Introducción a Mathematica

Ejemplo básico:

Un campo direccional simple $\frac{dx}{dt} = x$, con la condición inicial $x(0) = x_0$

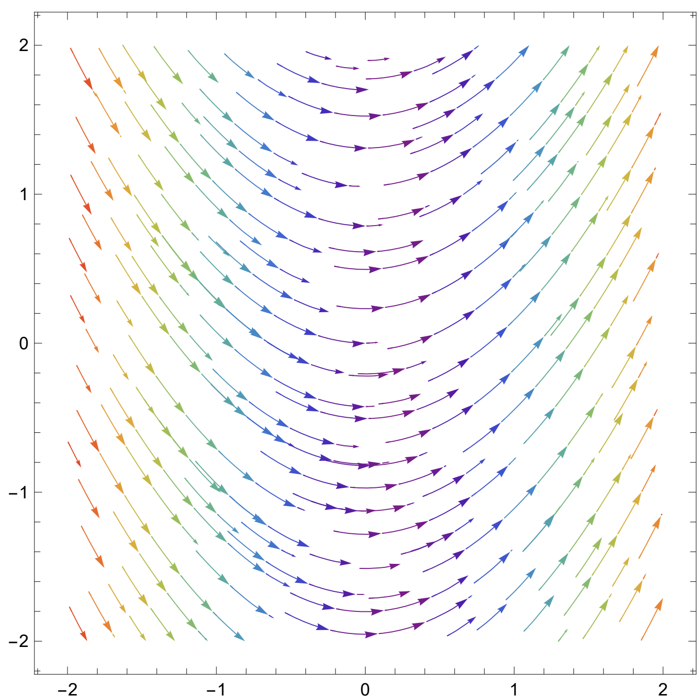
a) StreamPlot / VectorPlot

Para visualizar el campo direccional en el plano (t, x) o (en general) (x, y), solemos usar `StreamPlot` o `VectorPlot`. Sin embargo, para 1D solemos graficar la pendiente en un solo eje. Un ejemplo en 2D sería forzar una ecuación como $\frac{dx}{dt} = f(x, y)$.

In[2]:=

```
(* Versión 1D: dx/dt = x, "falsa" visualización en 2D para practicar *)
StreamPlot[{1, x}, {x, -2, 2}, {t, -2, 2}, StreamColorFunction → "Rainbow", AxesLabel → {"t", "x
```

Out[2]=



Este truco pone 1 para la componente en el eje horizontal (que sería t) y x como la componente vertical (d/dt), simulando las pendientes.

b) Solución analítica con DSolve

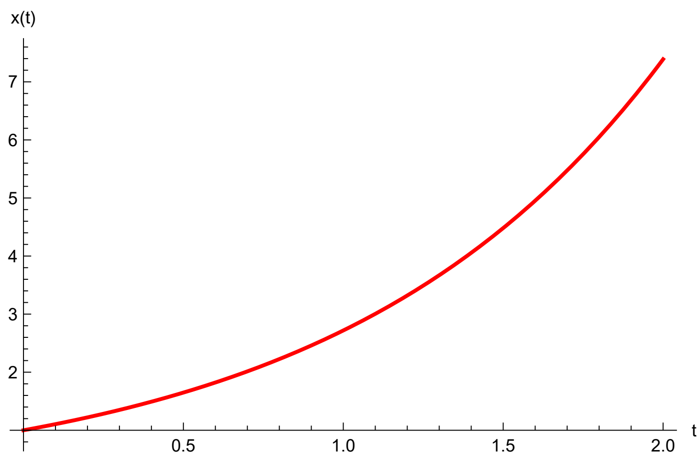
In[3]:=
```
sol = DSolve[{x'[t] == x[t], x[0] == x0}, x[t], t]
```

Out[3]= $\{\{x[t] \to e^{t}\, x0\}\}$

c) Graficar la solución con Plot

In[4]:=
```
Plot[Evaluate[x[t] /. sol /. x0 → 1], {t, 0, 2},
 AxesLabel → {"t", "x(t)"}, PlotStyle → Red]
```

Out[4]=

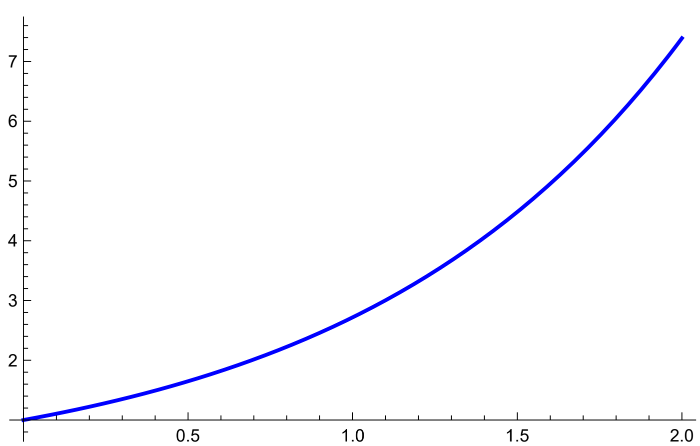


Tenemos que fijar $x_0 = 1$ para obtener una solución.

d) Solución numérica con NDSolve

In[5]:=
```
solNum = NDSolve[{x'[t] == x[t], x[0] == 1}, x, {t, 0, 2}];
Plot[Evaluate[x[t] /. solNum], {t, 0, 2}, PlotStyle → Blue]
```

Out[6]=



En problemas más complicados (no solucionables con `DSolve`), `NDSolve` es la opción.

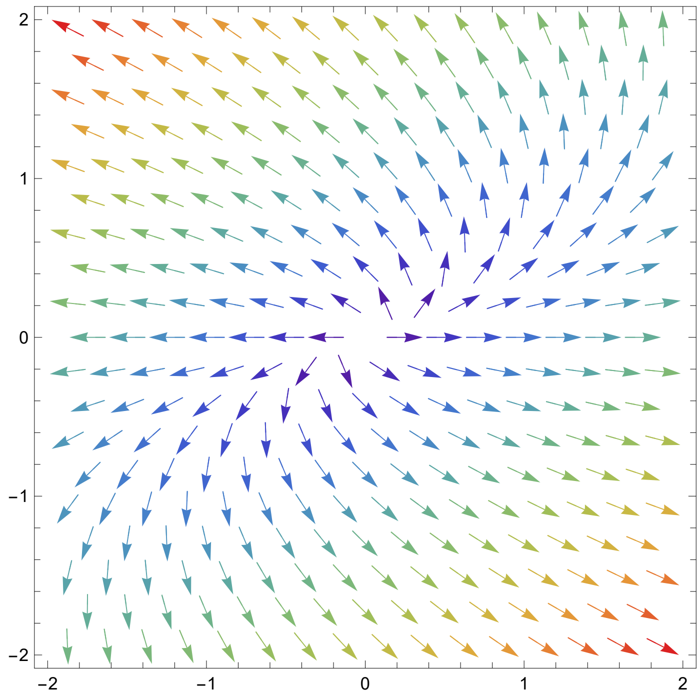
Campos direccionales en 2 variables

Para un sistema $\begin{cases} \frac{dx}{dt} = f(x, y), \\ \frac{dy}{dt} = g(x, y), \end{cases}$ podemos usar VectorPlot o StreamPlot.

In[7]:=
```
f[x_, y_] := x - y;
g[x_, y_] := y;

VectorPlot[{f[x,y], g[x,y]}, {x, -2, 2}, {y, -2, 2},
 VectorColorFunction → "Rainbow",
 AxesLabel → {"x", "y"}]
```

Out[9]=



Esto muestra las flechas del campo (f, g) en la región $(-2 \le x \le 2, \ -2 \le y \le 2)$.

Modelo presa-depredador simple (Lotka-Volterra)

$$\begin{cases} \frac{dx}{dt} = a\,x - b\,x\,y, \\ \frac{dy}{dt} = -c\,y + d\,x\,y. \end{cases}$$

donde:

x = población de presas,

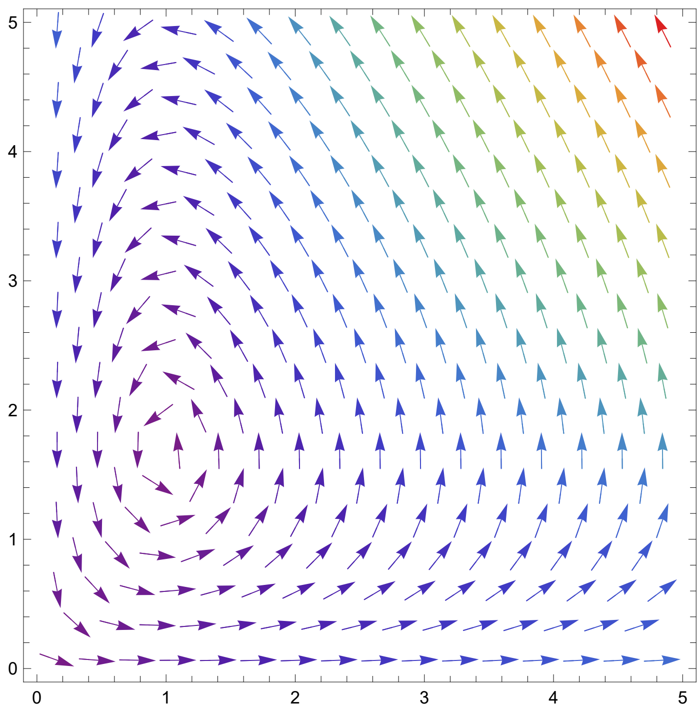
y = población de depredadores,
a,b,c,d son constantes positivas.

a) VectorPlot del campo

In[10]:=
```
a = 0.5; b = 0.3; c = 0.5; d = 0.5;
f[x_, y_] := a*x - b*x*y;
g[x_, y_] := -c*y + d*x*y;

VectorPlot[{f[x, y], g[x, y]}, {x, 0, 5}, {y, 0, 5},
 VectorColorFunction → "Rainbow",
 PlotRange → All,
 AxesLabel → {"x", "y"}]
```

Out[13]=



b) Solución numérica con NDSolve

Fijamos condiciones iniciales, por ejemplo: $x(0) = 1.5$ y $y(0) = 1.0$

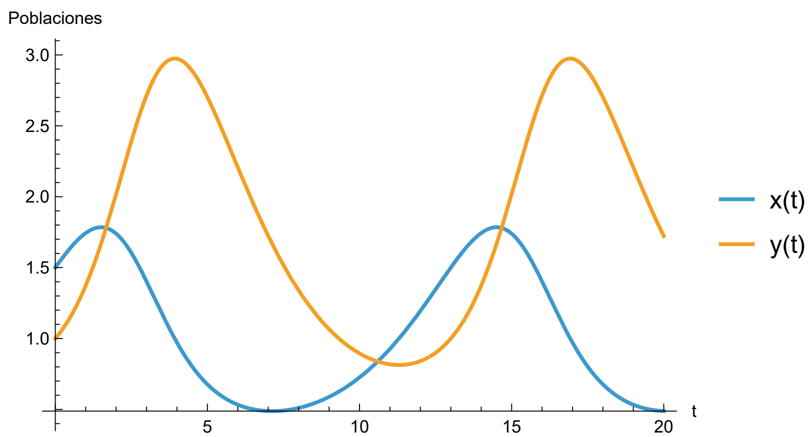
In[14]:=
```
solLV = NDSolve[
  {x'[t] == f[x[t], y[t]],
   y'[t] == g[x[t], y[t]],
   x[0] == 1.5, y[0] == 1.0},
  {x, y}, {t, 0, 20}
];
```

c) Graficar soluciones

En el tiempo:

In[15]:=
```
Plot[
  Evaluate[{x[t], y[t]} /. solLV], {t, 0, 20},
  PlotLegends → {"x(t)", "y(t)"},
  AxesLabel → {"t", "Poblaciones"}
]
```

Out[15]=



Plano fase (x,y)

In[16]:=
```
ParametricPlot[
  Evaluate[{x[t], y[t]} /. solLV], {t, 0, 20},
  PlotRange → All,
  PlotStyle → Thick,
  AxesLabel → {"x", "y"}
]
```

Out[16]=