Aplicaciones de GeoGebra al Análisis

Raúl Manuel Falcón Ganfornina[1] rafalgan@us.es
Ricardo Ríos Collantes de Terán[2] profesofricardo@yahoo.es
Eva Barrena Algara[3] ebarrena@us.es
Rosana Ramírez Campos[4] roracam@gmail.com

Resumen

En el presente trabajo se elaboran actividades con GeoGebra para utilizarlas en el aula de Matemáticas en el Bloque de Funciones y Gráficas de la ESO y en el bloque de Análisis en Bachillerato. Para ello se hace uso de las diferentes zonas de la ventana de trabajo del programa explorando, entre otras cosas, las posibilidades de la hoja de cálculo para crear una base de datos de funciones, estudiar su composición o crear una tabla de valores de las mismas, así como para resolver problemas de optimización.

1. Introducción.

En GeoGebra, sólo se pueden introducir funciones de variable ***x***. Algunas predefinidas son: Raíz cuadrada (***sqrt(x)***), exponencial (***exp(x), e^x^***), logaritmos (***ld(x), lg(x), ln(x)***), valor absoluto (***abs(x)***), trigonométricas (***sin(x), cos(x), tan(x), acos(x), asin(x), atan(x)***) y parte entera (***floor(x)***). En caso de querer introducir una función no predefinida, esto puede hacerse directamente en la **barra de entrada** (ej.: $f(x)=x^2$), dibujándose automáticamente su gráfica en todo su dominio. Si se quiere obtener dicha gráfica en un intervalo concreto, se usaría el comando: **Función[<Función>, <Valor inicial>, <Valor final>]**. Dada una función ***f(x)***, se define su función derivada como ***f'(x)*** ó ***Derivada[f]***, su integral indefinida como ***Integral[f]*** y su integral definida entre ***a*** y ***b*** (dibuja región) como ***Integral[f,a,b]***. Finalmente, dado un punto ***P = (a,b)***, se pueden obtener sus coordenadas ***a*** y ***b***, como ***x(P)*** e ***y(P).***

2. Base de datos de funciones.

A la hora de crear una actividad en GeoGebra es interesante disponer de una plantilla que permita generar actividades similares. La posibilidad de obtener números aleatorios facilita este hecho, al poder definir una base de datos de funciones, habilitando la hoja de cálculo. Desplazando las casillas hasta la columna **Z** en las celdas **Y1, Y2, Y3** e **Y4** se escribe, respectivamente **"Recta", Parábola", "Racional1"** y **"Racional2"** y, en las celdas **Z1, Z2, Z3** y **Z4**: **AleatorioEntre[-10, 10] x + AleatorioEntre[-10, 10]**, **AleatorioEntre[-10, 10] x² + AleatorioEntre[-10, 10] x + AleatorioEntre[-10, 10]**, **AleatorioEntre[-10, 10] / (x-AleatorioEntre[-10, 10])** y **AleatorioEntre[-10, 10] / ((x-AleatorioEntre[-10, 10]) (x-AleatorioEntre[-10, 10])**).

---

[1] ETS de Ingeniería de Edificación. Dpto. de Matemática Aplicada I. Universidad de Sevilla.
[2] I.E.S. Sofía. Jerez de la Frontera (Cádiz).
[3] ETS de Ingenieros. Dpto. de Matemática Aplicada II. Universidad de Sevilla.
[4] I.E.S. Manuel de Góngora. Tabernas (Almería).

II Jornadas de GeoGebra de Andalucía (2011), 1-5. ISBN: 978-84-937577-7-9.

Ocultando **Z1(x), Z2(x), Z3(x)** y **Z4(x)**, se seleccionan las celdas **Z1** a **Z4** y se crea una lista asociada **L_1**. En la ventana de propiedades, se marcan como auxiliares las funciones y la lista. Finalmente, basta crear una función que cambie de forma aleatoria al pulsar **F9**: **f(x)=Elemento[L_1, AleatorioEntre[1, Longitud[L_1]]]**[5].

3. Tabla de valores.

GeoGebra posibilita crear ejercicios relacionados con tablas de valores haciendo uso de su **hoja de cálculo**, de tal forma que el alumnado pueda completar una tabla de valores asociada a una función dada y el programa indique si la respuesta es correcta. Para ello, se introduce en la **barra de entrada** la función a estudiar, por ejemplo: **f(x)=x^2**. En la **hoja de cálculo**, se introduce en las celdas **A1**, **B1** y **C1** respectivamente: **"x"**, **"f(x)"** y ***f(x)***. En las columnas **A** y **B** se incluyen los datos que se facilitarán al alumnado, por ejemplo: **A2=1, A5=12, B3=4, B4=25**. Tras introducir en la celda **C2** el comando **Si[f(A2) = = B2, "Correcto", "Incorrecto"]**, se marca **C2** con el ratón y se arrastra el control de relleno hasta la última fila de datos. Finalmente, basta eliminar los ceros que GeoGebra genera de forma automática en los huecos de la tabla[6].

4. Curva que pasa por *n* puntos.

Unos de los problemas que nos encontramos con GeoGebra es que no se puede dibujar una función que pase por *n* puntos **$A_1,\ldots,A_n$**, salvo que ésta sea una función lineal. Resultaría pues interesante crear una curva a trozos entre cada par de puntos **A_i** y **A_{i+1}**. Para que dicha curva sea *suave* es necesario que las derivadas laterales de cada trozo coincidan. Se trata pues de un polinomio de grado tres, $f(x) = ax^3+bx^2+cx+d$, cuyos coeficientes se obtendrán al resolver un sistema de cuatro ecuaciones con cuatro incógnitas. En concreto, dados cuatro puntos **A, B, C** y **D**, se puede crear una herramienta que construya el polinomio de grado tres que pase por **B** y **C** y cuyas tangentes en estos puntos sean paralelas a **AC** y **BD**, respectivamente (Fig. 1).

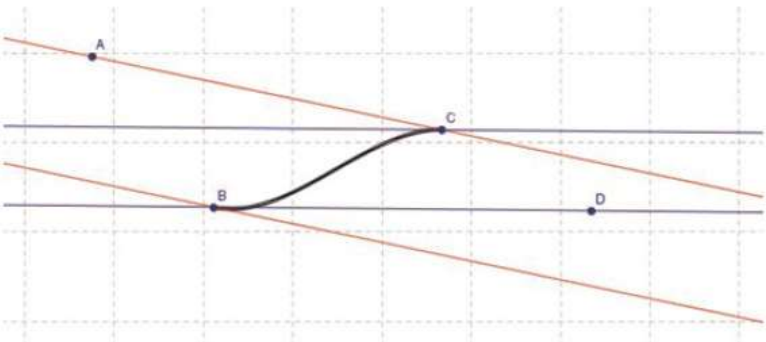

Figura 1: Generación de curva que pasa por dos puntos.

Para ello, en primer lugar se definen las dos pendientes de las rectas **AC** y **BD**: **m_1 = (y(C)-y(A)) / (x(C)-x(A))** y **m_2 = (y(D)-y(B)) / (x(D)-x(B))**. El sistema a resolver es entonces {**ax(B)^3 + bx(B)^2 + cx(B) + d = y(B), ax(C)^3 + bx(C)^2 + cx(C) + d = y(C), 3ax(B)^2 + 2b x(B) + c = m_1, 3ax(C)^2 + 2b x(C) + c = m_2**. Dicho sistema se resuelve multiplicando la matriz de coeficientes por el vector de términos independientes: **S =**

---

5 http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/BD.html

6 http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/TV.html

MatrizInversa[{{x(B)^3, x(B)^2, x(B), 1}, {x(C)^3, x(C)^2, x(C), 1}, {3x(B)^2, 2x(B), 1, 0}, {3x(C)^2, 2x(C), 1, 0}}] {{y(B)}, {y(C)}, {m$_1$}, {m$_2$}}. Se define entonces la función, haciéndola cero fuera del intervalo **(x(B),x(C)): p(x) = Si[x < x(C), Si[x > x(B), Elemento[Elemento[S, 1], 1] x^3 + Elemento[Elemento[S, 2], 1] x^2 + Elemento[Elemento[S, 3], 1] x + Elemento[Elemento[S, 4], 1], 0], 0]**. Es posible crear ahora una nueva herramienta, tal que, en **Objetos de Salida** seleccionamos la función **p(x)** y, en **Objetos de Entrada**, los cuatros puntos ordenados. Ya se está en disposición de crear una función que pasa por un determinado número de puntos, por ejemplo por 6 puntos: **A_1**, **A_2**, **A_3**, **A_4**, **A_5** y **A_6**. Basta crear los *trozos* uno a uno y después sumarlos. Para solventar el problema de los puntos extremos se crean dos puntos nuevos, por ejemplo, los simétricos con respecto al extremo del punto anterior o posterior, según corresponda: **$A_0 = 2\ A_1 - A_2$, $A_7 = 2\ A_6 - A_5$**. Se utilizaría entonces la herramienta creada para generar cinco funciones, **p_i**, a partir de los puntos **A_{i-1}**, **A_i**, **A_{i+1}** y **A_{i+2}**, para i=1, 2, 3, 4 y 5. Bastaría generar una nueva herramienta a partir de la función: **f(x)=Función[p_1(x) + p_2(x) + p_3(x) + p_4(x) + p_5(x), x(A_1), x(A_6)]**[7].

5. Coordenadas cartesianas.

El clásico juego "Batalla Naval" puede resultar de gran utilidad al comenzar a tratar el concepto de coordenada cartesiana. GeoGebra permite diseñar una actividad basada en dicho juego. Para ello, conviene visualizar una **Cuadrícula de Distancia 1-1**, haciendo uso de la **Vista Gráfica**. Se crea entonces como objeto auxiliar una lista **Flota** con **17** puntos correspondientes a los cinco barcos de la flota: Portaaviones (5 puntos), acorazado, (4 puntos), crucero (3 puntos), submarino (3 puntos) y destructor (2 puntos). Por ejemplo, **Flota={(1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (4, 6), (4, 7), (4, 3), (5, 3), (6, 3), (7, 3), (1, 4), (1, 5), (1, 6), (6, 6), (7, 6), (8, 6)}**. A continuación se crea el punto **A = (0,0)** asociado al disparo y se oculta el mismo. Para conocer si el disparo da en el blanco o no, se escribe en la **barra de entrada**: **Acierto = Suma[Secuencia[Si[Elemento[Flota, i] = = A, 1, 0], i, 1, 17]].** Seguidamente se crea un cuadrado que marque sobre la cuadrícula el efecto producido al efectuar el disparo: **Polígono[A+(0.43,0.43), A+(-0.43,0.43), A+(-0.43,-0.43), A+(0.43,-0.43)]}.** Se oculta los rótulos de los segmentos, se activa sus rastros y se maximizan sus grosores. Más tarde, en las **propiedades avanzadas** de los objetos Cuadrilátero y Segmento, se escribe en **Colores dinámicos**: Rojo: **Acierto = = 1**, Azul: **Acierto = = 0**. Colocando como auxiliar todos los elementos salvo **A**, basta cambiar las coordenadas de dicho punto para empezar a jugar[8].

6. Proporcionalidad directa / inversa.

El concepto de proporcionalidad puede tratarse a partir del análisis de la ecuación **e = v · t**, asociado al desplazamiento de dos ruedas de sendos vehículos que se desplazan a distintas velocidades. Para ello, se crean un deslizador **t** asociado al tiempo, definido en el intervalo **[0,20]**, y un deslizador **v** asociado a la velocidad, definido en **[0,2]**. Supondremos que el primer vehículo circula siempre a velocidad unidad, mientras que la velocidad del segundo vehículo será **v**. Se dibujan las gráficas de las funciones distancias, **f(x) = x** y **g(x) = v x**, y se define un punto sobre cada una de ellas, dependiendo de **t**: **P = (t,f(t))** y **Q = (t,g(t))**. Se crean las vías por las que circularán los

---

[7] http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/Gp.html

[8] http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/BN.html

vehículos a estudiar, **y= -6** e **y = -10** y se define, en función del tiempo, la posición de las ruedas de dichos vehículos: **A = (t,-10), B = (t,-9), c = Circunferencia[B, A], C = (v t,-6), D = (v t,-5)** y **d = Circunferencia[D, C]**. Para dar sensación de movimiento cíclico, se hace uso de la función cicloide (ocultando las curvas tras definirlas): **a = Curva[t + sin(t), cos(t) - 9, t, 0, 40]** y **b = Curva[t + sin(t), cos(t) - 5, t, 0, 80]**. Se define un punto en cada una de las cicloides, **E = a(t)** y **F = b(v t)**. Basta seleccionar **animación automática** en el deslizador tiempo y mover el deslizador velocidad para analizar la proporcionalidad entre espacio, velocidad y tiempo[9].

7. Derivadas y optimización.

GeoGebra aporta un punto visual a problemas de optimización habitualmente tratados únicamente de forma analítica, como por ejemplo puede ser maximizar el área de un cuadrilátero inscrito en una circunferencia dada de radio 1. Para tratar esta actividad se crea un deslizador **t** en el intervalo **[-1,1]**. Se dibuja una circunferencia centrada en el origen de coordenadas y de radio 1. En la **barra de entrada** se definen los vértices del cuadrilátero: **A=(t, sqrt(1 – t^2)), B=(-t, sqrt(1 - t^2)), C=(-t, -sqrt(1 - t^2))** y **D=(t, -sqrt(1 - t^2))**. Se crea el polígono **ABCD** y se halla su área con el comando **área**. Hallando el lado del cuadrilátero mediante el comando **distancia** puede observarse cómo, al mover el deslizador, varía el cuadrilátero inscrito en la circunferencia y consecuentemente, su área, pudiendo fijar el instante donde ésta es máxima. Esta primera aproximación puede ajustarse a partir del cálculo de derivadas. En concreto, se teclea en el campo de entrada la función **área(x) = abs(4x sqrt(1-x^2))** y se crea un punto **P=(t, área(t))** sobre la gráfica de la función. Activando **Tangentes** y pinchando sobre **P** y sobre la gráfica de la función **área**, puede obtenerse una línea **e** tangente a la función **área** en **P**. Se puede hacer uso del comando **m=Pendiente[e]** para obtener un triángulo ilustrativo de la pendiente de la tangente. La traza del punto **Z = (x(P), m)** determina la función derivada del área, que puede obtenerse tecleando el comando **Derivada[área]**. Basta mover el deslizador, para observar que el máximo y mínimo área se obtienen en los puntos donde la pendiente se anula[10].

8. Programación lineal.

La programación lineal es una de las aplicaciones más inmediatas y visuales del estudio de las inecuaciones. Las técnicas y procedimientos que involucra son sencillos a pesar de la importancia de los resultados que proporcionan. Como ejemplo a desarrollar planteamos el siguiente problema: *"En un comedor se desea diseñar un menú con más de 2000 calorías, 60 g. de proteínas y 80 g. de grasas. Para ello se dispone de dos platos tales que 100 g. del 1º contienen 250 calorías, 10 g. de proteínas y 15 g. de grasas, mientras que 100 g. del 2º contienen 800 calorías, 15 g. de proteínas y 20 g. de grasas. Si el precio de 100 g. del 2º plato es doble del de 100 g. del 1º, hallar cuántos gramos se deben servir de cada plato para minimizar el coste"*. En GeoGebra, se insertan las tres ecuaciones que corresponden con las inecuaciones necesarias para resolver el problema, se calculan todos los puntos de corte necesarios y se obtiene la región factible. Para ello se coloca un punto libre **A** y en la **barra de entrada** se escribe la condición **d: 5 x(A)+16 y(A)>40 ∧ 2 x(A)+3 y(A)>12 ∧ 3 x(A)+4 y(A)>16 ∧ x(A)>0**

9 http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/Velocidad.html

10 http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/Derivada.html

∧ y(A)>0. En las **propiedades avanzadas** del punto **A** dentro de **Colores Dinámicos** escribimos: Rojo: **Si[d,1,0],** Verde: **0,** Azul: **250**. Activando el rastro del punto **A**, éste será rojo dentro de la región factible y negro fuera de ella. En la **hoja de cálculo** se introducen por filas los vértices de la región factible. En la Columna **C** se incluye la función objetivo, de tal forma que, arrastrando el control de relleno, se puede obtener el precio en cada combinación posible, eligiendo aquélla en la que el coste sea mínimo[11].

9. Operaciones entre funciones: Diseño gráfico.

Los colores dinámicos de GeoGebra permiten obtener distintos tipos de diseños gráficos[12]. Usando esta idea, se puede observar cómo influye la composición de funciones en la creación de estos gráficos (Fig. 2). Como ejemplo, en caso de querer trabajar en la región *[-10,10] x [-10,10]*, se define en la **barra de entrada**: **mx = -10, Mx = 10, my = -10** y **My = 10**. A continuación, se crea un deslizador **t** definido en **[0,1]**, con incremento **0.001** y se definen los elementos: **n = 300, L = Secuencia[(mx + (Mx - mx) t, my + i (My - my) / n), i, 0, n], f(x) = sin(x)** y **g(x) = cos(x)**. Tras ocultar las dos funciones creadas, se habilita la **hoja de cálculo** y se rellenan las siguientes casillas: **A1=1, A2=A1+1, A3=A2+1**. Seguidamente se usa el control de relleno para completar la columna **A** hasta la celda **300**. Tras rellenar la casilla **B1=Elemento[L, A1]**, se activa el rastro de **B1** y, en sus **propiedades avanzadas**, se definen los siguientes colores dinámicos: Rojo: **f(x(B1)) + g(y(B1))**, Verde: **f(x(B1)) / g(y(B1))**, Azul: **f(x(B1)) g(y(B1))**. Tras usar el control de relleno para completar la columna B hasta la celda **300**, basta activar la animación automática de **t**[13].

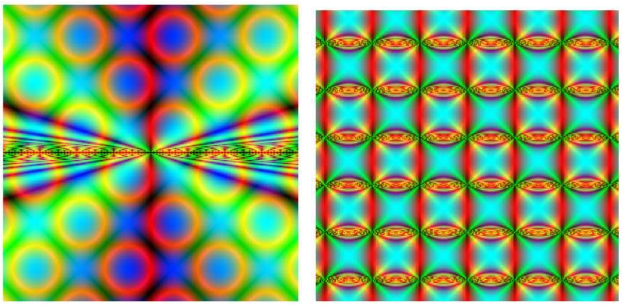

Figura 2: Diseños gráficos obtenidos con operaciones simples entre funciones.

10. Conclusiones.

En el presente trabajo se han esbozado varias actividades que muestran algunas de las posibilidades que ofrece GeoGebra en relación al campo del Análisis Matemático. En concreto, se ha comprobado la optimización de recursos y resultados que posibilita una buena compenetración entre la vista gráfica, la vista algebraica y la hoja de cálculo. Una buena forma de ampliar el abanico de posibilidades que ofrece el programa en este sentido es observar las actividades que presenta Daniel Mentrard en su página web[14].

---

[11] http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/PL.html

[12] http://geogebra.es/color_dinamico/color_dinamico.html

[13] http://www.geogebra.org/en/upload/files/spanish/omthales/JornadasGeoGebra2011/DG.html

[14] http://dmentrard.free.fr/GEOGEBRA/Maths/Fonctions/fonctions.htm