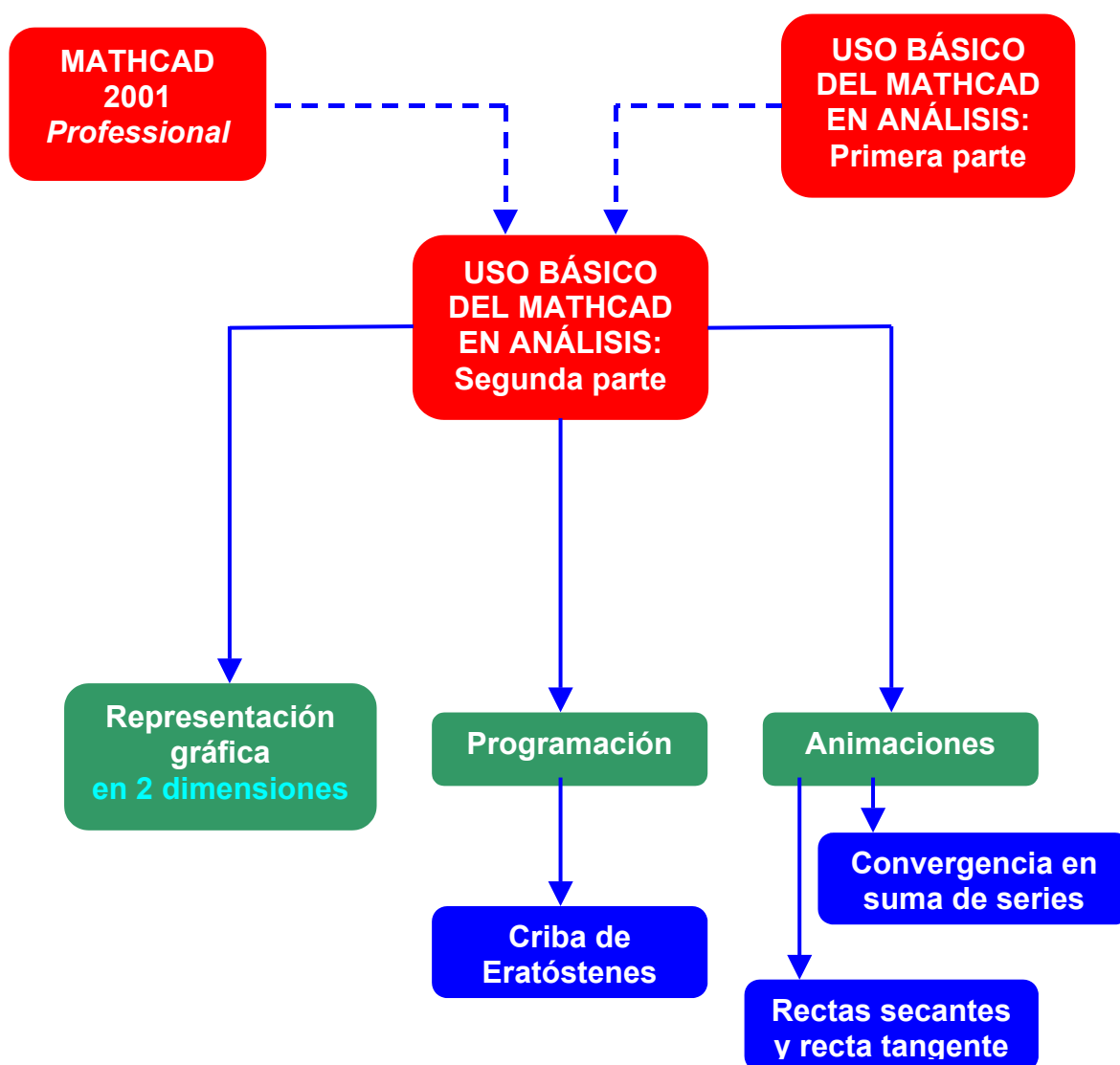


USO BÁSICO DEL MATHCAD EN ANÁLISIS (II): REPRESENTACIÓN EN TRES DIMENSIONES, PROGRAMACIÓN Y ANIMACIÓ

Autor: Patrici Molinàs Mata (pmolinas@uoc.edu), José Francisco Martínez Bosca (jmartinezbos@uoc.edu).

ESQUEMA DE CONTENIDOS



INTRODUCCIÓN

Hemos visto en el Mathblock “Uso básico del Mathcad en Análisis I: cálculo simbólico y numérico” la riqueza de posibilidades que ofrece Mathcad para efectuar operaciones de cálculo simbólico así como para resolver operaciones donde se persigue la obtención de un resultado numérico. Ahora queremos ilustrar las grandes posibilidades que tiene en otros tres ámbitos: la representación de funciones de dos variables, la programación y la generación de animaciones.

Las funciones de varias variables son difíciles de visualizar en el espacio tridimensional salvo en el caso de funciones de dos variables, que pueden ser representadas utilizando la tercera dimensión. Además de poder representar funciones en el espacio tridimensional, Mathcad también nos permite efectuar diversas operaciones como poderla girar en todos los ángulos y sentidos. Esto es relevante puesto que a menudo una propiedad de dicha función (como el límite en el origen: $(0,0)$) se observará mejor si podemos observar la función desde varios ángulos. Aparte del límite, la representación gráfica también permite entender otras propiedades de una función como son, por ejemplo, el gradiente y la derivada direccional.

Mathcad también representa una herramienta de programación de fácil utilización y totalmente incorporada al mismo entorno que permite el cálculo simbólico y numérico y la representación gráfica en una y dos dimensiones. Con Mathcad en general y, programando en particular, podemos generar secuencias de números (como sumas parciales de series) o objetos matemáticos (como rectas secantes) que son susceptibles de ser representados de forma encadenada constituyendo una animación. Veremos seguidamente dos ejemplos al respecto.

OBJETIVOS DOCENTES

- Describir la representación gráfica de funciones de dos variables, sus Proporciónar elementos suficientes para que el estudiante Introducir el concepto de función, proporcionar su representación tabular y gráfica. Saber determinar el dominio y el recorrido de una función cualquiera.
- Ilustrar las posibilidades de escribir pequeños programas para aplicar algún aspecto de la teoría con programas.
- Alcanzar un buen dominio de los elementos básicos necesarios para generar, activar y guardar una animación.

CONOCIMIENTOS PREVIOS

Es imprescindible —previamente a la lectura de este Mathblock— el haber desarrollado cierta destreza en el manejo del programa Mathcad. Para ello es fundamental trabajar el Mathblock “Uso básico de Mathcad” que encontraréis entre los Mathblocks de Álgebra. También se recomienda, aunque no es imprescindible, la lectura y asimilación de los contenidos del Mathblock: “Uso básico del Mathcad en Análisis I” puesto que en él se describen procedimientos y métodos de cálculo avanzado con dicho programa, de gran utilidad para la programación y la generación de animaciones.

Después de haber trabajado este Mathblock podéis trabajar el Mathblock “Funciones de varias variables” puesto que dominaréis la representación gráfica de funciones de dos variables. También sería conveniente regresar a los Mathblocks de “Derivación” y de “Series de números reales” puesto que los ejemplos aquí presentados hacen referencia a los conceptos que se trabajan allí.

CONCEPTOS FUNDAMENTALES

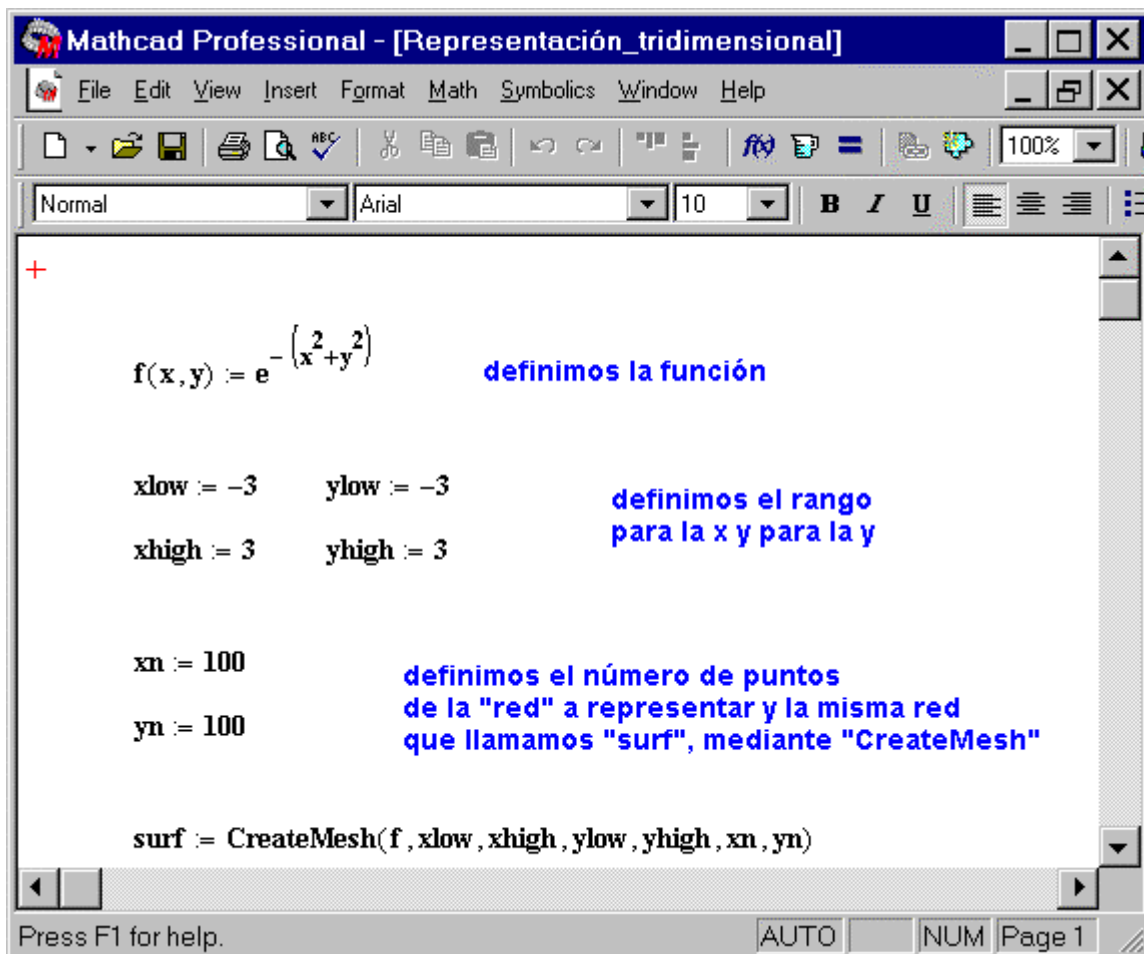
- **Representación gráfica de una función de dos variables. Límite en (0,0)**

Supongamos que tenemos una función de dos variables como, por ejemplo:

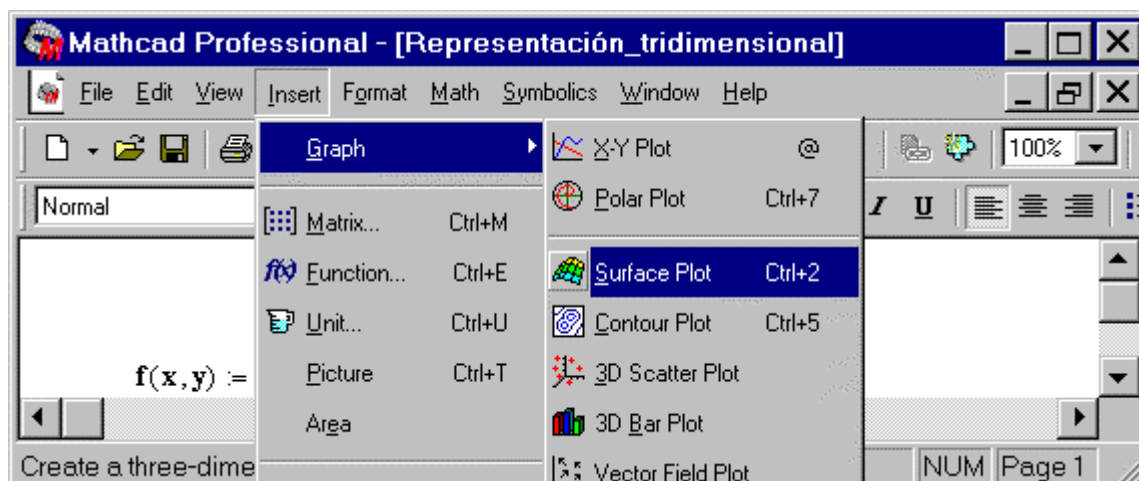
$$f(x, y) = e^{-(x^2+y^2)}$$

y deseamos conocer en qué punto la función es máxima o a qué valor tiende cuando una de las dos variables o ambas tienden a infinito. Si desconocemos como realizar los cálculos analíticos necesarios –que hemos descrito en “Funciones de varias variables I” y en “Funciones de varias variables II”– siempre nos queda el recurso de buscar el máximo numéricamente (véase “Uso básico del Mathcad en Análisis I”) y de representar la función para conocer su límite lejos del origen. De hecho, si hemos conseguido averiguar tanto el valor del máximo como el comportamiento asintótico de la función, también queremos comprobarlos y la gráfica de la función nos resultará de gran utilidad.

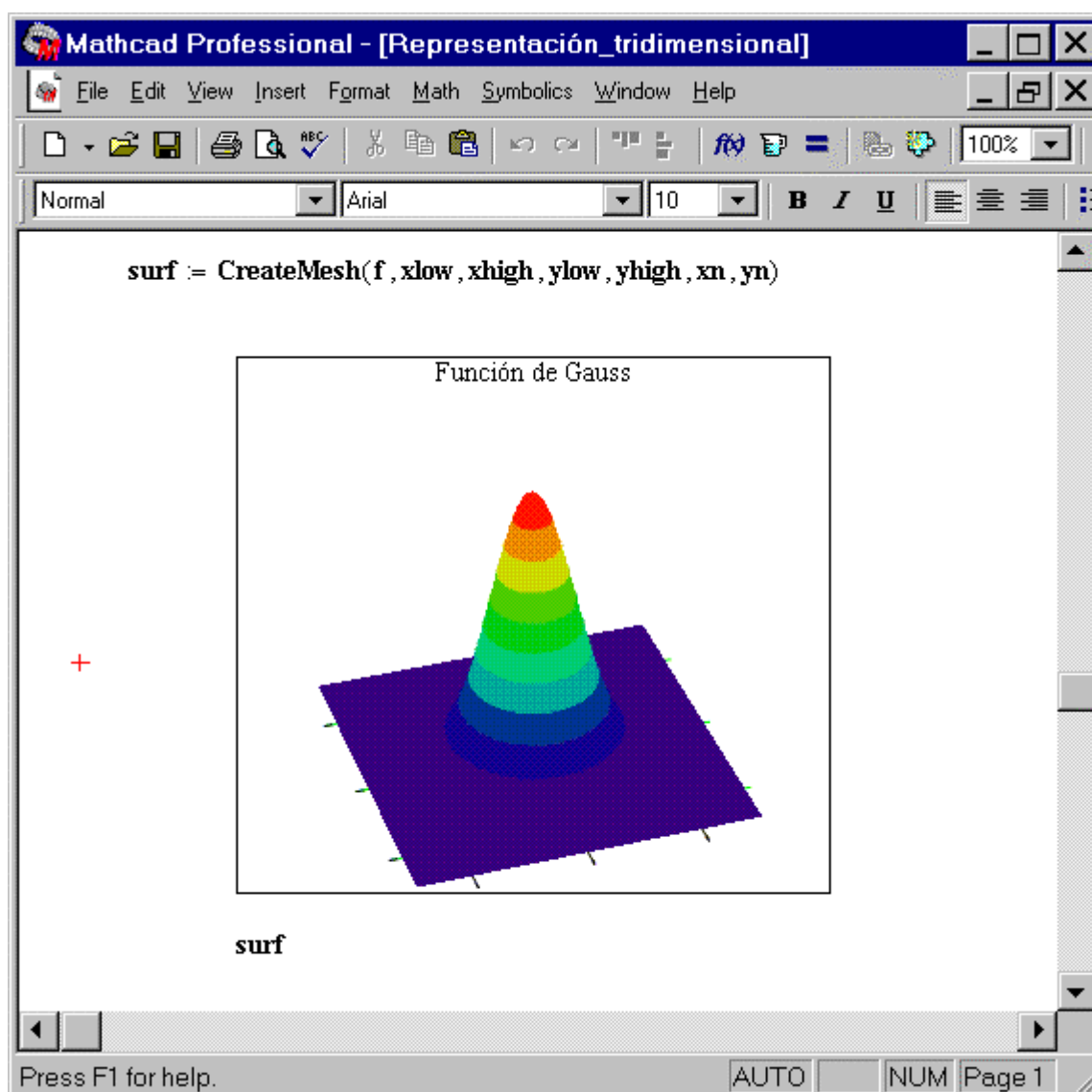
Vamos a generar una gráfica tridimensional de esta función de la siguiente manera. En primer lugar, definimos la función en una hoja de trabajo de Mathcad y seguidamente creamos una “red” de puntos en el espacio cuyas coordenadas z equivalen al valor de la función $z = f(x, y)$ en los puntos de coordenadas (x, y) escogidos. Si esta red es suficientemente fina (los puntos están suficientemente juntos) conseguiremos emular la superficie continua que representa una función tridimensional. Empecemos generando esta red de puntos:



Una vez calculados los valores que constituyen la superficie a representar, vamos a representarla mediante la instrucción *Surface Plot*:



Debemos introducir el nombre de la red de puntos a representar en la parte inferior del rectángulo para conseguir representar la gráfica tridimensional.

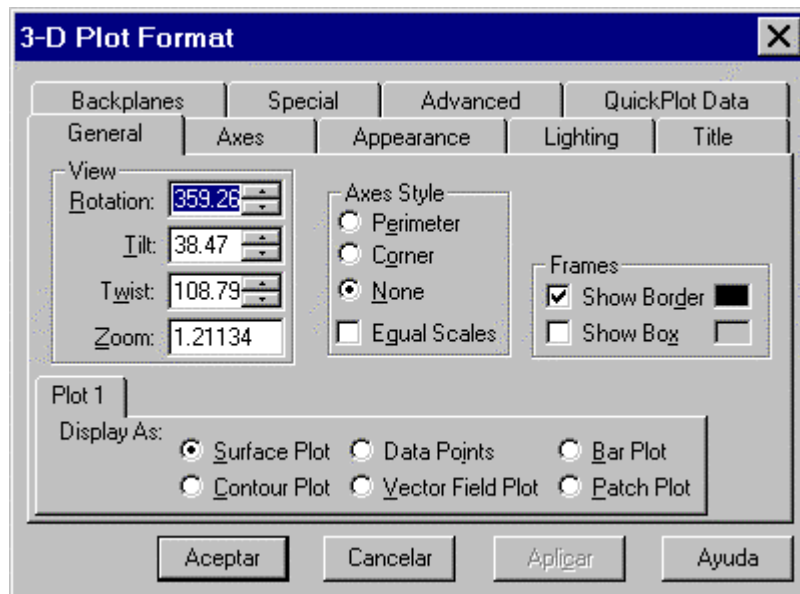


Si situamos el botón encima de la representación tridimensional y, con el botón izquierdo del mouse lo desplazamos, veremos rotar la figura alrededor de los tres grados de libertad angulares en el espacio.

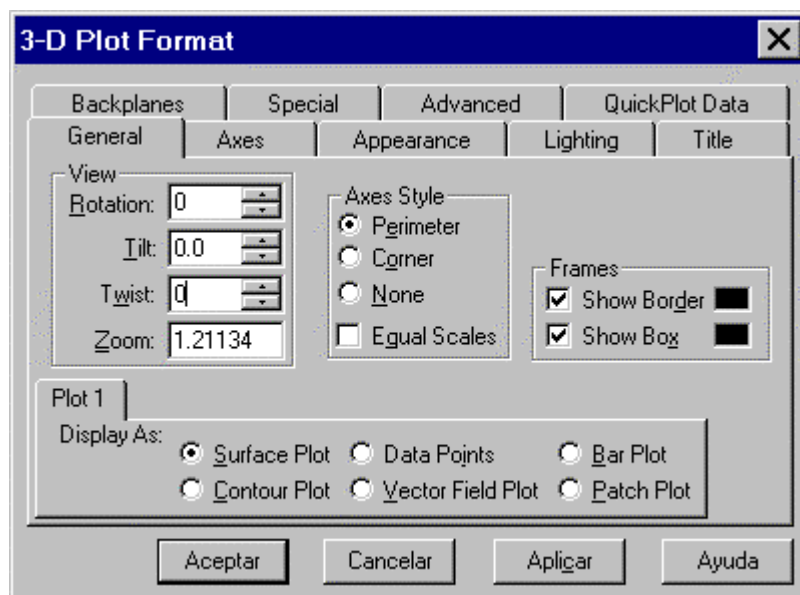
Seguidamente podemos modificar muchos aspectos de dicha representación. Basta con hacer doble click sobre la figura para que aparezca el menú "3-D Plot Format". En éste encontramos los botones o pastillas necesarios para modificar, por ejemplo:

- los ángulos de orientación en el espacio: *Rotation*, *Tilt* y *Twist*, el tipo de representación tridimensional: *Surface Plot*, *Data Points*, *Contour Plot*, etc, el tipo de ejes: *Axes Style* la presencia o no de una "caja" de ejes alrededor de la figura: *Show box*, etc. Utilizando estas opciones podemos llegar a representar la función de perfil y conocer su valor máximo. Todo esto modificando la pestaña General.

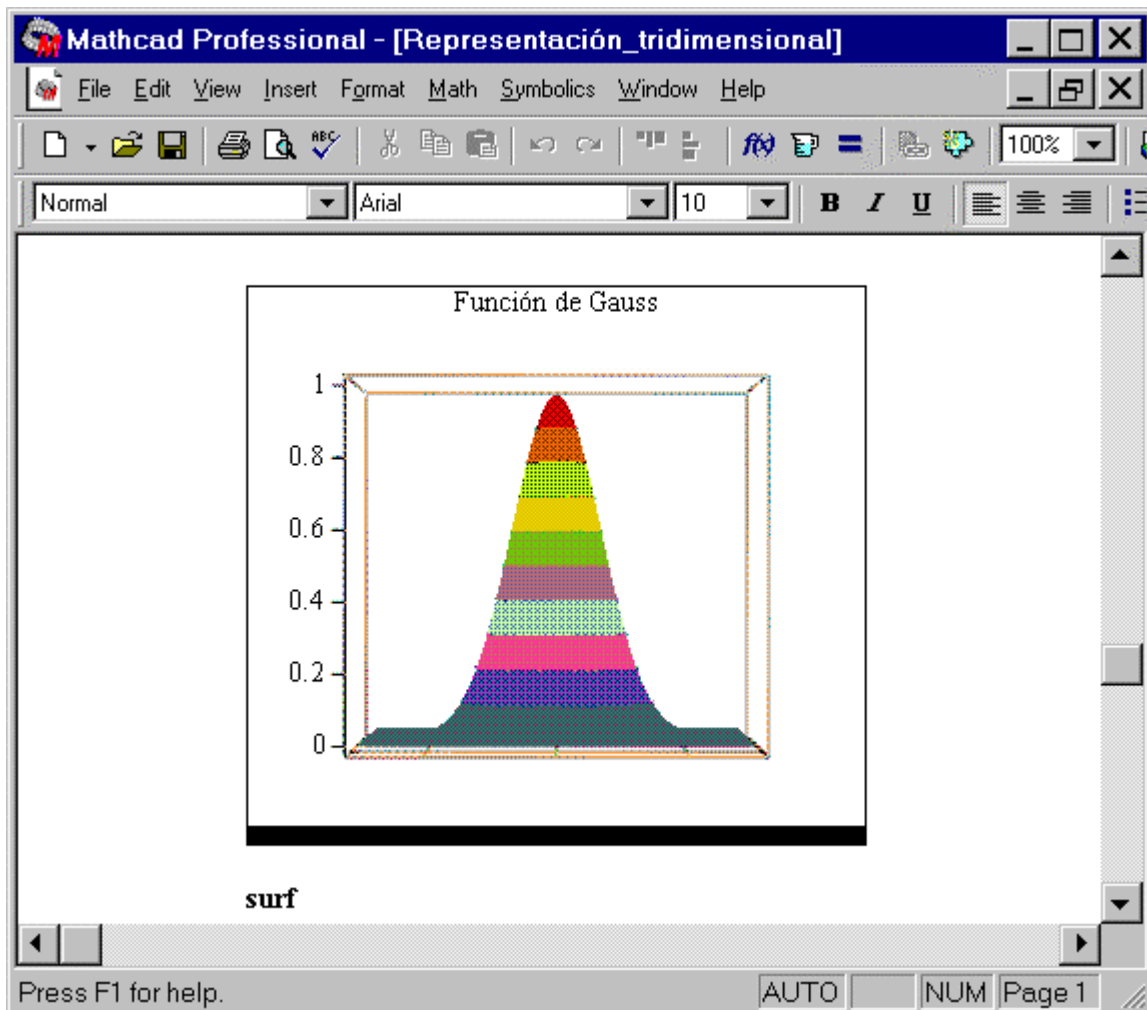
Así por ejemplo, si pasamos de esta configuración:



a esta otra:



conseguimos determinar que el máximo de esta función vale 1:



- **Programación. ¿Qué son los números primos y como obtenerlos?**

Supongamos que queremos general números primos, es decir, números que tan sólo son divisibles por ellos mismos o por la unidad. Podemos empezar el estudio mediante aquellos que conocemos de memoria:

1,2,3,5,7,11,13,17,...

Si queremos proporcionar más números primos debemos utilizar algún tipo de algoritmo. ¿Qué es un algoritmo? Un algoritmo es un conjunto específico de instrucciones para llevar a cabo un proceso o solucionar un problema, a menudo con la restricción que el proceso termine en algún momento. La palabra “algoritmo” es una distorsión de nombre del matemático persa al-Khwarizmi (approx. 780-850).

El matemático griego Eratóstenes (276-194aC) propuso el siguiente algoritmo para la obtención de todos los números primos más pequeños que el número natural N . Si un número

es múltiplo de otro, no puede ser primo. Entonces vamos a suprimir, de entre todos estos números, aquellos que sean múltiplos de los primeros primos, distintos de cero, que conocemos de memoria: 2,3,5,... hasta el valor del mayor primo, que sea menor que \sqrt{N} , aquellos números que no se vean suprimidos de la lista, serán primos. No es necesario seguir suprimiendo para múltiplos de números primos mayores que \sqrt{N} porque si un número es múltiplo de un primo mayor que \sqrt{N} también lo es de un primo menor que \sqrt{N} y, por lo tanto, ya habrá suprimido. Pongamos un ejemplo, si estamos suprimiendo múltiplos hasta $N = 40$, basta con suprimir los múltiplos de 2, 3 y 5 $< \sqrt{40} = 6,32...$ puesto que un múltiplo de 7 mayor que $\sqrt{40} = 6,32...$ como 21, ya habrá sido suprimido al haber suprimido los múltiplos de 3.

Por ejemplo, tomemos $N = 50$ y calculemos su raíz $\sqrt{N} = 7,071...$. Debemos ir suprimiendo múltiplos de primos distintos de cero hasta llegar al 7, es decir, debemos realizar cuatro operaciones: suprimir los múltiplos de 2, de 3, de 5 y de 7. Los números restantes (no suprimidos de la lista) serán los naturales. Este es el algoritmo que Eratóstenes inventó y que nosotros vamos a programar.

Suprimamos los múltiplos de 2 con un trazo azul:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

de 3 (que no sean múltiplos de 2) con un trazo rojo:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

de 5 (que no sean múltiplos ni de 2, ni de 3) con un trazo verde:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

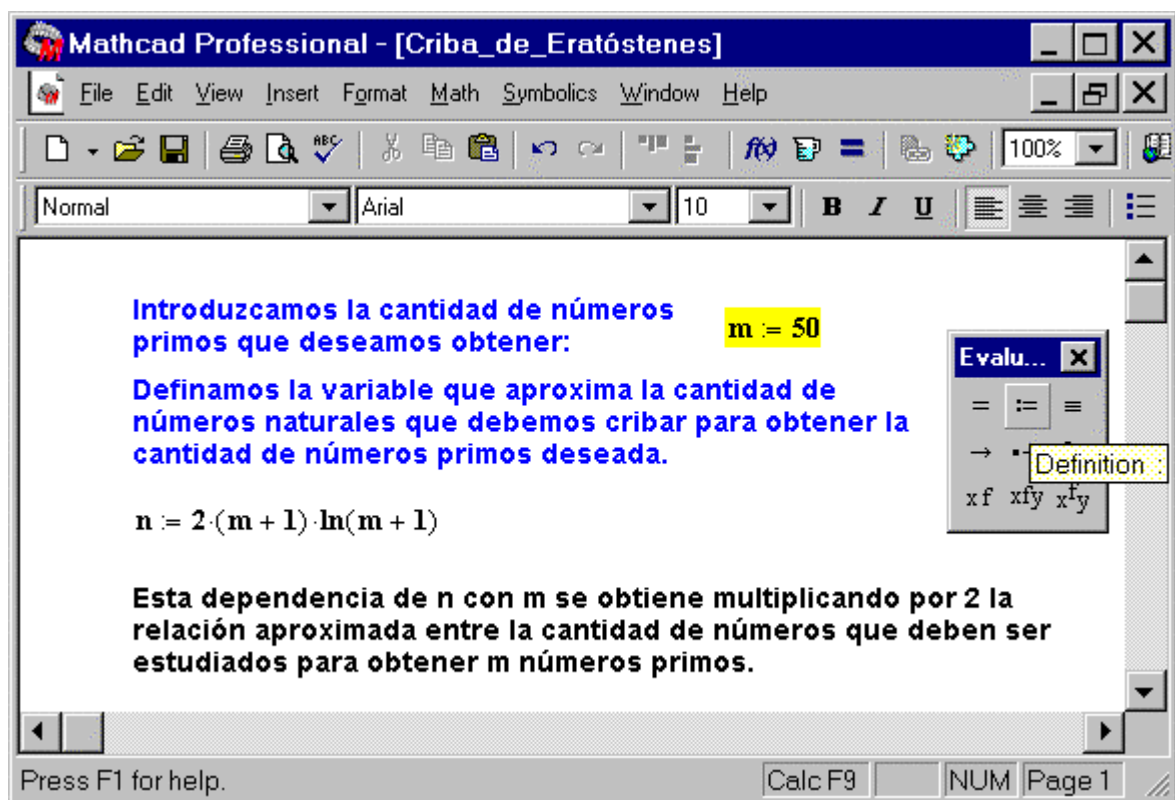
y de 7 (que no sean múltiplos ni de 2, ni de 3, ni de 5) con un trazo marrón:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

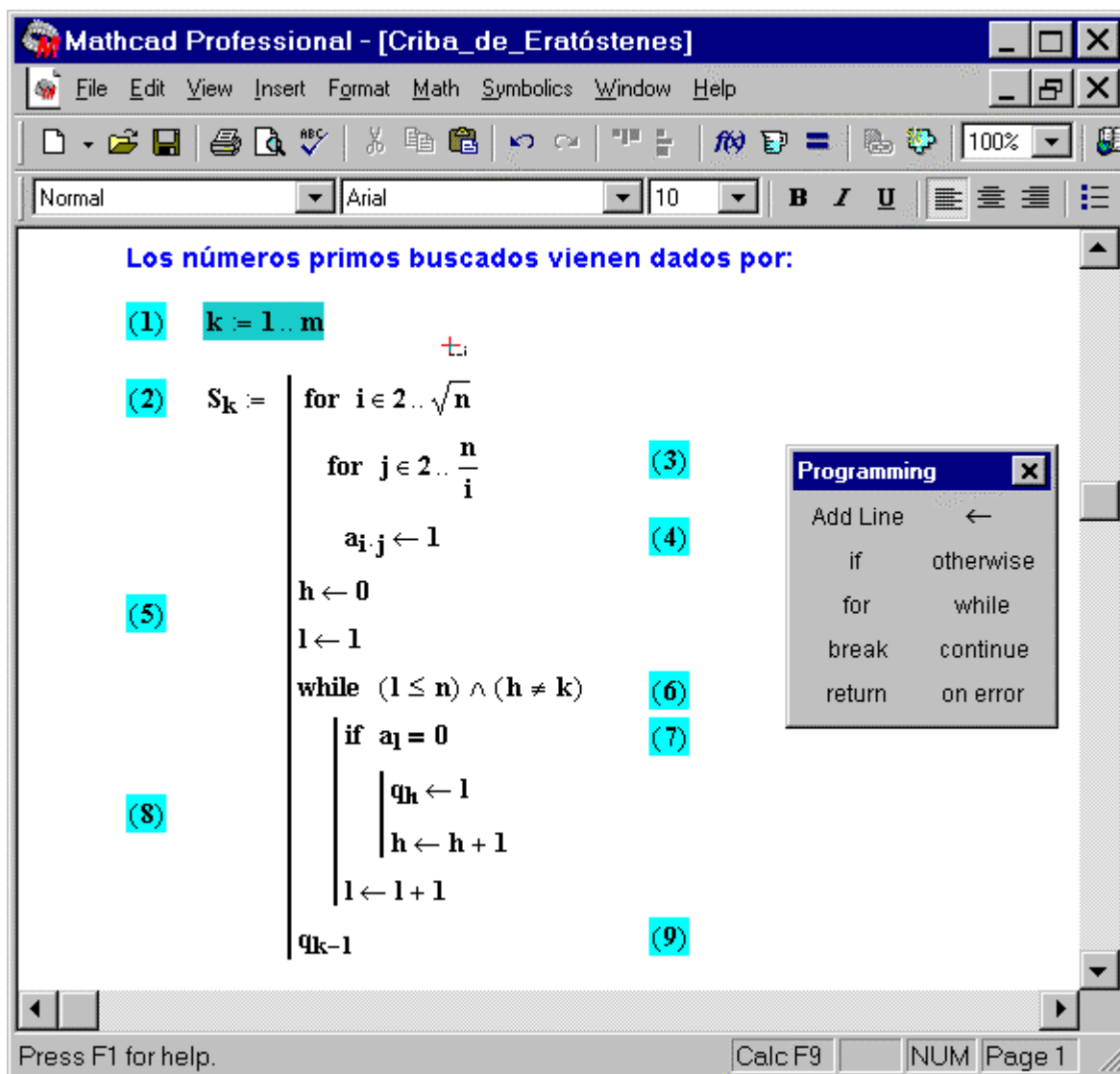
Así los números que no han sido suprimidos son los números primos entre 1 y 50, es decir:

1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43 y 47.

Veamos ahora como podemos programar este algoritmo con Mathcad. Empezamos por definir parámetros con la ayuda del modo "Definition" del menú "Evaluation", ("View">"Toolbars">"Evaluation"):



Luego entramos en la programación utilizando las funciones propias de ésta que encontramos en el menú "Programming".



Comentemos cada paso de programa:

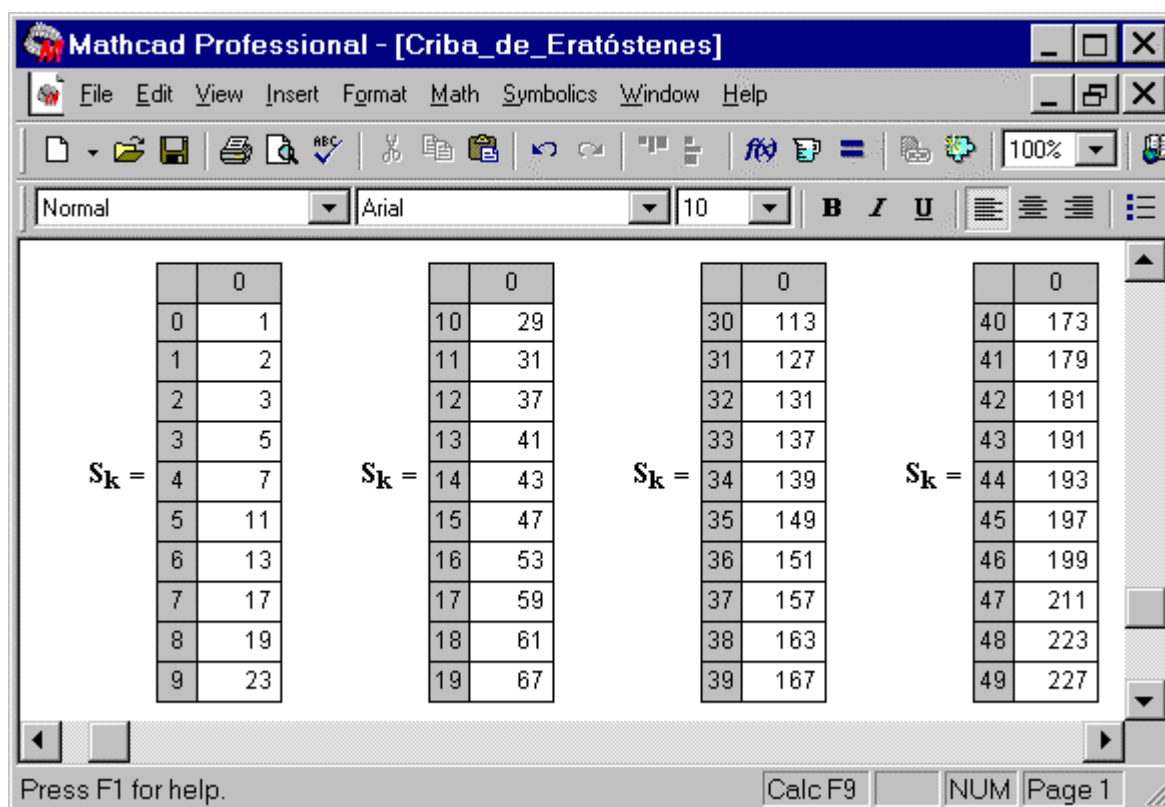
En (1) inicializamos el vector que contendrá los números primos.

En (2) definimos cada una de sus componentes mediante el programa.

Dentro de un doble b́ucle (3) que barre todos los números desde 1 hasta n , asignamos el valor 1 a los que son múltiplos (4).

Después de definir dos parámetros en (5) efectuamos una instrucción *while* ("mientras") que nos permite mediante la condición (7) ir reconociendo aquellos valores que son distintos de cero, es decir, cuyos índices corresponden a números primos que vamos asignando (8) a componentes del vector en (9), que es el resultado del programa.

Sólo nos queda ahora, presentar un subconjunto de los 50 primeros números primos:



- **Animación sobre un valor que converge. Suma de series infinitas**

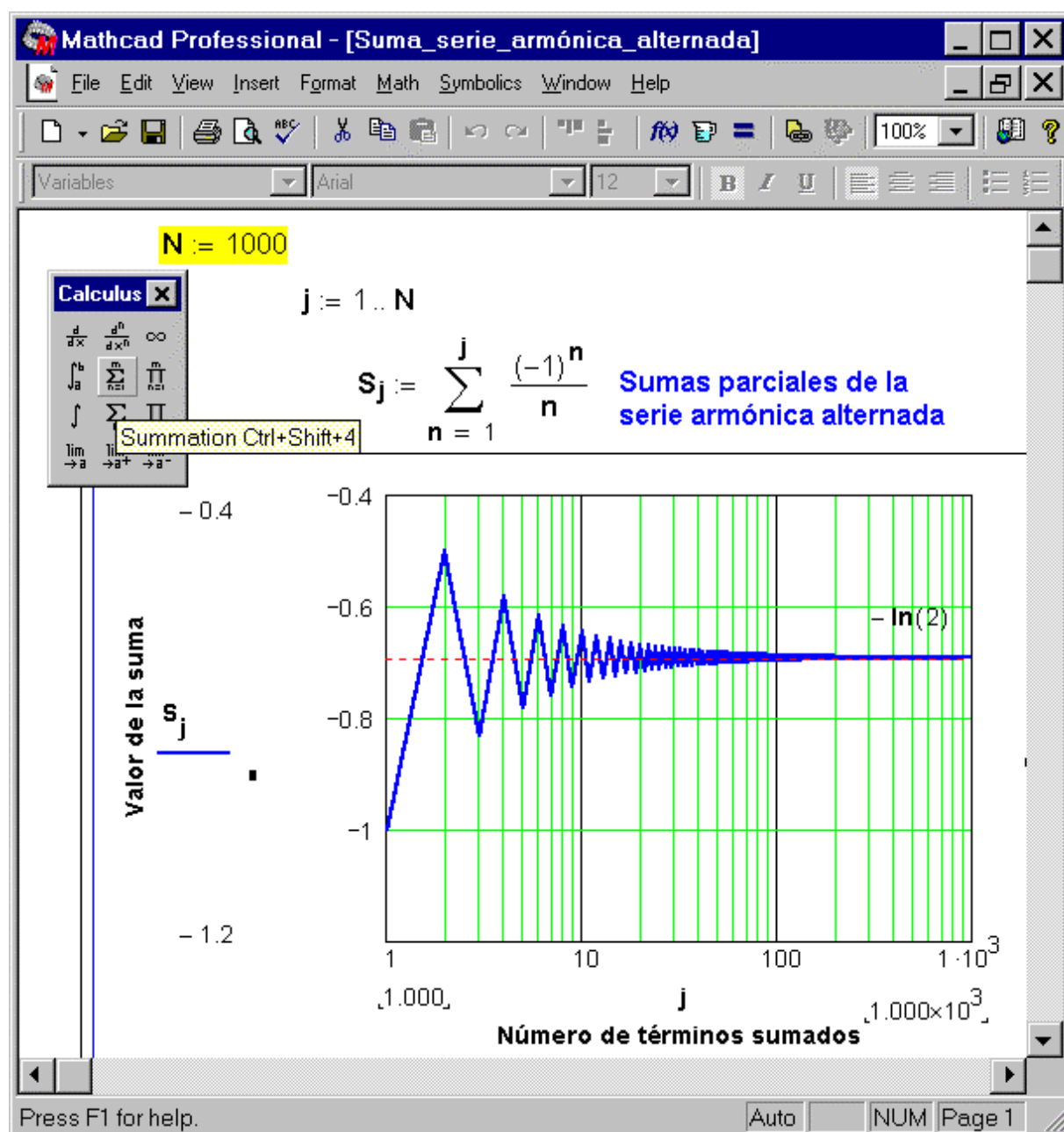
El concepto de convergencia se encuentra entre los que —a pesar de estar asociados a la variación de un solo número— mejor se prestan a ser visualizados en una animación. Vamos a calcular el valor numérico de la suma de la siguiente serie infinita:

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{n}$$

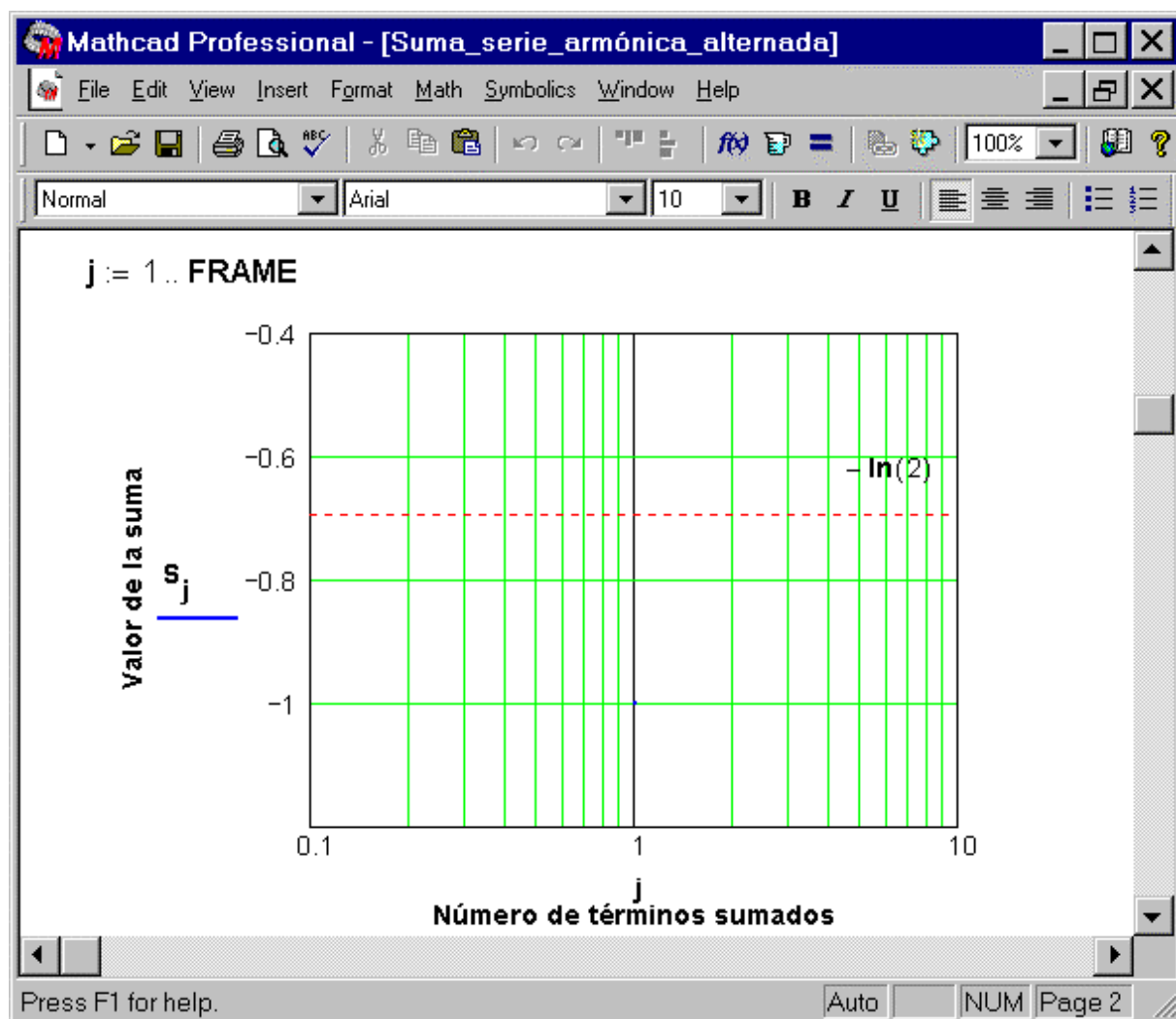
Para hacerlo, definiremos las sumas parciales de la serie de la siguiente forma:

$$S_j = \sum_{n=1}^j \frac{(-1)^n}{n}$$

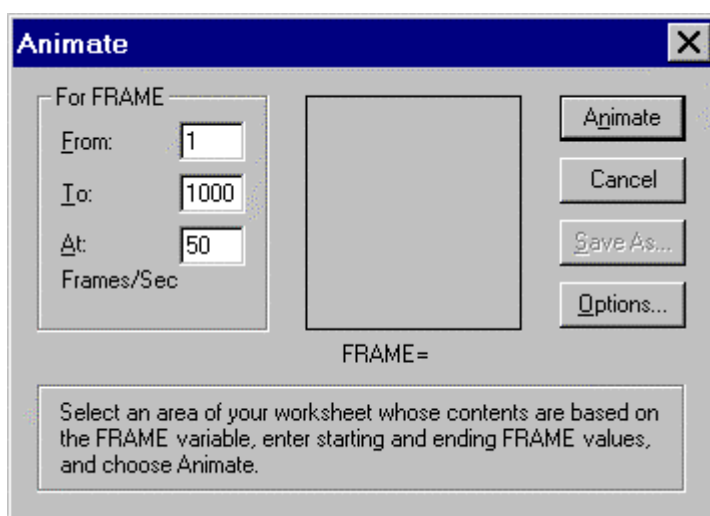
vamos a representar con Mathcad las primeras 100 sumas parciales de esta serie.



Queda clara la convergencia de la suma infinita al valor de $-\ln(2)$. La convergencia hacia este resultado se puede animar y hasta generar un fichero avi con la animación de la siguiente forma. En primer lugar definimos los valores a representar utilizando como valor superior a representar FRAME.



Seguidamente abrimos el menú *Animate* (*View>Animate*), sustituimos los valores mínimo (*From*) y máximo (*To*) y la frecuencia de barrido de las imágenes (*At*):



Finalmente basta con escoger la región que debe ser animada y proceder a lanzar la animación. Una vez finalizada ésta, se puede optar por salvarla en un fichero mediante el botón *Save as*. Esta animación, en particular, se encuentra en el fichero adjunto *convergencia.avi*:



convergencia.avi

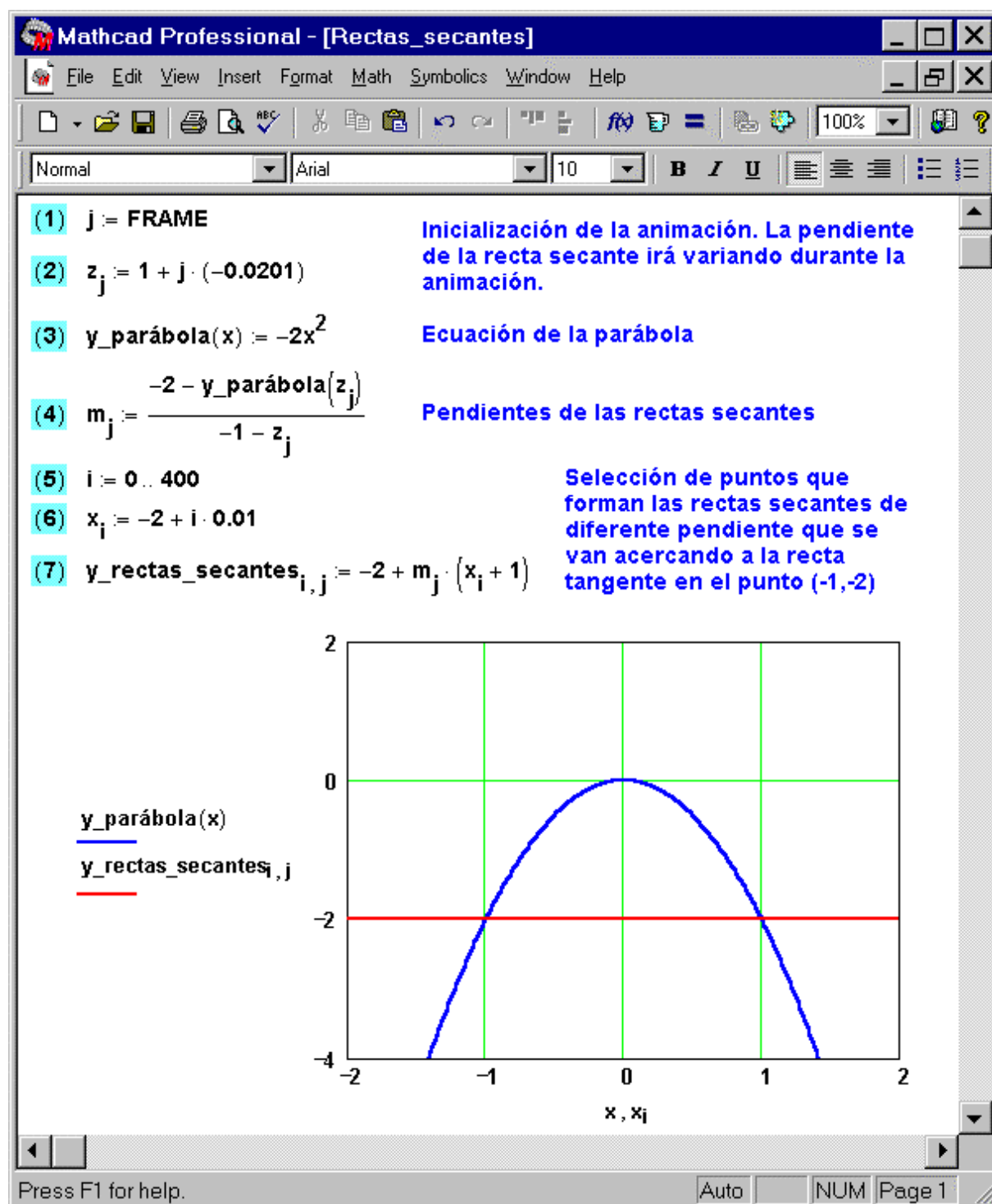
- **Animación sobre un objeto matemático. Rectas secantes que se convierten en la recta tangente**

Para finalizar este Mathblock vamos a generar la animación de un objeto matemático, la recta secante entre dos puntos de una parábola. Al aproximar el segundo punto hacia el primero hasta que coincidan, la recta secante se convertirá en tangente a la curva en el primer punto.

Veamos que hace el programa que reproducimos en la siguiente figura.

En primer lugar (1) se inicializa la variable discreta que va a permitir la animación mediante *FRAME*. Cada recta secante va a corresponder a un valor de la variable j . En efecto, determinamos el valor de la pendiente m de la recta que pasa por el punto $(-1, -2)$ y por otro punto de la parábola mediante la ecuación (4).

Luego construimos una matriz de puntos donde cada columna corresponde a las imágenes de una recta secante de pendiente diferente (7). Cada recta viene dada por 400 puntos (5) en el rango $[-2, 2]$.



Adjuntamos el fichero *rectas_secantes.avi* generado:



rectas_secantes.avi

BIBLIOGRAFÍA

- [1] H. Benker (Translated A.Rudd) (2000): "Practical use of Mathcad. Solving mathematical problems with a computer algebra system", Springer Verlag, New York, 504pp.
- [2] Ph.J. Pritchard, (1998): "Mathcad: a tool for engineering problem solving. B.E.S.T. Series", McGraw-Hill, Boston, 338pp.
- [3] R.W. Larsen (2001): "Introduction to MathCAD 2000", Prentice Hall, Upper Saddle River, NJ, 250pp.
- [4] J. Rowell (1993): "Mathcad Education Library: Calculus", Mathsoft, Cambridge, MA.
- [5] D. Kyrianov (2002): "The Mathcad 2001i Handbook", Charles River Media, Hingham, MA, 574pp.
- [6] K.A. Ansari (1999): "Numerical Methods for Engineers with Mathcad", Ulyssian Publications, Spokane, WA, 360pp.
- [7] S.C. Chapra and R.P. Canale (2002): "Numerical Methods for Engineers with Programming and Software Applications", 3rd edition, McGraw-Hill, New York.
- [8] MathSoft Engineering & Education (2001): "Mathcad: user's guide with reference manual", MathSoft Engineering & Education, Cambridge, MA.
- [9] MathSoft, Inc (traducción de J. A. Moreno y D. Ser) (1999): "Mathcad 8. Manual de usuario y guía de referencia de Mathcad 8", ediciones Anaya Multimedia, S.A., Madrid.
- [10] B. Birkenland (1997): "Mathematics with Mathcad", Studentlitteratur, Lund, Suecia.

ENLACES

- [W1] <http://www.mathsoft.com/>
Corporación Mathsoft que produce el programa Mathcad.
- [W2] <http://www.addlink.es/>
Distribuidor oficial del programa Mathcad en España.
- [W3] <http://ist.uwaterloo.ca/ic/mathcad/>
En la Universidad de Waterloo hay un importante esfuerzo en la enseñanza de las matemáticas y disciplinas cuantitativas con software, en particular con Mathcad. Son muy instructivas las animaciones que se presentan para entender el funcionamiento del programa.
- [W4] <http://www.math.odu.edu/cbii/calcanim>
Animaciones para el cálculo.
- [W5] <http://www2.latech.edu/~schroder/mathcd.html>
Relación de archivos interesantes sobre cálculo con el Mathcad.
- [W6] <http://courses.lugo.com/mcad.htm>
Relación de archivos interesantes sobre cálculo con el Mathcad.
- [W7] <http://www.softwarecientifico.com/paginas/mathcad.html>
Distribuidora Software Científico en que se explica en qué consiste Mathcad y lo que éste ofrece.