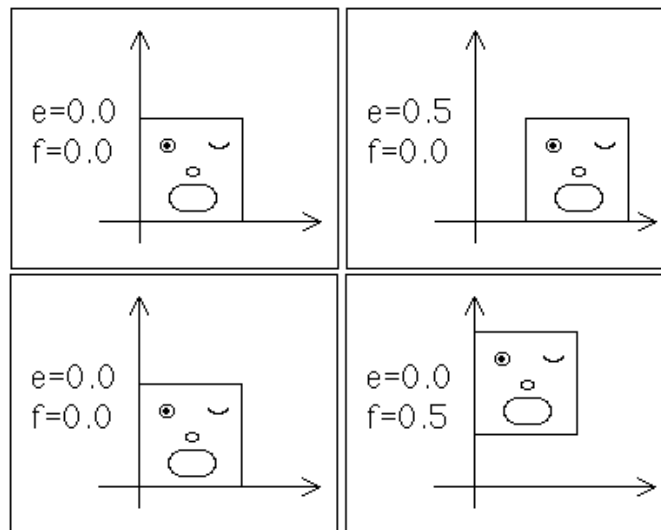


Observem que una rotació amb $\theta = 180$, $\varphi = 0$ correspon a una reflexió respecte a l'eix y , i que $\theta = 0$, $\varphi = 180$ és una reflexió respecte a l'eix x . A què correspon una rotació rígida de 180 graus (amb $\theta = \varphi = 180$)

En una **translació** de vector (e, f) , e mesura la translació horitzontal i f la translació vertical. Així, la seva equació és

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

produint els següents efectes:



Així, una afinitat també està determinada pels coeficients $r, s, \theta, \varphi, e, f$. Si $r = s$ i $\theta = \varphi$, l'afinitat es diu **semblança**.

Els coeficients a, b, c, d, e, f i els $r, s, \theta, \varphi, e, f$, estan obviament relacionats:

$$\begin{aligned} a &= r \cos(\theta), & b &= -s \sin(\varphi) \\ c &= r \sin(\theta), & d &= s \cos(\varphi) \end{aligned}$$

Una altra manera de determinar una afinitat és donant “on van a parar tres punts no alineats del pla”. Per exemple, si

- $(0,0)$ va a parar al punt $(0.5,0)$
- $(1,0)$ va a parar al punt $(0.5,0.75)$
- $(0,1)$ va a parar al punt $(0,0)$

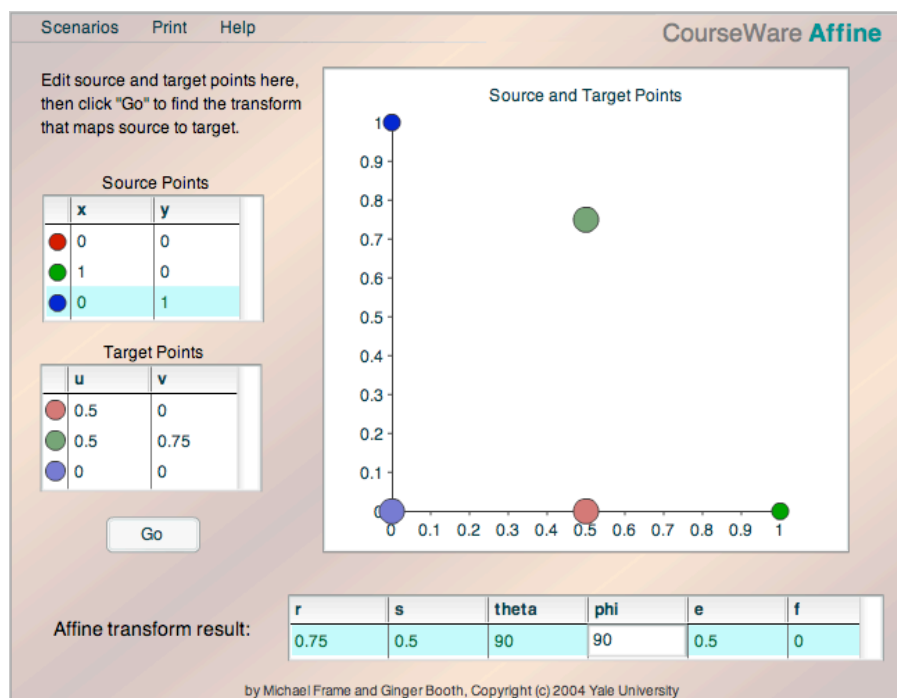
llavors l'afinitat ve donada pels coeficients

$$a = 0, b = -0.5, c = 0.75, d = 0, e = 0.5, f = 0$$

o, equivalentment,

$$r = 0.75, s = 0.5, \theta = 90, \varphi = 90, e = 0.5, f = 0.$$

Podem resoldre el problema utilitzant el software de <http://classes.yale.edu/Fractals/Software/affine/affine.html>.



Q-19: Observem que, per exemple,

$$r = 0.75, s = -0.5, \theta = 90, \varphi = -90, e = 0.5, f = 0$$

també és solució. Per què?

Q-20: És natural pensar que les **afinitats contractives** són les que $|r| < 1$ i $|s| < 1$. Ara bé, l'afinitat

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 1/2 & -1.25/\sqrt{2} \\ 0 & 1.25/\sqrt{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

és contractiva, encara que

$$r = 0.5, s = 1.25, \theta = 0, \varphi = 45, e = 0, f = 0.$$

Podries donar una caracterització més acurada de les afinitats contractives? És possible que $|r| < 1$, $|s| < 1$ i que l'afinitat no sigui contractiva?

