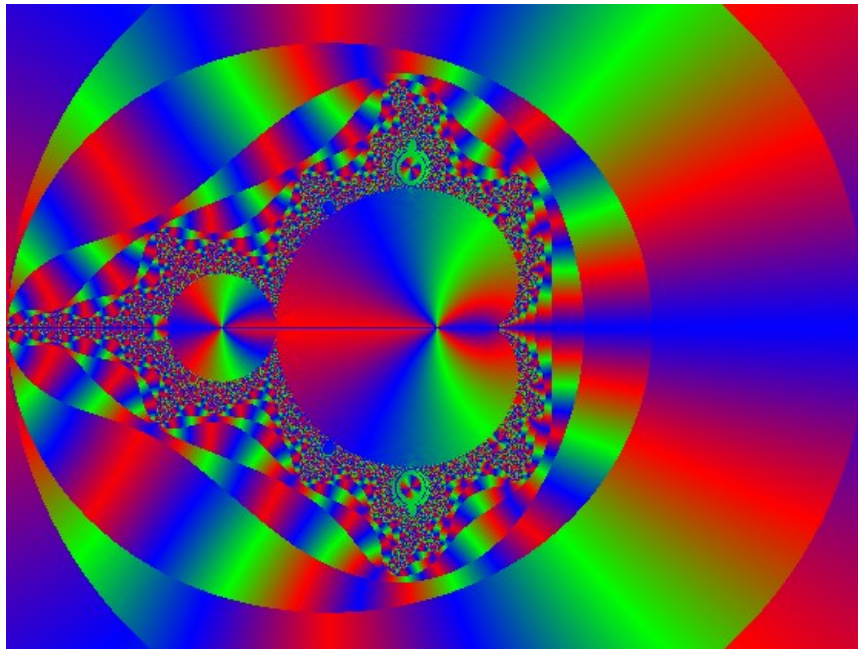


Fractal10

Guida alle Formule

Aprire, capire e scrivere formule .FRM



*La formula minima: $z = z*z + c$*

Compatibile con i file di Fractint e WinFract

Sommario

Sommario	2
1. Che cos'è una formula	5
Il formato .FRM.....	5
Che cosa trovi già installato.....	5
2. Aprire e usare una formula.....	6
Passo 1: aprire il file.....	7
Passo 2: scegliere la formula	7
Passo 3: metterla in uso	7
Passo 4: da qui in poi è un frattale come gli altri	7
Salvare le modifiche	7
3. Anatomia di una formula.....	8
Le quattro parti.....	8
Come si leggono i separatori	9
Che cosa succede a ogni punto dell'immagine	9
Nome, simmetria e direttive	9
Commenti	10
4. Il linguaggio: variabili e operatori	11
Variabili predefinite	11
Operatori	12
5. Le funzioni.....	13
Due esempi che cambiano tutto	14
Le funzioni trascendenti	15
Una funzione con un'oddità storica	15
6. Condizioni e rami	16
Le parole chiave.....	16
Le condizioni	17
Il ciclo WHILE	17
7. Il bailout.....	18
Le forme ammesse	18

Che effetto ha sull'immagine	18
8. La simmetria	20
Il problema delle dichiarazioni false	20
L'interruttore nella tab Formule	21
9. Scrivere la prima formula	22
Passo 1: partire da una copia	22
Passo 2: cambiare l'esponente	22
Passo 3: rendere l'esponente regolabile	22
Passo 4: aggiungere una condizione	23
Passo 5: annotare e salvare	24
Da qui in avanti	24
10. Colorare una formula	25
Colorazione interna ed esterna	25
Potenziale e stima della distanza	26
Vedere l'orbita	26
11. Quando qualcosa non funziona	27
La formula non compila	27
La formula compila ma l'immagine è tutta nera	27
L'immagine è tutta di un colore uniforme	27
L'immagine cambia aspetto mentre ingrandisci	27
Il risultato non coincide con WinFract	28
Vedere il codice generato	28
12. IFS e L-System	29
IFS: sistemi di funzioni iterate	29
L-System: regole di riscrittura	29
Appendice A — Tutto quello che il linguaggio accetta	30
Struttura (5/5)	30
Simmetria (4/4)	30
Variabili (11/11)	30
Operatori (6/6)	31
Controllo di flusso (4/4)	31

Bailout (5/5).....	31
Funzioni (31/31)	32
Appendice B — Le formule incluse.....	33
fractal10.frm (22)	33
fractal10_extra.frm (18)	33
fractal10_numeri.frm (15).....	33
fractal10_teoremi_fix.frm (3).....	33
Appendice C — Promemoria	34
Struttura	34
Da ricordare.....	34
Scheletri pronti	34

1. Che cos'è una formula

Fractal10 arriva con oltre cento tipi di frattale già pronti, ma il suo motore sa fare una cosa in più: leggere una regola matematica scritta da te e disegnarla. Quella regola è una formula.

Una formula dice al programma che cosa fare, per ogni punto dell'immagine, ripetendo lo stesso calcolo molte volte. Il colore del punto dipende da come si comporta quel calcolo: se il valore scappa verso l'infinito, dopo quante ripetizioni, o se resta imprigionato per sempre.

Il formato .FRM

Le formule stanno in file di testo con estensione .FRM. Un singolo file ne può contenere decine, una dopo l'altra. È lo stesso formato usato da Fractint e WinFract dagli anni Novanta: i file scritti allora si aprono oggi senza conversioni, e le formule scritte con Fractal10 restano leggibili da quei programmi.

Non c'è differenza fra «le formule dei file» e «le formule che scrivi tu»: sono la stessa cosa. Scrivere una formula nuova significa aggiungere un blocco di testo a un file .FRM, con l'editor incorporato o con un qualsiasi editor di testo.

Che cosa trovi già installato

Nella cartella formulas del programma ci sono quattro file, per un totale di 58 formule pronte all'uso, organizzate per famiglia e commentate. Sono anche il miglior punto di partenza per scriverne di nuove: aprirne una e modificarla è più facile che partire dal foglio bianco. L'elenco completo è nell'Appendice B.

File	Contenuto
fractal10.frm	Raccolta principale: divergenti classiche, convergenti di tipo Newton e Magnet, ibride e trascendenti.
fractal10_extra.frm	Varianti e formule sperimentali.
fractal10_numeri.frm	Formule costruite attorno a proprietà numeriche.
fractal10_teoremi_fix.frm	Formule legate a teoremi matematici.

Nota — Puoi aggiungere i tuoi file .FRM nella stessa cartella: l'aggiornamento del programma non li tocca.

2. Aprire e usare una formula

Tutto quello che riguarda le formule sta nella tab Formule. Il percorso è sempre lo stesso: si apre un file, si sceglie una formula dall'elenco, la si mette in uso. Da quel momento si comporta come qualsiasi altro frattale del programma.

Apri .FRM...Apri .IFS...Apri .L...

(nessun file)

Formule Disponibili

Editor Formula

Usa FormulaSalvaSalva Come...

EliminaDebug Code

☒ **Simmetria stile WinFract (XYAXIS di default)**
Formula con simmetria dichiarata → usa quella. Senza dichiarazione: ON → XYAXIS (come WinFract); OFF → rileva la simmetria vera (XAXIS/YAXIS/XYAXIS/ORIGIN o nessuna).
☐ **Singolarità come inside (immagine pulita)**
Formule degeneri/esplosive (es. z/z , coefficienti enormi).
OFF (default) → la singolarità è escape, fedele a Fractint.
ON → la zona singolare diventa inside, immagine pulita (non identica a WinFract).

Passo 1: aprire il file

Il pulsante Apri .FRM... apre la finestra di selezione, che parte dalla cartella formulas. Scelto il file, tutte le formule contenute compaiono nell'elenco, in ordine alfabetico, con accanto fra parentesi la simmetria dichiarata.

Passo 2: scegliere la formula

Cliccando su una voce dell'elenco il testo della formula compare nell'editor sottostante. Da lì si può leggere e, volendo, modificare.

Passo 3: metterla in uso

Il pulsante Usa Formula compila il testo dell'editor e lo imposta come frattale corrente. La compilazione traduce la formula in codice macchina: la prima volta richiede un paio di secondi, poi il risultato resta in memoria e i render successivi sono immediati.

Se il testo contiene un errore la compilazione si ferma e viene mostrato un messaggio. Il capitolo 11 spiega come interpretarlo.

Passo 4: da qui in poi è un frattale come gli altri

Una volta in uso, la formula compare fra i tipi di frattale e risponde a tutti i comandi del programma:

- zoom e navigazione con il mouse, come su Mandelbrot;
- numero di iterazioni e risoluzione dalla tab Frattale;
- i parametri p1-p5 dalla tab Parametri, se la formula li usa;
- palette, rotazione dei colori, metodi di colorazione interna ed esterna;
- esportazione in PNG, GIF, MP4 e salvataggio dei preset.

Nota — Il PNG salvato ricorda anche da quale file e con quale nome è stata presa la formula: ricaricando l'immagine, il programma ritrova la formula e ricostruisce la vista.

Salvare le modifiche

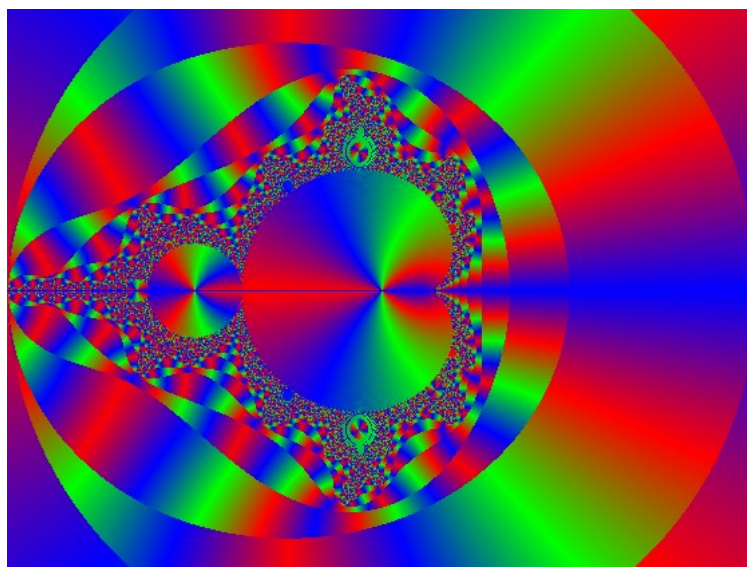
Controllo	Tipo	Default	Funzione
Salva	Pulsante	—	Riscrive la formula modificata dentro il file .FRM di provenienza, al posto della versione precedente.
Salva Come...	Pulsante	—	Salva in un file .FRM a scelta, lasciando intatto l'originale. È il modo prudente di conservare una variante.
Elimina	Pulsante	—	Toglie la formula selezionata dal file.

Nota — Conviene tenere le proprie formule in un file separato, per esempio mie_formule.frm, invece di modificare quelli installati: restano più facili da ritrovare e da portare su un altro computer.

3. Anatomia di una formula

Ogni formula ha la stessa struttura. Ecco la più semplice possibile, quella che disegna l'insieme di Mandelbrot:

```
Base {  
  z = 0, c = pixel:  
  z = z*z + c  
  |z| <= 4  
}
```



*La formula minima: $z = z*z + c$*

Le quattro parti

Parte	Nell'esempio	A cosa serve
Nome	Base	Identifica la formula. È il nome che comparirà nell'elenco e fra i tipi di frattale.
Inizializzazione	$z = 0, c = \text{pixel}$	Eseguita una sola volta per ogni punto, prima di cominciare. Prepara i valori di partenza. Termina con i due punti.
Ciclo	$z = z*z + c$	Ripetuto a ogni iterazione. È il cuore della formula: la regola che viene applicata più e più volte.
Bailout	$ z \leq 4$	Condizione di continuazione. Finché è vera si va avanti; appena diventa falsa il punto è «sfuggito» e si passa al successivo.

Come si leggono i separatori

I due punti separano l'inizializzazione dal ciclo. L'ultima riga prima della graffa chiusa è sempre il bailout. In molti file scritti per Fractint troverai anche una virgola a separare il ciclo dal bailout: entrambe le scritture funzionano.

```
Nome(SIMMETRIA) {  
    inizializzazione :  
    ciclo  
    bailout  
}
```

Che cosa succede a ogni punto dell'immagine

Il programma ripete questo procedimento per ciascun pixel:

1. assegna alla variabile pixel la coordinata del punto sul piano complesso;
2. esegue una volta l'inizializzazione;
3. esegue il ciclo, e dopo ogni passaggio controlla il bailout;
4. se il bailout diventa falso, si ferma e colora il punto in base a quante iterazioni sono servite;
5. se non diventa mai falso entro il numero massimo di iterazioni, il punto è considerato interno all'insieme.

Nota — Questo spiega perché il numero di iterazioni conta tanto: è il limite di pazienza del programma. Alzandolo si distinguono punti che prima sembravano tutti interni.

Nome, simmetria e direttive

Il nome può essere seguito da una simmetria fra parentesi e da direttive fra parentesi quadre. Entrambe sono facoltative e vanno prima della graffa aperta:

```
Newton4(XYAXIS)[periodicity=0] {  
    ...  
}
```

La simmetria è spiegata nel capitolo 8. La direttiva `periodicity=0` disattiva un'ottimizzazione che, sulle formule che convergono verso un punto fisso invece di sfuggire, produrrebbe un risultato sbagliato: serve alle formule di tipo Newton.

Commenti

Il punto e virgola introduce un commento: tutto quello che segue, fino a fine riga, viene ignorato. Vale sia fuori sia dentro le formule, ed è il modo per annotare a cosa serve un parametro o quale vista conviene.

```
; --- Le mie formule ---  
Prova {  
    ; p1 controlla la forma; provare (-0.4, 0.6)  
    z = pixel, c = p1:  
    z = z*z + c          ; il passo di iterazione  
    |z| <= 4  
}
```

4. Il linguaggio: variabili e operatori

Tutti i valori con cui lavora una formula sono numeri complessi, cioè coppie di numeri: una parte reale e una immaginaria. Anche quando scrivi un numero solo, come 2, il programma lo tratta come il complesso (2, 0).

Variabili predefinite

Alcuni nomi sono già pronti all'uso e non vanno inizializzati. Sono quelli con cui la formula comunica con il resto del programma.

Nome	Funziona	Descrizione
pixel	Sì	Coordinata del punto in esame, costante durante il ciclo
p1 e p2	Sì	I primi due parametri complessi impostabili dalla tab Parametri
da p3 a p5	Sì	Gli altri tre parametri complessi
P1 maiuscolo	Sì	I nomi dei parametri non distinguono maiuscole e minuscole
LastSqr	Sì	Modulo quadro dell'ultimo <code>sqr()</code> calcolato
maxit	Sì	Numero massimo di iterazioni impostato nell'interfaccia
e	Sì	Costante di Nepero
pi	Sì	Costante pi greco
scrnpix	Sì	Coordinate del pixel sullo schermo, in colonna e riga
whitesq	Sì	Vero sui pixel «bianchi» della scacchiera schermo
Variabili libere	Sì	Si possono definire variabili con nome a piacere

Oltre a questi puoi definire tutte le variabili che vuoi, con il nome che preferisci: basta assegnare loro un valore nell'inizializzazione.

Nota — I parametri p1-p5 sono il ponte fra la formula e l'interfaccia: quello che scrivi nei campi P1-P5 della tab Parametri finisce lì. Una formula che li usa si può esplorare senza riscriverla.

Operatori

Operatore	Funziona	Descrizione
+ - * /	Sì	Le quattro operazioni fra numeri complessi
^ (potenza)	Sì	Elevamento a potenza, anche con esponente complesso
\ (divisione)	Sì	Barra rovesciata come sinonimo di divisione
 z modulo quadro	Sì	Le barre verticali danno il modulo AL QUADRATO, non il modulo
Confronti	Sì	Minore, maggiore, minore o uguale, maggiore o uguale, uguale, diverso
&& e 	Sì	Congiunzione e disgiunzione logica nelle condizioni

Attenzione alle barre verticali

Nelle formule Fractint la scrittura $|z|$ non è il modulo, ma il modulo AL QUADRATO. È una convenzione storica che sorprende chi arriva dalla matematica, e spiega perché il bailout classico si scriva $|z| \leq 4$ e non $|z| \leq 2$: quattro è il quadrato di due.

Se ti serve il modulo vero puoi usare `abs()`.

5. Le funzioni

Le funzioni si applicano a un valore complesso e ne restituiscono un altro. Sono il modo più diretto per cambiare radicalmente l'aspetto di un frattale: sostituire una funzione con un'altra, nella stessa formula, produce un'immagine completamente diversa.

Fractal10 ne riconosce 31, verificate una per una:

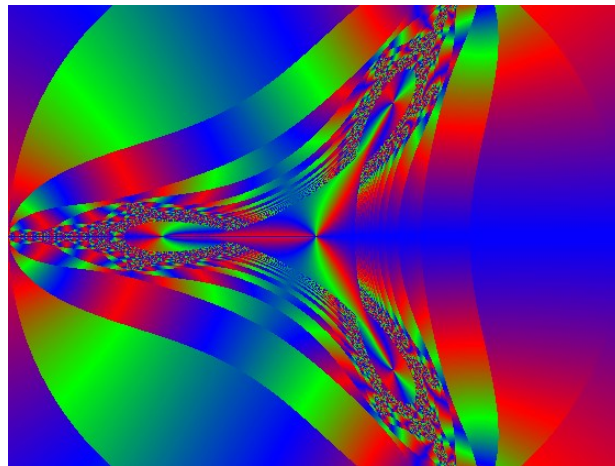
Funzione	Funziona	Descrizione
<code>sin()</code>	Sì	Seno
<code>sen()</code>	Sì	Seno, sinonimo in italiano e spagnolo
<code>cos()</code>	Sì	Coseno
<code>tan()</code>	Sì	Tangente
<code>cot()</code>	Sì	Cotangente
<code>sinh()</code>	Sì	Seno iperbolico
<code>cosh()</code>	Sì	Coseno iperbolico
<code>tanh()</code>	Sì	Tangente iperbolica
<code>cotanh()</code>	Sì	Cotangente iperbolica
<code>asin()</code>	Sì	Arcoseno
<code>acos()</code>	Sì	Arcocoseno
<code>atan()</code>	Sì	Arcotangente
<code>exp()</code>	Sì	Esponenziale
<code>log()</code>	Sì	Logaritmo naturale
<code>sqrt()</code>	Sì	Radice quadrata
<code>raiz()</code>	Sì	Radice quadrata, sinonimo spagnolo
<code>sqr()</code>	Sì	Quadrato; aggiorna anche LastSqr
<code>abs()</code>	Sì	Valore assoluto
<code>cabs()</code>	Sì	Valore assoluto, sinonimo Fractint
<code>cosxx()</code>	Sì	Coseno nella variante storica di WinFract, con segno invertito
<code>trunc()</code>	Sì	Tronca alla parte intera
<code>floor()</code>	Sì	Arrotonda per difetto
<code>ceil()</code>	Sì	Arrotonda per eccesso
<code>round()</code>	Sì	Arrotonda all'intero più vicino
<code>ident()</code>	Sì	Identità: restituisce l'argomento invariato
<code>zero()</code>	Sì	Restituisce zero

one()	Sì	Restituisce uno
flip()	Sì	Scambia parte reale e immaginaria
conj()	Sì	Coniugato
real()	Sì	Parte reale
imag()	Sì	Parte immaginaria

Due esempi che cambiano tutto

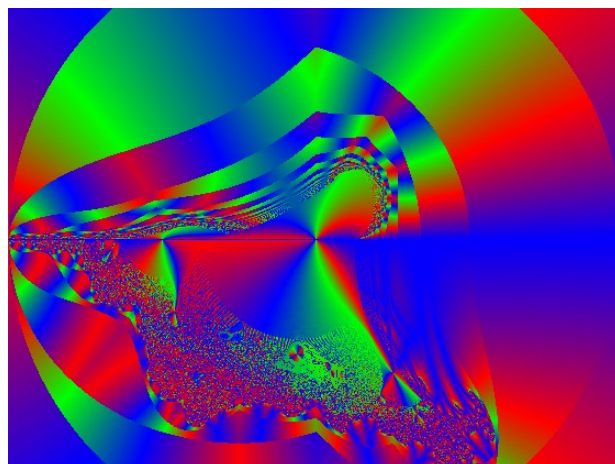
Partendo dalla formula base del capitolo 3, basta una funzione per ottenere frattali classici molto diversi.

```
z = conj(z)*conj(z) + c ; il Tricorn
```



*conj() al posto di z*z: il Tricorn*

```
z = (abs(real(z)) + flip(abs(imag(z))))^2 + c ; il Burning Ship
```

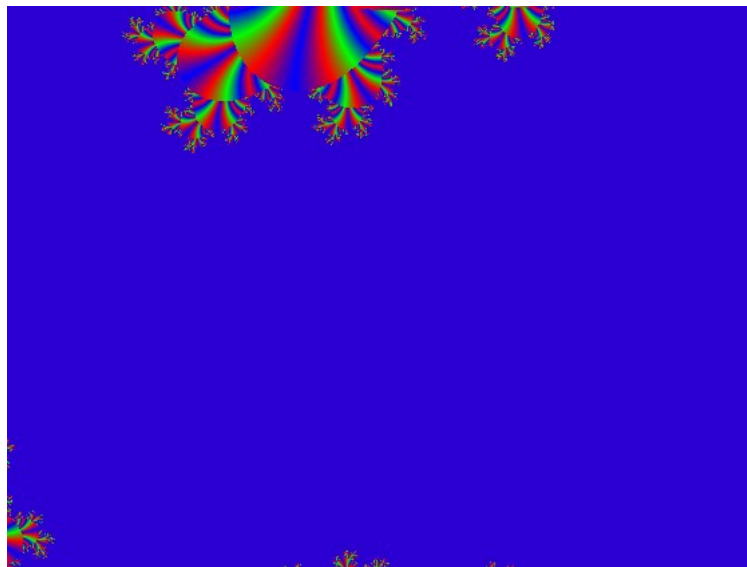


abs() sulle componenti: il Burning Ship

Le funzioni trascendenti

Seno, coseno, esponenziale e logaritmo producono strutture che si ripetono all'infinito lungo una direzione, molto diverse dai frattali polinomiali. Di solito richiedono un bailout più alto, perché i valori crescono in fretta.

```
Trascendente {  
  z = pixel, c = p1:  
  z = sin(z) + c  
  |z| <= 50  
}
```



Una funzione trascendente: $\sin(z) + c$

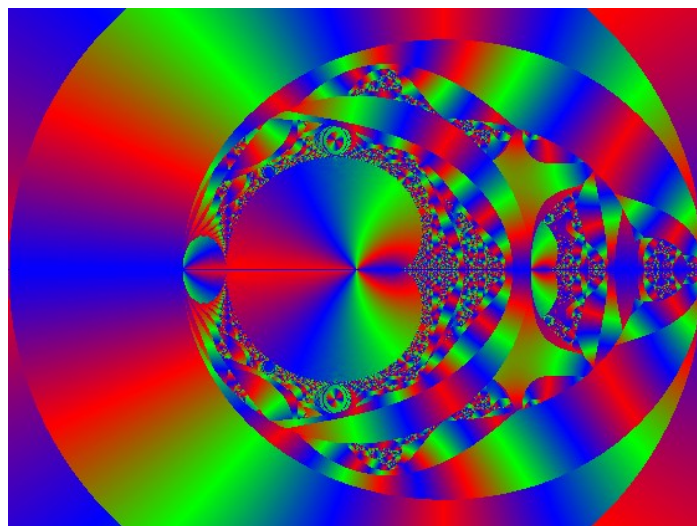
Una funzione con un'oddità storica

La funzione cosxx è una versione del coseno con un segno invertito, nata da un errore nelle prime versioni di Fractint e poi mantenuta apposta: molte immagini d'epoca sono state create con quella, e sostituirla con cos darebbe un risultato diverso. Serve solo a riprodurre fedelmente vecchi lavori.

6. Condizioni e rami

Il ciclo non deve per forza fare sempre la stessa cosa: può decidere in base al valore raggiunto. È quello che permette formule «ibride», che si comportano in un modo in una zona del piano e in un altro modo altrove.

```
Condizionale {  
  z = 0, c = pixel:  
  IF (|z| < 1)  
    z = z*z + c  
  ELSE  
    z = z*z - c  
  ENDIF  
  |z| <= 4  
}
```



Un ramo IF / ELSE dentro il ciclo

Le parole chiave

Costrutto	Funziona	Descrizione
IF / ENDIF	Sì	Esecuzione condizionale semplice
IF / ELSE / ENDIF	Sì	Due rami alternativi
ELSEIF	Sì	Più condizioni in cascata
WHILE / ENDWHILE	Sì	Ciclo interno all'iterazione

Ogni IF va chiuso da un ENDIF. Fra i due possono comparire quanti ELSEIF si vuole e al massimo un ELSE. I blocchi si possono annidare.

Le condizioni

Dentro le parentesi si scrivono confronti, combinabili con `&&` (e) oppure `||` (oppure):

```
IF (|z| < 1 && real(z) > 0)
...
ENDIF
```

Nota — Nei confronti Fractint usa solo la parte reale dei valori complessi. Scrivere `IF (z > 0)` significa quindi confrontare la parte reale di z con zero.

Il ciclo WHILE

Oltre ai rami condizionali si può ripetere un gruppo di istruzioni dentro la singola iterazione, con un ciclo WHILE chiuso da ENDWHILE. Serve alle formule che devono fare più passi di calcolo prima di controllare il bailout.

```
Ripetuta {
  z = 0, c = pixel, n = 0:
  n = 0
  WHILE (n < 3)
    z = z*z + c
    n = n + 1
  ENDWHILE
  |z| <= 4
}
```

La condizione segue le stesse regole di quella di IF, e dentro un WHILE si possono annidare blocchi IF e altri cicli.

Nota — Ricordati di far avanzare la variabile che controlla il ciclo, qui $n = n + 1$: senza, la condizione resterebbe vera per sempre. Per sicurezza Fractal10 interrompe comunque ogni ciclo dopo 10.000 ripetizioni, così una formula sbagliata produce un'immagine strana invece di bloccare il programma.

7. Il bailout

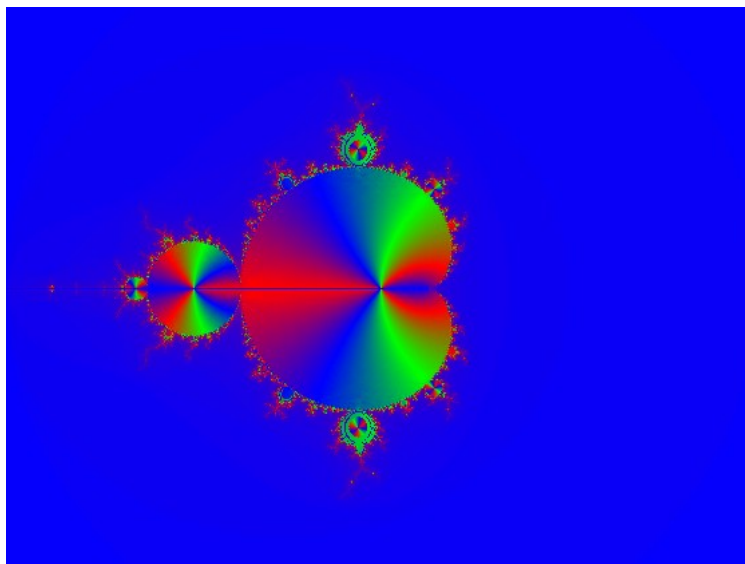
Il bailout è la condizione che decide quando fermarsi. Viene controllata dopo ogni iterazione: finché è vera si continua, appena diventa falsa il punto ha «sfuggito» e riceve un colore in base al numero di iterazioni impiegate.

Le forme ammesse

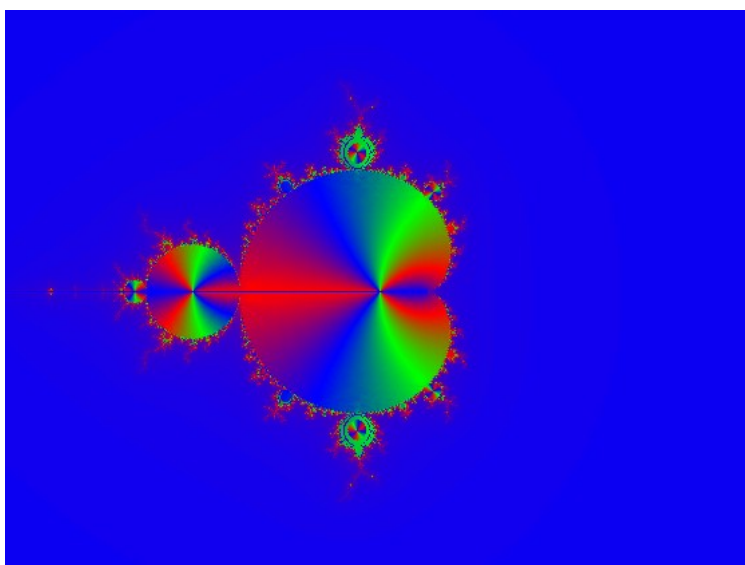
Forma	Funziona	Descrizione
$ z \leq \text{valore}$	Sì	Condizione di fuga classica sul modulo quadro
$\text{LastSqr} \leq \text{valore}$	Sì	Condizione basata sull'ultimo $\text{sqr}()$
$\text{abs}(z) < \text{valore}$	Sì	Condizione sul valore assoluto
Espressione con parametri	Sì	La soglia può essere calcolata
Condizione su $\text{real}()$	Sì	Il test può usare una componente

Che effetto ha sull'immagine

Il valore di soglia cambia la larghezza delle bande di colore attorno all'insieme. Un valore basso le stringe, uno alto le allarga e ammorbidisce il passaggio. La forma dell'insieme, invece, resta quasi la stessa.



Bailout $|z| \leq 4$, il valore classico



Stessa formula con bailout $|z| \leq 100$: le bande esterne si allargano

Nota — Quattro è il valore classico ed è quello che conviene usare se vuoi riprodurre immagini storiche. Le formule trascendenti richiedono di solito valori molto più alti, da 50 in su.

8. La simmetria

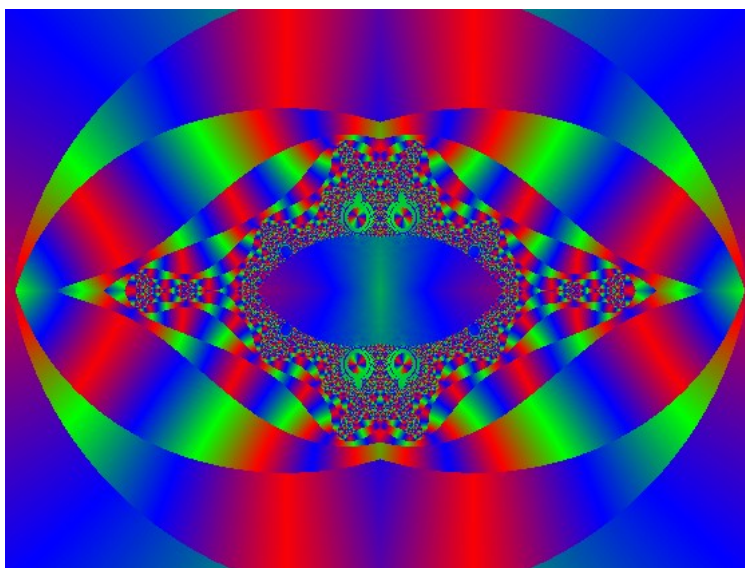
Molti frattali sono simmetrici: la metà inferiore dell'immagine è il riflesso di quella superiore. Quando è così, il programma può calcolare solo metà dei punti e specchiare il risultato, dimezzando il tempo di calcolo. Per farlo però deve saperlo in anticipo, e la formula glielo dice dichiarandolo accanto al nome.

Dichiarazione	Funziona	Descrizione
XAXIS	Sì	Simmetria rispetto all'asse reale
YAXIS	Sì	Simmetria rispetto all'asse immaginario
XYAXIS	Sì	Simmetria rispetto a entrambi gli assi
ORIGIN	Sì	Simmetria centrale rispetto all'origine

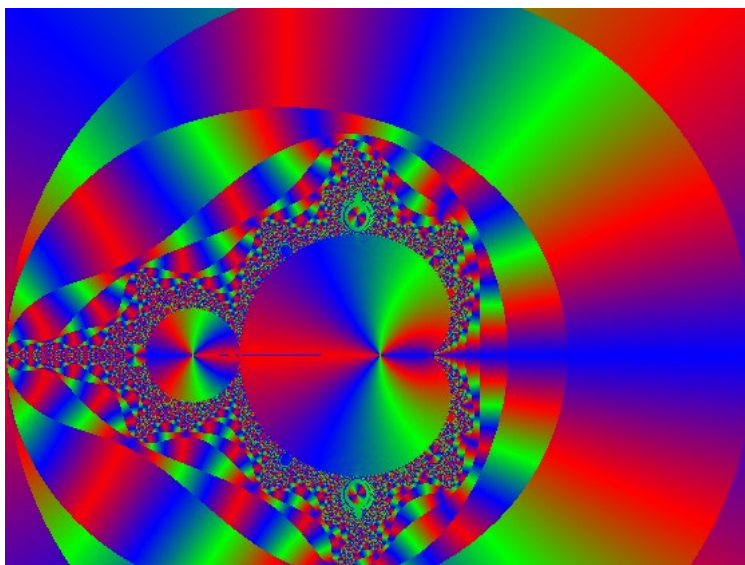
```
MiaFormula(XAXIS) {  
    ...  
}
```

Il problema delle dichiarazioni false

La dichiarazione non viene verificata: è una promessa che la formula fa al programma. Molte formule scritte a mano negli anni Novanta dichiarano una simmetria che in realtà non hanno, spesso per copia e incolla da un'altra formula. Il risultato è un'immagine specchiata che non corrisponde al frattale vero.



La formula dichiara XYAXIS e la simmetria viene applicata: metà immagine è specchiata



Stessa formula calcolata senza simmetria: il frattale vero è diverso, la dichiarazione era falsa

Le due immagini vengono dalla stessa identica formula. La prima applica la simmetria XYAXIS dichiarata; la seconda calcola ogni punto davvero. La dichiarazione era falsa, e la differenza si vede.

L'interruttore nella tab Formule

In fondo alla tab Formule la casella «Simmetria stile WinFract (XYAXIS di default)» decide come comportarsi:

Stato	Comportamento
Attiva (predefinito)	Se la formula dichiara una simmetria, viene usata quella. Se non dichiara nulla, si assume XYAXIS. È il comportamento di WinFract, e va tenuto per riprodurre fedelmente immagini d'epoca.
Disattiva	Il programma verifica quale simmetria la formula possieda davvero e usa solo quella. L'immagine è il frattale vero, anche se diverso da quello che mostrava WinFract.

Nota — Il sintomo tipico di una dichiarazione falsa è un'immagine che «salta» fra due aspetti diversi mentre si ingrandisce, a seconda che l'asse di simmetria sia o no dentro l'inquadratura. In quel caso disattiva la casella.

9. Scrivere la prima formula

Il modo più rapido per cominciare non è partire dal foglio bianco, ma modificare qualcosa che già funziona. Questo capitolo costruisce una formula in cinque passi, ognuno verificabile subito a schermo.

Passo 1: partire da una copia

Apri un file .FRM, scegli una formula semplice e premi subito Salva Come... per salvarla in un file tuo, per esempio mie_formule.frm. Da qui in poi lavori su quello e l'originale resta intatto.

Nell'editor cambia il nome, altrimenti avrai due formule con lo stesso nome:

```
MiaPrima {  
  z = 0, c = pixel:  
  z = z*z + c  
  |z| <= 4  
}
```

Premi Usa Formula e poi GENERA FRATTALE: devi vedere il Mandelbrot. Se lo vedi, la catena funziona e puoi cominciare a cambiare le cose.

Passo 2: cambiare l'esponente

La modifica più istruttiva è anche la più piccola. Sostituisci il prodotto con una potenza e alza l'esponente:

```
z = z^3 + c
```

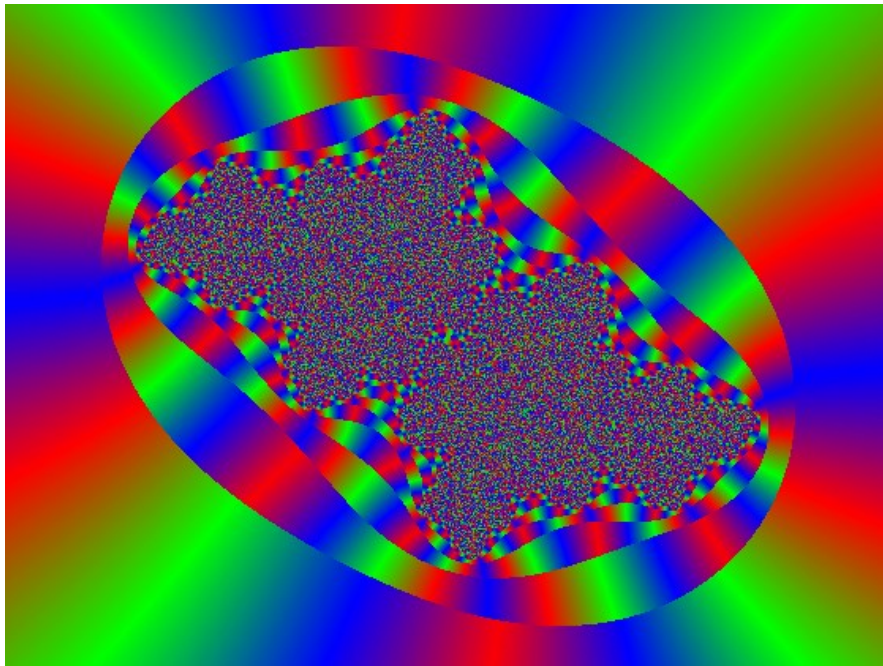
Rigenera. L'insieme ha ora due lobi invece di uno: un frattale con esponente n ha n meno uno bulbi principali. Prova con 4 e con 5.

Passo 3: rendere l'esponente regolabile

Invece di riscrivere la formula ogni volta, fai controllare l'esponente da un parametro:

```
MiaPrima {  
  z = 0, c = pixel:  
  z = z^p1 + c  
  |z| <= 4  
}
```

Ora nella tab Parametri il campo P1 è attivo. Scrivi 3 nella parte reale, rigenera; poi 4, poi 2.5. Un valore non intero produce una figura sfrangiata: l'esponente complesso è perfettamente lecito.



Un parametro cambia la figura senza toccare la formula.

Passo 4: aggiungere una condizione

Facciamo comportare la formula in due modi diversi a seconda di dove si trova il valore:

```
MiaPrima {
  z = 0, c = pixel:
  IF (|z| < 1)
    z = z^p1 + c
  ELSE
    z = z^p1 - c
  ENDIF
  |z| <= 4
}
```

Il confine fra i due comportamenti diventa visibile nell'immagine come una discontinuità: è la circonferenza di raggio uno.

Passo 5: annotare e salvare

Prima di salvare, scrivi nei commenti che cosa fa il parametro e quali valori vale la pena provare. Fra sei mesi non te lo ricorderai, e un commento costa una riga:

```
MiaPrima {  
  ; p1 = esponente; interessante fra 2 e 6  
  ; con p1 non intero l'insieme si sfrangia  
  z = 0, c = pixel:  
  ...  
}
```

Premi Salva. La formula è nel tuo file e comparirà nell'elenco ogni volta che lo aprirai.

Da qui in avanti

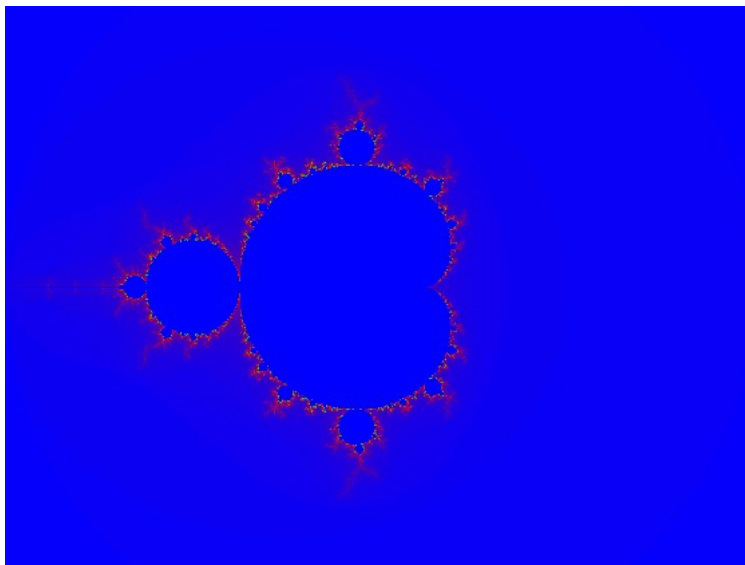
- Sostituisci z^{p1} con una funzione: sin, exp, conj, abs. L'Appendice A elenca tutto quello che puoi usare.
- Cambia l'inizializzazione da $z = 0, c = \text{pixel}$ a $z = \text{pixel}, c = p1$: trasformi un insieme di Mandelbrot nel corrispondente insieme di Julia.
- Alza il bailout e osserva come si allargano le bande.
- Prova a disattivare la casella della simmetria e verifica se la tua formula è davvero simmetrica.

10. Colorare una formula

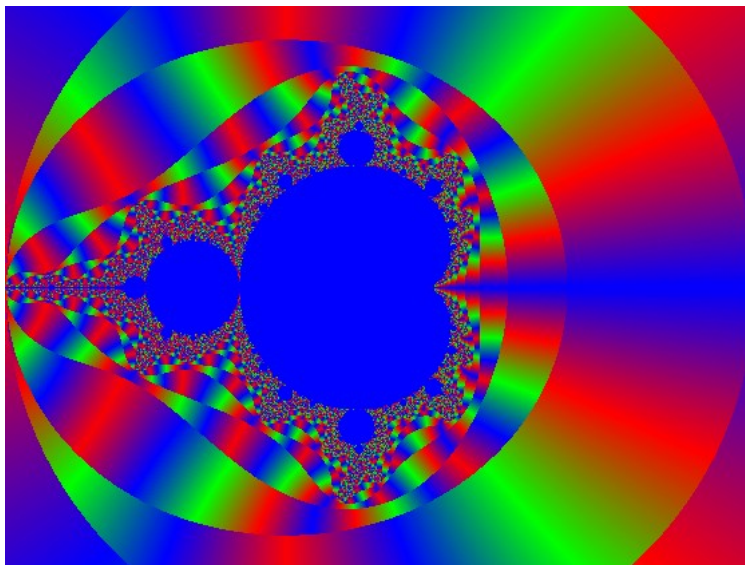
Le formule custom non sono cittadini di seconda classe: tutti i metodi di colorazione del programma funzionano anche su di loro, senza che la formula debba prevedere nulla.

Colorazione interna ed esterna

Nella tab Parametri, il gruppo «Metodi di Colorazione» decide come tradurre in colore quello che è successo durante l'iterazione. La colorazione esterna riguarda i punti sfuggiti, quella interna i punti rimasti intrappolati.



Colorazione esterna ITER: bande di iterazione



Stessa immagine con colorazione esterna ATAN

Sono la stessa formula e la stessa vista: cambia solo il metodo di colorazione esterna. Il dettaglio di tutti i metodi è nella Guida all'Interfaccia.

Potenziale e stima della distanza

Nella tab Rendering due opzioni cambiano il modo di calcolare il colore invece che di interpretarlo, e funzionano anche sulle formule custom:

- «Attiva Rendering Potenziale» produce sfumature continue al posto delle bande;
- «Attiva Stima Distanza» colora in base alla distanza dal bordo dell'insieme, mantenendo visibili i filamenti sottili anche a forte ingrandimento. Per riuscirci il programma calcola da solo la derivata della tua formula.

Nota — La derivata viene ricavata simbolicamente dal testo della formula. Su formule molto contorte il calcolo può non riuscire: in quel caso l'opzione non produce l'effetto atteso e conviene tornare alla colorazione normale.

Vedere l'orbita

La visualizzazione dell'orbita, nella tab Navigazione, funziona anche sulle formule custom. È lo strumento più utile per capire perché una formula si comporta in un certo modo: Shift più click su un punto mostra la successione dei valori calcolati, e si vede a occhio se il valore scappa subito, se gira attorno a un ciclo o se converge.

11. Quando qualcosa non funziona

La formula non compila

Il messaggio di errore compare in una finestra al momento di premere «Usa Formula». Le cause più frequenti sono poche:

Sintomo	Causa probabile
Errore di sintassi su una riga che sembra giusta	Manca il carattere due punti che chiude l'inizializzazione, oppure manca la riga del bailout.
Nome non riconosciuto	Una funzione scritta male o non supportata. Controlla l'elenco nell'Appendice A: i nomi non distinguono maiuscole e minuscole, ma devono essere esatti.
Errore su una parentesi	Parentesi non bilanciate, oppure barre verticali aperte e non chiuse.
La formula non compare nell'elenco	Il blocco non è stato riconosciuto: verifica che ci siano sia la graffa aperta sia quella chiusa, e che il nome non contenga spazi.

La formula compila ma l'immagine è tutta nera

Vuol dire che nessun punto sfugge: il bailout non viene mai raggiunto. Controlla che il ciclo modifichi davvero la variabile controllata dal bailout, e che la soglia non sia enorme.

L'immagine è tutta di un colore uniforme

Il caso opposto: tutti i punti sfuggono alla prima iterazione. Spesso è un'inizializzazione sbagliata, per esempio un valore di partenza già oltre la soglia.

L'immagine cambia aspetto mentre ingrandisci

È quasi sempre una dichiarazione di simmetria falsa. Il capitolo 8 spiega come riconoscerla e come disattivarla.

Il risultato non coincide con WinFract

Fractal10 riproduce fedelmente il comportamento di Fractint nella grande maggioranza dei casi, ma restano differenze note su formule che degenerano, per esempio quelle che dividono per zero o che crescono in modo esplosivo. La casella «Singolarità come inside (immagine pulita)» nella tab Formule permette di scegliere fra due comportamenti:

Stato	Comportamento
Disattiva (predefinito)	Il punto singolare viene considerato sfuggito, come fa Fractint. Fedele all'originale, ma può produrre zone dall'aspetto disturbato.
Attiva	La zona singolare viene considerata interna all'insieme. L'immagine risulta più pulita, ma non identica a quella di WinFract.

Vedere il codice generato

Il pulsante «Debug Code» mostra il codice che il programma ha ricavato dalla formula. È uno strumento di sviluppo, non serve per usare le formule: può tornare utile solo per segnalare un problema a chi sviluppa Fractal10.

12. IFS e L-System

Nella stessa tab convivono altri due formati, che non sono formule di iterazione ma descrivono il frattale in un modo diverso. Si aprono con i pulsanti «Apri .IFS...» e «Apri .L...».

IFS: sistemi di funzioni iterate

Un file .IFS contiene una tabella di trasformazioni geometriche, ognuna con una probabilità. Il disegno nasce applicando a caso una trasformazione dopo l'altra a un punto e segnando dove passa: dopo molte migliaia di passaggi la forma emerge. È il metodo con cui si ottiene la celebre felce di Barnsley.

Ogni insieme trovato nel file viene aggiunto all'elenco dei frattali con il prefisso ifs_.

L-System: regole di riscrittura

Un file .L descrive una curva con regole che sostituiscono simboli ad altri simboli, ripetutamente. Il risultato viene poi interpretato come una sequenza di movimenti di una penna: avanti, gira a destra, gira a sinistra. È il modo naturale di descrivere curve come il fiocco di neve di Koch o strutture ramificate.

Ogni sistema trovato viene aggiunto all'elenco dei frattali con il prefisso lsys_.

Nota — Per questi due formati i parametri p1-p5, il bailout e i metodi di colorazione non hanno effetto: la loro immagine non nasce da un'iterazione punto per punto.

Appendice A — Tutto quello che il linguaggio accetta

Questa tabella non è compilata a mano: ognuna delle 66 voci è verificata scrivendo una formula che la usa, compilandola ed eseguendola. Se una riga dice che qualcosa funziona, è perché è stato provato.

Alla data di generazione di questo manuale, 66 costrutti su 66 risultano funzionanti.

Struttura (5/5)

Elemento	Funziona	Descrizione
Formula minima	Sì	Le tre sezioni init, loop e bailout separate da due punti e virgola
Commenti con ;	Sì	Tutto quello che segue un punto e virgola viene ignorato
Più istruzioni su una riga	Sì	Istruzioni separate da virgola sulla stessa riga
Sezione init vuota	Sì	La formula può partire senza inizializzazioni
Direttiva periodicit�=0	Sì	Disattiva il controllo di periodicit�, necessario alle formule convergenti tipo Newton. Va scritta subito dopo il nome, prima della graffa: Nome[periodicity=0] {

Simmetria (4/4)

Elemento	Funziona	Descrizione
XAXIS	Sì	Simmetria rispetto all'asse reale
YAXIS	Sì	Simmetria rispetto all'asse immaginario
XYAXIS	Sì	Simmetria rispetto a entrambi gli assi
ORIGIN	Sì	Simmetria centrale rispetto all'origine

Variabili (11/11)

Elemento	Funziona	Descrizione
pixel	Sì	Coordinata del punto in esame, costante durante il ciclo
p1 e p2	Sì	I primi due parametri complessi impostabili dalla tab Parametri
da p3 a p5	Sì	Gli altri tre parametri complessi
P1 maiuscolo	Sì	I nomi dei parametri non distinguono maiuscole e minuscole
LastSqr	Sì	Modulo quadro dell'ultimo sqr() calcolato
maxit	Sì	Numero massimo di iterazioni impostato nell'interfaccia

e	Sì	Costante di Nepero
pi	Sì	Costante pi greco
scrnpix	Sì	Coordinate del pixel sullo schermo, in colonna e riga
whitesq	Sì	Vero sui pixel «bianchi» della scacchiera schermo
Variabili libere	Sì	Si possono definire variabili con nome a piacere

Operatori (6/6)

Elemento	Funziona	Descrizione
+ - * /	Sì	Le quattro operazioni fra numeri complessi
^ (potenza)	Sì	Elevamento a potenza, anche con esponente complesso
\ (divisione)	Sì	Barra rovesciata come sinonimo di divisione
 z modulo quadro	Sì	Le barre verticali danno il modulo AL QUADRATO, non il modulo
Confronti	Sì	Minore, maggiore, minore o uguale, maggiore o uguale, uguale, diverso
&& e 	Sì	Congiunzione e disgiunzione logica nelle condizioni

Controllo di flusso (4/4)

Elemento	Funziona	Descrizione
IF / ENDIF	Sì	Esecuzione condizionale semplice
IF / ELSE / ENDIF	Sì	Due rami alternativi
ELSEIF	Sì	Più condizioni in cascata
WHILE / ENDWHILE	Sì	Ciclo interno all'iterazione

Bailout (5/5)

Elemento	Funziona	Descrizione
 z <= valore	Sì	Condizione di fuga classica sul modulo quadro
LastSqr <= valore	Sì	Condizione basata sull'ultimo sqr()
abs(z) < valore	Sì	Condizione sul valore assoluto
Espressione con parametri	Sì	La soglia può essere calcolata
Condizione su real()	Sì	Il test può usare una componente

Funzioni (31/31)

Elemento	Funziona	Descrizione
<code>sin()</code>	Sì	Seno
<code>sen()</code>	Sì	Seno, sinonimo in italiano e spagnolo
<code>cos()</code>	Sì	Coseno
<code>tan()</code>	Sì	Tangente
<code>cot()</code>	Sì	Cotangente
<code>sinh()</code>	Sì	Seno iperbolico
<code>cosh()</code>	Sì	Coseno iperbolico
<code>tanh()</code>	Sì	Tangente iperbolica
<code>cotanh()</code>	Sì	Cotangente iperbolica
<code>asin()</code>	Sì	Arcoseno
<code>acos()</code>	Sì	Arcocoseno
<code>atan()</code>	Sì	Arcotangente
<code>exp()</code>	Sì	Esponenziale
<code>log()</code>	Sì	Logaritmo naturale
<code>sqrt()</code>	Sì	Radice quadrata
<code>raiz()</code>	Sì	Radice quadrata, sinonimo spagnolo
<code>sqr()</code>	Sì	Quadrato; aggiorna anche LastSqr
<code>abs()</code>	Sì	Valore assoluto
<code>cabs()</code>	Sì	Valore assoluto, sinonimo Fractint
<code>cosxx()</code>	Sì	Coseno nella variante storica di WinFract, con segno invertito
<code>trunc()</code>	Sì	Tronca alla parte intera
<code>floor()</code>	Sì	Arrotonda per difetto
<code>ceil()</code>	Sì	Arrotonda per eccesso
<code>round()</code>	Sì	Arrotonda all'intero più vicino
<code>ident()</code>	Sì	Identità: restituisce l'argomento invariato
<code>zero()</code>	Sì	Restituisce zero
<code>one()</code>	Sì	Restituisce uno
<code>flip()</code>	Sì	Scambia parte reale e immaginaria
<code>conj()</code>	Sì	Coniugato
<code>real()</code>	Sì	Parte reale
<code>imag()</code>	Sì	Parte immaginaria

Appendice B — Le formule incluse

Elenco delle formule presenti nella cartella formulas, raggruppate per file. Fra parentesi la simmetria dichiarata, dove c'è.

fractal10.frm (22)

MandelbrotTricorn, BurningShip, BurningShipJ, Celtic, Buffalo, Lambda3, SpiderWeb, Talis, Quartica, PerpendicularMandel, Phoenix, PhoenixP, PhoenixCube, Magnet1M, Magnet2M, Newton4Mod, Nova, CosMandel, ExpJulia, SinhShip, Barnsley1J, Barnsley1M

fractal10_extra.frm (18)

BirdOfPrey(), Cactus(), CactusJ(), CubicTricorn(), Glynn(), HalleyStar(), InvMandel(), ManowarJ(), ManowarM(), NewtonStar5(), NewtonStar7(), PerpShipJ(), QuarticTricorn(), Sharkfin(), StarJulia6(), StarLambda(), Tippetts(), InvMandel618()

fractal10_numeri.frm (15)

Alpha137(), GoldenAngle137(), Hepta7(), Hepta7Tricorn(), Prime11Julia(), Prime11Newton(), Stella3(), Tetra4Mandel(), Tetra4Newton(), Tri3Mandel(), Tri3Newton(), Stella4(), Stella5(), Stella8(), Stella10()

fractal10_teoremi_fix.frm (3)

DipoloFix(), GravitazioneFix(), TreCorpiFix()

In totale 58 formule installate.

Appendice C — Promemoria

Una pagina da tenere sotto mano mentre si scrive.

Struttura

```
Nome(SIMMETRIA)[direttive] {  
    inizializzazione, eseguita una volta :  
    ciclo, eseguito a ogni iterazione  
    condizione di continuazione  
}
```

Da ricordare

- $|z|$ è il modulo AL QUADRATO: il bailout classico è $|z| \leq 4$.
- I due punti chiudono l'inizializzazione; l'ultima riga è sempre il bailout.
- Il punto e virgola introduce un commento.
- Tutti i valori sono numeri complessi, anche quando ne scrivi uno solo.
- Nei confronti conta solo la parte reale.
- I nomi di funzioni e parametri non distinguono maiuscole e minuscole.
- Ogni IF va chiuso da un ENDIF, ogni WHILE da un ENDWHILE.
- La simmetria dichiarata non viene verificata: se è falsa, l'immagine è sbagliata.

Scheletri pronti

Insieme di Mandelbrot, cioè il parametro varia con il punto:

```
z = 0, c = pixel:
```

Insieme di Julia, cioè il parametro è fisso e il punto è il valore di partenza:

```
z = pixel, c = p1:
```

Formula convergente di tipo Newton, che richiede la direttiva:

```
Nome[periodicity=0] {  
    z = pixel:  
    z = z - (z^3 - 1)/(3*z^2)  
    |z^3 - 1| >= 0.0001  
}
```