



## Guía de introducción a Mathematica

### I. Objetivos

1. Familiarizar al estudiante con aplicaciones prácticas de los conceptos fundamentales de electricidad y magnetismo.
2. Introducir y aplicar funciones específicas de Mathematica para resolver problemas relacionados con electricidad y magnetismo.
3. Resolver integrales relacionadas con fenómenos de inducción magnética.

### II. Gráficas 2D de Funciones y Campos Vectoriales en Mathematica

Para la representación visual de funciones y campos vectoriales en 2D en Mathematica, utilizamos las funciones **"Plot"** para graficar funciones y **"VectorPlot"** para graficar campos vectoriales.

#### Gráfica de funciones 2D: Plot

La función **Plot** permite graficar funciones de una sola variable. Se utiliza de la siguiente manera:

```
Plot[Funcion, {Variable, Valor_inicial, Valor_final}]
```

Listing 1: Gráfica de una función en 2D

En esta estructura, **función** es la expresión matemática que se desea graficar, y la **variable** es la variable independiente sobre la cual se grafica la función. El intervalo  $\{valor\_inicial, valor\_final\}$  determina el dominio de la función que se graficará. A continuación se muestra un ejemplo para graficar una función cuadrática  $y = x^2$ :

```
Plot[x^2, {x, -5, 5}]
```

Listing 2: Gráfica de la función cuadrática

Este código genera una gráfica de la función  $y = x^2$  en el intervalo de  $-5 \leq x \leq 5$ .

Este gráfico muestra una parábola que abre hacia arriba, y puedes ver cómo los valores de  $y$  aumentan a medida que  $x$  se aleja del origen. Más detalles y personalización de gráficos con **Plot** pueden encontrarse en **Plot**.

## Gráfica de Campos Vectoriales 2D: VectorPlot

La función **VectorPlot** se utiliza para graficar campos vectoriales en 2D. Un campo vectorial en 2D puede describir direcciones y magnitudes de vectores en un plano. La sintaxis general es la siguiente:

```
VectorPlot[{Componente_x, Componente_y}, {Variable_x, Valor_inicial, Valor_final},
           {Variable_y, Valor_inicial, Valor_final}]
```

Listing 3: Gráfica de un campo vectorial en 2D

En esta estructura, **componente\_x** y **componente\_y** son las expresiones que definen las componentes del campo vectorial en las direcciones  $x$  y  $y$ . Los intervalos  $\{valor\_inicial, valor\_final\}$  determinan el rango de las variables  $x$  y  $y$  sobre los cuales se graficará el campo.

El gráfico resultante muestra un campo vectorial con flechas que indican la dirección y la magnitud de los vectores en cada punto del plano. Más información y ejemplos adicionales de cómo trabajar con **VectorPlot** están disponibles en **VectorPlot**.

**StreamPlot: Representación de Líneas de Flujo** La función **StreamPlot** es una alternativa a **VectorPlot**, pero en lugar de dibujar vectores individuales, genera líneas de flujo que siguen las trayectorias de las partículas en el campo vectorial. La sintaxis es similar a la de **VectorPlot**:

```
StreamPlot[{Componente_x, Componente_y}, {Variable_x, Valor_inicial, Valor_final},
           {Variable_y, Valor_inicial, Valor_final}]
```

Listing 4: Gráfica de líneas de flujo de un campo vectorial

La diferencia entre ambas representaciones radica en que **VectorPlot** muestra vectores individuales en puntos discretos, mientras que **StreamPlot** visualiza trayectorias suaves que siguen el flujo del campo vectorial.

## Ejemplo Practico: Campo Eléctrico de un Dipolo

El campo eléctrico de un dipolo se genera por dos cargas de igual magnitud y signo opuesto, separadas por una distancia pequeña. Su intensidad y dirección varían en función de la posición en el espacio. La representación de campos vectoriales es algo más compleja que la de una función común, pero se detallará cómo definimos las ecuaciones del campo eléctrico y por qué definimos el código como lo hemos hecho.

## Ecuaciones del Campo Eléctrico

$$\mathbf{E}(\mathbf{r}) = \frac{1}{4\pi\epsilon_0 r^3} [3(\mathbf{p} \cdot \hat{r})\hat{r} - \mathbf{p}] \quad (1)$$

donde  $\mathbf{p}$  es el momento dipolar eléctrico,  $\hat{r}$  es el vector unitario en la dirección radial y  $r$  la distancia al centro del dipolo.

La implementación de campos vectoriales suele requerir cierto conocimiento sobre cómo los software interpretan el código a ejecutar. Para este ejemplo, lo haremos de la forma más simple posible, definiendo la ecuación dentro del argumento de las funciones **VectorPlot** y **StreamPlot** para evitar conflictos de interpretación en las variables del entorno de Mathematica.

## Implementación en Mathematica

```
(*Definimos las constantes*)
p = {1, 0}; (*Dipolo alineado en el eje x*)
eps0 = 1; (*Para simplificar la expresion*)

(*Definimos el campo electrico directamente dentro de VectorPlot*)
VectorPlot[(1/(4*Pi *eps0*Norm[{x, y}]^3))*(3 (p . Normalize[{x, y}])*
  Normalize[{x, y}] - p), {x, -2, 2}, {y, -2, 2},
  VectorScaling -> Automatic, VectorPoints -> 20]

(*Graficamos el campo electrico con StreamPlot*)
StreamPlot[(1/(4*Pi *eps0*Norm[{x, y}]^3))*(3 (p . Normalize[{x, y}])*
  Normalize[{x, y}] - p), {x, -2, 2}, {y, -2, 2},
  StreamPoints -> 50]
```

Listing 5: Grafico2D del campo electrico de un dipolo

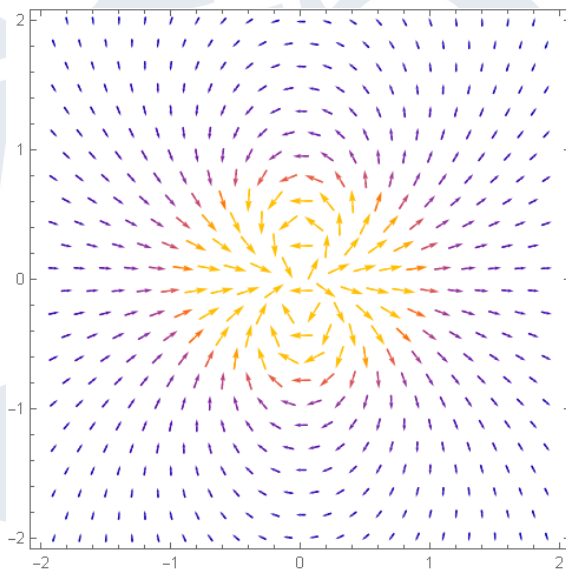


Figura 1: Gráfico con **VectorPlot**

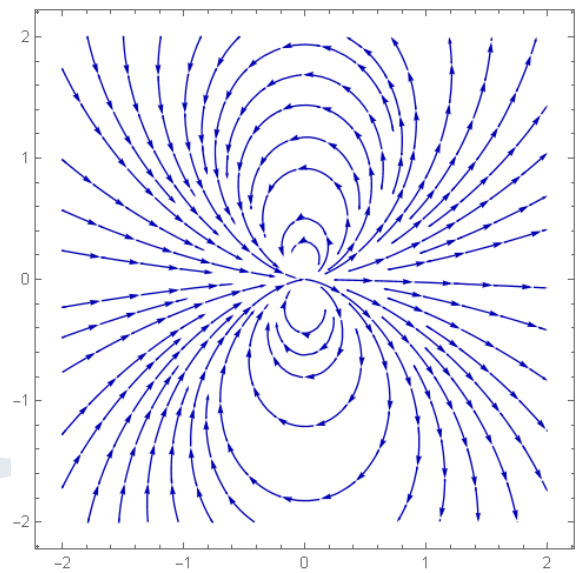


Figura 2: Gráfico con **StreamPlot**

En este código,  $r = \text{Norm}[\{x, y\}]$  es la distancia radial entre el dipolo y el punto de observación, y  $\hat{r} = \text{Normalize}[\{x, y\}]$  es el vector unitario en la dirección de  $\mathbf{r}$ . Esta fórmula describe el campo eléctrico generado por un dipolo en el plano  $xy$ , donde el término  $3(\mathbf{p} \cdot \hat{r})\hat{r}$  corresponde a la parte de la dirección radial del campo, y  $-\mathbf{p}$  a la parte perpendicular al campo.

## III. Resolución de ecuaciones diferenciales en Mathematica

### DSolve: Resolvedor Diferencial

Para la resolución de ecuaciones diferenciales analíticas, empleamos la función **"DSolve"**, la cual permite obtener la solución a partir de las condiciones del sistema. En Mathematica, la función y los argumentos que recibe se escriben de la siguiente manera:

```
DSolve[Equaciones, Variable_Dependiente[Variable_Independiente],
Variable_Independiente]
```

Listing 6: Funcion DSolve y sus argumentos

La función **DSolve** recibe la ecuación diferencial a trabajar, junto con la variable sobre la que el software va a encontrar sus soluciones y, por último, la variable independiente del sistema. Además, puede recibir otros argumentos que definen las condiciones iniciales del sistema o el dominio de la variable independiente. En el siguiente ejemplo se ilustra mejor cómo introducir estos argumentos. Procederemos a resolver la ecuación:

$$x''(t) + 2x(t) = 0 \quad \text{donde} \quad x(0) = 0, \quad x'(0) = 5 \quad (2)$$

```
DSolve[{x''[t] + 2*x[t] == 0, x[0] == 0, x'[0] == 5}, x[t], t]
```

Listing 7: Funcion diferencial con parametros de condiciones iniciales

```
{{x[t] -> (5 Sin[Sqrt[2] t])/Sqrt[2]}}
```

Listing 8: Output al problema de condiciones iniciales

En el ejemplo anterior, la ecuación de segundo orden tiene como condición inicial  $x(0) = 0$  y  $x'(0) = 5$ . Las condiciones cuentan como ecuaciones que recibe la función **DSolve**, por lo que deben ir todas las ecuaciones entre corchetes y separadas por comas. Más información y ejemplos sobre las posibilidades de la función pueden encontrarse en **DSolve**.

### NDSolve: Resolvedor diferencial numérico

En la física, cuando una función es de tipo no analítica, suelen hacerse aproximaciones para dar solución, o en su defecto, obtener los valores numéricos que satisfacen las ecuaciones. Para obtener estos valores numéricos empleamos la función **NDSolve**.

```
NDSolve[Equations, Variable_Dependiente, {Variable_Independiente, Upper_Limit,
Lower_Limit}]
```

Listing 9: Funcion NDSolve y sus argumentos

La función recibe la ecuación diferencial a resolver, junto con la variable a resolver y un rango de valores para los que nos interese obtener soluciones. Dado que la función no suele ofrecer un *Output* que nos permita analizar el fenómeno, suele hacerse una gráfica de la misma que nos permite ver el comportamiento de la función en la región especificada. Aquí se muestra un pequeño ejemplo para la siguiente ecuación no analítica:

$$y'(x) = -y(x) + \sin x \quad \text{donde} \quad y(0) = 1 \quad (3)$$

```
sol = NDSolve[{y'[x] == -y[x] + Sin[x], y[0] == 1}, y, {x, 0, 10}]
Plot[Evaluate[y[x] /. sol], {x, 0, 10}, PlotStyle -> Thick, AxesLabel -> {x, y}]
```

Listing 10: Funcion diferencial con parametros de condiciones iniciales.

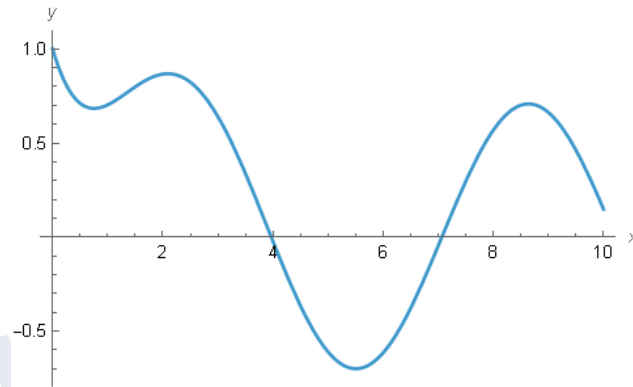


Figura 3: Gráfico de la solución a la ecuación diferencial

Tratamos de dar solución a la ecuación  $y'(x) = -y + \sin x$  con condición inicial  $y(0) = 1$  en el rango de  $[0, 10]$ . En cuanto a la gráfica, escribimos " $y[x] /. sol$ ", lo que permite que se reemplace  $y[x]$  por su solución numérica, y eso dentro de **Evaluate**, cuya función es evaluar la solución antes de graficarla.

### Ejemplo práctico: Péndulo simple

El péndulo simple es un sistema clásico en mecánica que permite estudiar el movimiento oscilatorio. Su análisis involucra una ecuación diferencial que describe la dinámica angular del sistema.

### Ecuación del Movimiento

El péndulo simple consiste en una masa  $m$  suspendida de una cuerda de longitud  $L$ . La ecuación de movimiento proviene de la segunda ley de Newton aplicada a la componente tangencial:

$$mL \frac{d^2\theta}{dt^2} = -mg \sin \theta. \quad (4)$$

Dividiendo por  $mL$ :

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin \theta = 0. \quad (5)$$

Para pequeñas oscilaciones, podemos aproximar  $\sin \theta \approx \theta$ , obteniendo la ecuación diferencial lineal:

$$\frac{d^2\theta}{dt^2} + \omega^2 \theta = 0, \quad \text{donde } \omega = \sqrt{\frac{g}{L}}. \quad (6)$$

### Solución Analítica

La ecuación diferencial resultante es de la forma de un oscilador armónico simple, cuya solución general es:

$$\theta(t) = \theta_0 \cos(\omega t + \phi), \quad (7)$$

donde  $\theta_0$  es la amplitud y  $\phi$  es una fase inicial determinada por las condiciones iniciales.

## Implementación en Mathematica

```
(* Definir parametros *)
g = 9.81;
L = 1;
omega = Sqrt[g/L];

(* Solucion analitica *)
DSolve[{theta''[t] + omega^2 theta[t] == 0, theta[0] == Pi/6, theta'[0] == 0},
  theta[t], t]
```

Listing 11: Solucion analitica al pendulo simple

```
{{theta[t] -> 0.523599 Cos[3.13209 t]}}
```

Listing 12: Output de la solucion analitica del pendulo simple

## Solución numérica

Tratando la ecuación diferencial sin hacer ninguna aproximación para ángulos pequeños, podemos darle solución numérica al problema. En este caso, el ángulo inicial será  $\pi/2$ .

## Implementación en Mathematica

```
(*Conservando las constantes definidas en el caso analitico*)
sol = NDSolve[{Derivative[2][theta][t] + (g/L) Sin[theta[t]] == 0,
  theta[0] == Pi/2, theta'[0] == 0}, theta, {t, 0, 10}];

(*Grafica de la solucion*)
Plot[Evaluate[theta[t] /. sol], {t, 0, 10}, AxesLabel -> {t, theta}]
```

Listing 13: Solucion numerica al pendulo simple

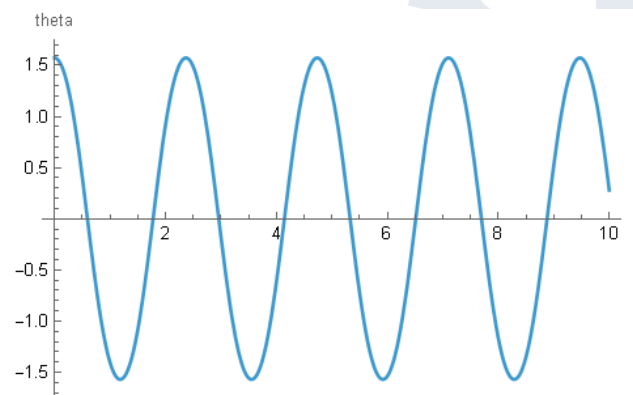


Figura 4: Gráfico de las soluciones numéricas del péndulo simple

## IV. Ajustes lineales y no lineales en Mathematica

### Ajuste Lineal: Función Fit en Mathematica

Para realizar un ajuste lineal de un conjunto de datos en Mathematica, empleamos la función **"Fit"**, la cual permite encontrar una curva ajustada de la forma más adecuada según el modelo que se

elija. En el caso del ajuste lineal, el modelo corresponde a una recta de la forma  $y = Ax + B$ , donde  $A$  es la pendiente y  $B$  es la intersección con el eje  $y$ . La función se emplea de la siguiente manera:

```
Fit[Datos, {Base_1, Base_2, ..., Base_n}, Variable]
```

Listing 14: Función Fit y sus argumentos

La función **Fit** recibe los datos a ajustar, un conjunto de bases (en el caso de una recta, 1 y  $x$ ) y la variable independiente  $x$ . Estos argumentos definen el modelo del ajuste lineal. En el siguiente ejemplo se ilustra cómo introducir estos argumentos utilizando un conjunto de datos específicos:

```
datos= {{1, 2}, {2, 4}, {3, 5}, {4, 8}};
Fit[datos, {1, x}, x]
```

Listing 15: Ajuste lineal de datos

```
-4.44089*10^-16 + 1.9 x
```

Listing 16: Resultado del ajuste lineal

En este ejemplo, el conjunto de datos consiste en pares de puntos  $(x, y)$ , y la función **Fit** ajusta una recta a esos puntos, resultando en la ecuación  $y = 2 + 1.5x$ , donde  $A = 1.5$  y  $B = 2$ . Más información y ejemplos de las posibilidades de la función pueden encontrarse en **Fit**.

### Ajuste No Lineal: Función NonlinearModelFit en Mathematica

Para realizar un ajuste no lineal de un conjunto de datos en Mathematica, empleamos la función **NonlinearModelFit**, que permite encontrar la mejor curva de ajuste a partir de un modelo no lineal especificado por el usuario. A diferencia del ajuste lineal, en este caso el modelo no tiene que ser una recta, sino que puede tomar la forma de cualquier función no lineal, como una exponencial, una función trigonométrica, una función de potencia, entre otras. La función se emplea de la siguiente manera:

```
NonlinearModelFit[Datos, Modelo_no_lineal, Variables, Variable_independiente]
```

Listing 17: Función NonlinearModelFit y sus argumentos

La función **NonlinearModelFit** recibe los datos a ajustar, el modelo no lineal que debe ser ajustado a los datos, las variables que deben ser ajustadas y la variable independiente  $x$ . Usualmente guardamos el resultado en una variable y luego usamos **Normal** para que nuestro output sea en forma funcional. En el siguiente ejemplo se ilustra cómo introducir estos argumentos utilizando un conjunto de datos específicos y un modelo exponencial:

```
Nlmf = NonlinearModelFit[{{1, 2}, {2, 5}, {3, 14}, {4, 40}},
  a Exp[b x], {a, b}, x]
Normal[Nlmf]
```

Listing 18: Ajuste no lineal de datos con modelo exponencial

```
0.619855 Exp[1.04164 x]
```

Listing 19: Output del ajuste no lineal



En este ejemplo, el modelo elegido es una función exponencial  $y = a \exp(bx)$ , donde  $a$  y  $b$  son los parámetros que se ajustarán. La función **Normal** devuelve la función con los valores de  $a$  y  $b$ . Cabe resaltar que podemos hacer más cosas, como graficar el ajuste del modelo o obtener los residuos. Más información de la función puede encontrarse en `NonlinearModelFit`.

## V. Resolución de Integrales

Para la resolución de integrales en Mathematica, empleamos la función **Integrate**, la cual permite calcular tanto integrales definidas como indefinidas. La forma básica de su uso es la siguiente:

```
Integrate[Expresion, Variable]
```

Listing 20: Uso básico de la función Integrate

En esta forma, **Expresión** es la función que se desea integrar, y **Variable** es la variable de integración. En el caso de una integral definida, se debe proporcionar también el intervalo de integración, como se muestra en el siguiente ejemplo:

```
Integrate[Expresion, {Variable, a, b}]
```

Listing 21: Integral definida

Aquí, **a** y **b** son los límites inferior y superior del intervalo de integración. Mathematica puede manejar tanto integrales sencillas como integrales más complejas, utilizando técnicas como la integración por partes, sustituciones trigonométricas, o el uso de tablas.

En el siguiente ejemplo se calcula una integral definida:

$$\int_0^1 x^2 dx \quad (8)$$

```
Integrate[x^2, {x, 0, 1}]
```

Listing 22: Ejemplo de integral definida

```
1/3
```

Listing 23: Resultado de la integral definida

En este caso, la integral de  $x^2$  en el intervalo de 0 a 1 se evalúa y el resultado es  $\frac{1}{3}$ . Además, **Integrate** puede recibir parámetros adicionales para cambiar el comportamiento de la integración, como el uso de métodos numéricos cuando la integral no tiene una solución cerrada analíticamente. Para más información, se puede consultar la documentación completa de la función `Integrate`.

## Resolución de Integral de Biot-Savart para el campo $\vec{B}$

La ley de Biot-Savart describe cómo una corriente eléctrica produce un campo magnético en el espacio. La expresión general de la ley es la siguiente:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int \frac{I d\mathbf{L} \times (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} \quad (9)$$



Para una corriente en una geometría simple, como una corriente infinita o una corriente en un lazo circular, la integral se puede simplificar. A continuación, se muestra un ejemplo de cómo aplicar la ley de Biot-Savart para una corriente infinita en el eje  $z$ , cuyo resultado es bien conocido como:

$$\mathbf{B} = \frac{\mu_0 I}{2\pi R} \hat{\theta} \quad (10)$$

## Implementación en Mathematica

```
(*Elemento de linea diferencial*)
dl = {0, 0, 1};

(*Vector de posicion desde la linea de corriente al punto de observacion*)
r = {x, y, z};

(*Magnitud de r*)
rMag = Sqrt[x^2 + y^2 + z^2];

(*Producto cruzado dl x r*)
crossProduct = Cross[dl, r];

(*Expresion de Biot-Savart*)
dB = (mu0 * I0 / (4 Pi)) * (crossProduct / rMag^3);

(*Integracion sobre todo el eje z*)
B = Integrate[ dB, {z, -Infinity, Infinity},
  Assumptions -> x^2 + y^2 > 0];
B
```

Listing 24: Resolución de Integral de Biot-Savart para una línea infinita

```
{-((I0 mu0 y)/(2 Pi (x^2 + y^2))), (I0 mu0 x)/(2 Pi (x^2 + y^2)), 0}
```

Listing 25: Output de la integral de Biot-Savart para una línea infinita

El resultado del output puede parecer no ser la expresión conocida, pero debemos recordar que en el resultado anterior  $x^2 + y^2 = \rho^2$ ,  $x = \rho \cos \theta$  y que  $y = \rho \sin \theta$ . Los cambios en la expresión anterior nos conducen al resultado simplificado que conocemos.

Una observación importante que hacer es que no se puede graficar el campo de inducción a partir del *output* de la integral, dado que es una expresión en parte simbólica y por ello no está definida correctamente como un argumento válido para la función **VectorPlot** o **StreamPlot**. Pero es posible emplear el resultado obtenido para definir estas componentes del campo  $\vec{B}$  directamente, tal que así:

```
(*Definiendo constantes*)
I0 = 1;
mu0 = 1;

(*Definir las funciones Bx y By directamente*)
Bx[x_, y_] := -((I0*mu0*y)/(2 Pi*(x^2 + y^2)))
By[x_, y_] := (I0*mu0*x)/(2 Pi*(x^2 + y^2))

(*Graficar el campo magnético usando StreamPlot*)
StreamPlot[{Bx[x, y], By[x, y]}, {x, -5, 5}, {y, -5, 5},
  StreamStyle -> "Arrow"]
```

Listing 26: Gráfico del campo de inducción  $B$  de una línea infinita de corriente

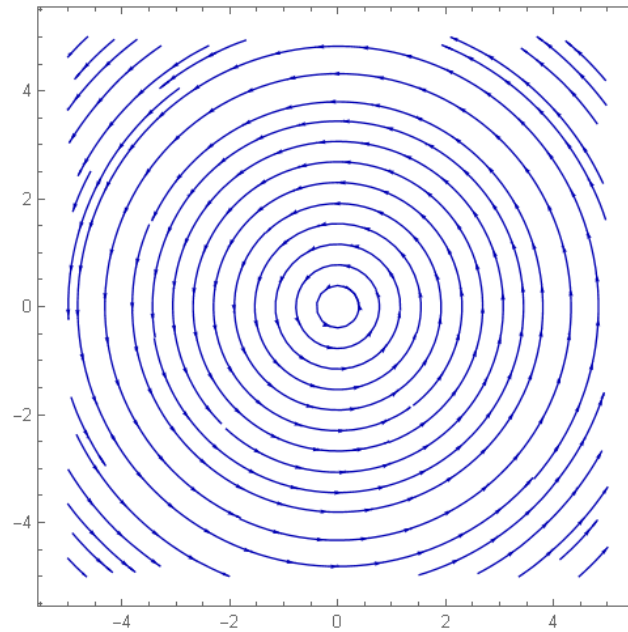


Figura 5: Gráfico del campo de inducción  $\vec{B}$

## VI. Ejercicio de tarea

Resuelva la integral de la ley de Biot-Savart ahora para el caso de una espira de radio  $\rho$  que se encuentra sobre el plano  $xy$ . Además, utilice la función **VectorPlot** o **StreamPlot** para visualizar la distribución del campo de inducción  $\vec{B}$  en el plano  $xz$  o en el plano  $yz$ .

## Referencias

- F.W. Zemansky, Young M.W Sears y Freedman R.A. (2009). *Física Universitaria: Con Física Moderna*. 12.<sup>a</sup> ed. Vol. 1. Addison-Wesley.
- Wangsness, Ronald K. (1968). *Campos electromagnéticos*. Primera. México: Editorial Limusa. ISBN: 9681813162.