

DNL_Bifurcaciones_Locales

May 3, 2022

```
[1]: using Plots, Interact, DifferentialEquations, BifurcationKit, Setfield, ␣
      ↪ ForwardDiff
      using NonLinearDynamicsCourse
```

WebIO._IJuliaInit()

1 Bifurcacion Saddle-Node en el plano y separacion de escalas temporales

Volvemos al sistema minimal con no linealidad cuadrática:

$$\dot{x} = a - y$$

$$\dot{y} = x^2 - y$$

Donde se puede ver que lo que sucede a medida que a se aproxima a cero por los positivos es que los dos puntos fijos (un punto silla y un atractor) se van acercando y para $a = 0$ colapsan y se produce una bifurcacion de nodo-silla en el plano.

El cambio del flujo debido a la bifurcacion es analogo a lo que observabamos en 1D, ya que la bifurcacion se produce a lo largo de una curva y en la direccion perpendicular a la curva el flujo es atractor. Esto se ve mas claro si le agregamos un parametro b que permita amplificar la disipacion (obviamente positivo).

$$\dot{x} = a - y$$

$$\dot{y} = x^2 - by$$

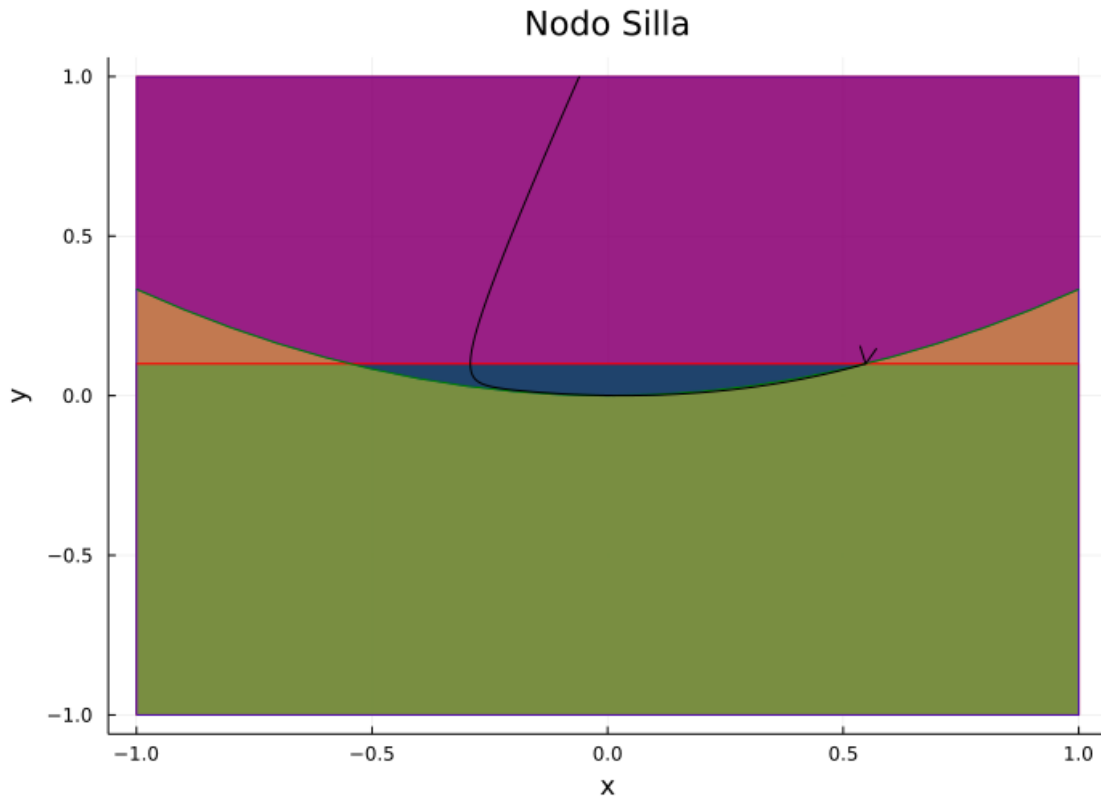
Cuando b es muy grande como se puede probar en las celdas que siguen el flujo converge muy rapidamente a la nulclina de $\dot{y} = 0$ y una vez sobre esa curva se mueve mas lentamente hacia el nodo atractor. Es decir que en este caso la dimension adicional entra como una variable rapida (un transitorio), pero la dinamica interesante sucede a lo largo de la curva (parabola) y sobre esa curva se produce la bifurcacion (explorarlo en el grafico interactivo debajo).

```
[2]: function nodosilla2d!(du,u,p,t)
      (a,b) = p
      du[1] = a-u[2]
      du[2] = u[1]*u[1]-b*u[2]
      du
    end
```

[2]: nodosilla2d! (generic function with 1 method)

```
[28]: # Esto es interactivo en ipynb
xini=-0.06;yini=1.0;a=0.1;b=3.0
#@manipulate throttle=0.5 for xini=-1:0.01:1, yini=-1:0.1:1,a=0.0:0.01:0.2,b=1.
↪0:0.1:5.0
    flux2d_nullclines(nodosilla2d!, [xini;yini], 30.0, [a,b]; title="Nodo Silla")
#end
```

[28]:



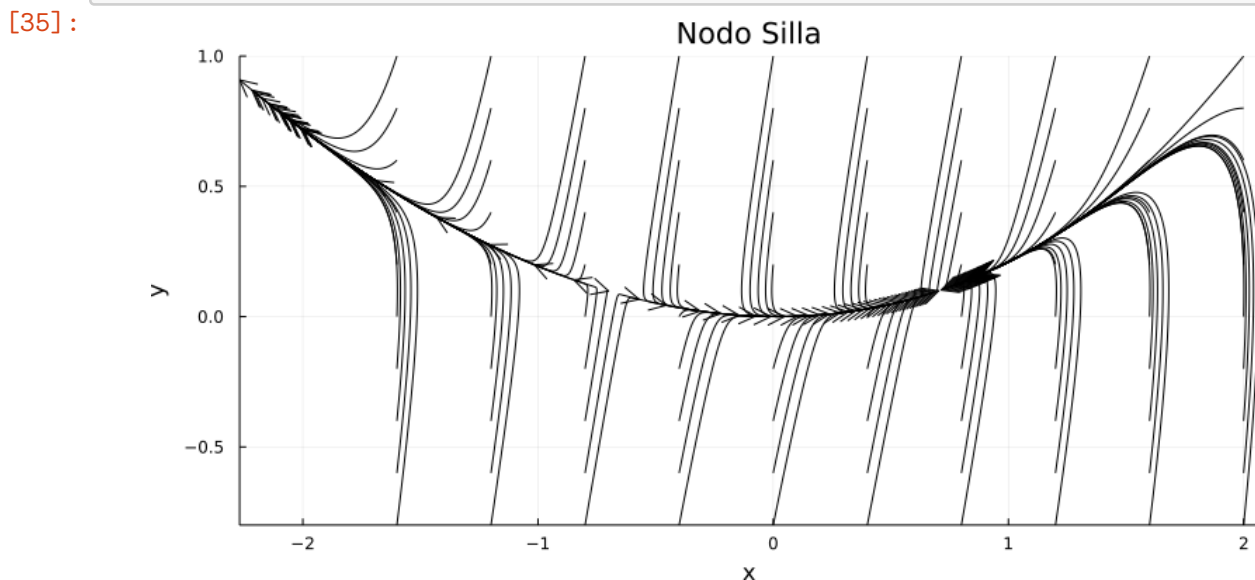
Esta separación de escalas de tiempo, la misma que veíamos en los osciladores de relajación es muy importante ya que: - muchos sistemas naturales (y artificiales) presentan esta separación de escalas de tiempo, cada dinámica tiene su propio tiempo característico - nos permite separar el análisis del problema no lineal en dos etapas; una primera etapa donde el sistema relaja siguiendo una dirección de evolución rápida y podemos considerar a las otras direcciones lentas como parámetros, seguida de una evolución lenta sobre alguna variedad (en 2D una curva) donde en general ocurre la dinámica interesante - las bifurcaciones ocurren siempre sobre las variedades “lentas”, esto está vinculado con el Critical Slowing Down, siempre cerca de una bifurcación la dinámica se enlentece, básicamente porque en las otras direcciones transversales a la variedad el flujo converge (o diverge) exponencialmente, pero sobre la variedad se mueve enlentecido por la proximidad de una bifurcación - cuando se estudian sistemas de más dimensiones nos va a interesar lo que ocurre sobre las variedades lentas (o centrales) y vamos a colapsar la dinámica a una evolución con menos grados de libertad (por lo general uno o dos) sobre esas variedades, proceso que se conoce como **reducción**

a la **variedad central** - la bifurcaciones ocurren variedades de muy baja dimension. La Saddle Node por ejemplo ocurre siempre en una variedad unidimensional, la Hopf que vamos a ver más adelante en una variedad bidimensional. Y luego aunque tengamos sistemas n-dimensionales la mayor parte de los cambios cualitativos que ocurren en el flujo se van a poder explicar por estas bifurcaciones que ocurren en 1D o 2D.

Si bien todo esto puede parecer al principio muy abstracto, es uno de los conceptos centrales junto con el de formas normales en el estudio cualitativo de los sistemas dinámicos.

Como ejemplo podemos ver en la evolucion de multiples trayectorias en el gráfico de abajo (para un valor grande de b) que la dinámica rápida colapsa a un entorno de la nulclina parabolica y luego sobre esa region evoluciona como un sistema 1D en un entorno de una bifurcacion nodo silla.

```
[35]: a = 0.1; b = 5.0
function prob_func(prob,i,repeat)
    remake(prob,u0=u0_arr[i])
end
condition(u,t,integrator) = (u[1]*u[1]+u[2]*u[2]) > 4.5
affect!(integrator) = terminate!(integrator)
u0_arr=[[-2+i*0.4;-1+j*0.2] for i=1:10 for j=1:10]
prob = EnsembleProblem(ODEProblem(nodosilla2d!,u0_arr[1],(5.
    ↪0),[a,b]),prob_func=prob_func)
sol = solve(prob,callback=DiscreteCallback(condition,affect!),
    EnsembleThreads(),trajectories=length(u0_arr))
p1 = plot(sol,vars=(1,2),arrows=true;legend=false,xlabel="x",ylabel="y",c=:
    ↪black,
    size=(800,400),title="Nodo Silla",fmt=:png)
```



Vamos a hacer un cambio de coordenadas que pone de evidencia que este sistema tiende a la bifurcacion SN sobre una recta cuando b es grande.

$$w = \frac{y-a}{b} - x$$

$$z = \frac{x}{\sqrt{b}}$$

con la transformacion inversa:

$$x = \sqrt{b}z$$

$$y = bw + b\sqrt{b}z + a$$

El nuevo sistema queda escrito

$$\dot{w} = \frac{\dot{y}}{b} - \dot{x} = \frac{x^2 - by}{b} + y - a = \frac{x^2}{b} - a = z^2 - a$$

$$\dot{z} = \frac{\dot{x}}{\sqrt{b}} = \frac{1}{\sqrt{b}}(a - y) = \frac{1}{\sqrt{b}}(a - bw - b\sqrt{b}z - a) = -\sqrt{b}w - bz$$

Pasando en limpio, nos queda un sistema donde el término cuadrático queda solo con el parámetro de bifurcación:

$$\dot{w} = z^2 - a$$

$$\dot{z} = -\sqrt{b}w - bz$$

Como es un cambio lineal de coordenadas el sistema sigue teniendo dos puntos fijos para $a > 0$ con la misma estabilidad. El punto fijo estable está en:

$$w_* = \mp\sqrt{ab}$$

$$z_* = \pm\sqrt{a}$$

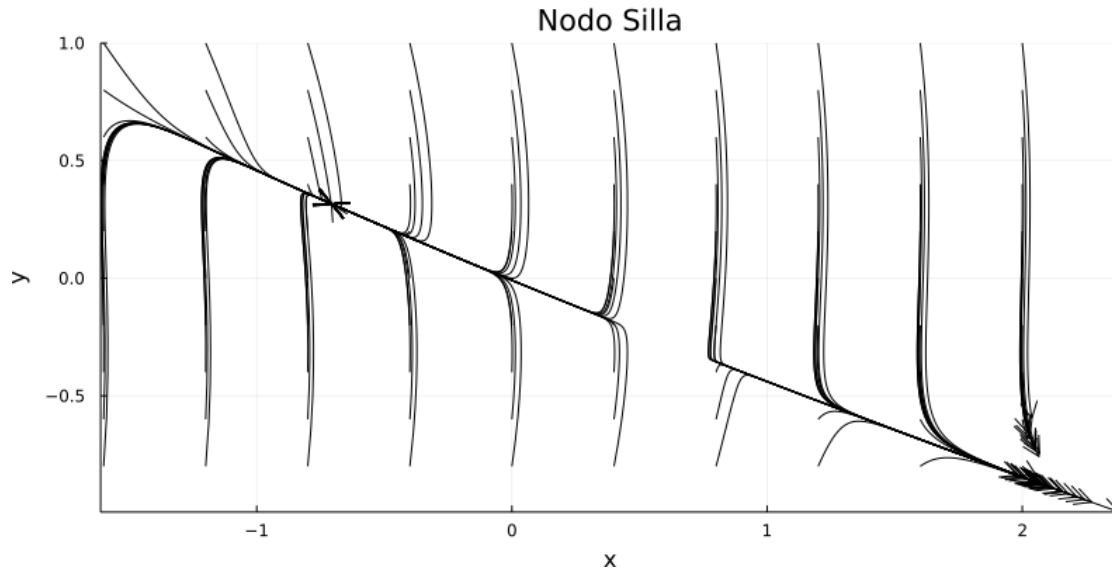
El cambio lineal de coordenadas nos garantiza que el retrato de fases es topologicamente equivalente, pero la forma del flujo puede cambiar porque es un sistema no lineal y de hecho este cambio lo que hace es “rectificar” la parabola y ahora la nulclina de la segunda variable es la recta $z = -w/\sqrt{b}$ como puede verse en el diagrama de abajo

```
[5]: function nodosilla2da!(du,u,p,t)
      (a,b) = p
      du[1] = u[2]*u[2]-a
      du[2] = -sqrt(b)*u[1]-b*u[2]
      du
    end
```

```
[5]: nodosilla2da! (generic function with 1 method)
```

```
[36]: prob = EnsembleProblem(ODEProblem(nodosilla2da!,u0_arr[1],(30.
      ↪0),[a,b]),prob_func=prob_func)
      sol = solve(prob,callback=DiscreteCallback(condition,affect!),
      EnsembleThreads(),trajectories=length(u0_arr))
      p1 = plot(sol,vars=(1,2),arrows=true;legend=false,xlabel="x",ylabel="y",c=:
      ↪black,
      size=(800,400),title="Nodo Silla",fmt=:png)
```

```
[36]:
```



En esta versión del sistema la variable z es la variable “rápida” que converge impulsada por $-\sqrt{b}w - bz$ a la nulclina diagonal. Si bien no tenemos el sistema desacoplado una vez que estamos sobre esa nulclina, al igual que en el caso de los osciladores de relajación la ecuación que gobierna el flujo es la “lenta”, ya que en este caso podemos considerar sobre la recta $z = w/\sqrt{b}$ y reemplazar en la primer ecuación, escalar el tiempo con $(\tau = t/b)$ y el parametro de bifurcación $\alpha = ab$, con lo cual nos queda la forma normal 1D de la saddle node:

$$\frac{dw}{dt} = z^2 - a = \frac{w^2}{b} - a$$

$$\frac{dw}{d(b\tau)} = \frac{w^2}{b} - \frac{\alpha}{b}$$

$$\frac{dw}{d\tau} = w^2 - \alpha$$

Finalmente puede ser instructivo ver como se modifican las variedades estables del punto silla con el parametro b en el sistema original. Recordar que estas curvas (en azul) actúan como separatrices del flujo, Todo lo que este del lado de estas curvas donde esta el atractor converge a el y todo lo que esta del otro se escapa. Si b es chico las variedades se acercan al atractor y la region estable es muy chica. A medida que b crece la variedad inestable (curva roja) sigue la parabola y la estable se transforma en una recta perpendicular a la parabola en ese punto, y todo lo que esta a la derecha queda en la cuenca de atracción del punto fijo de la derecha

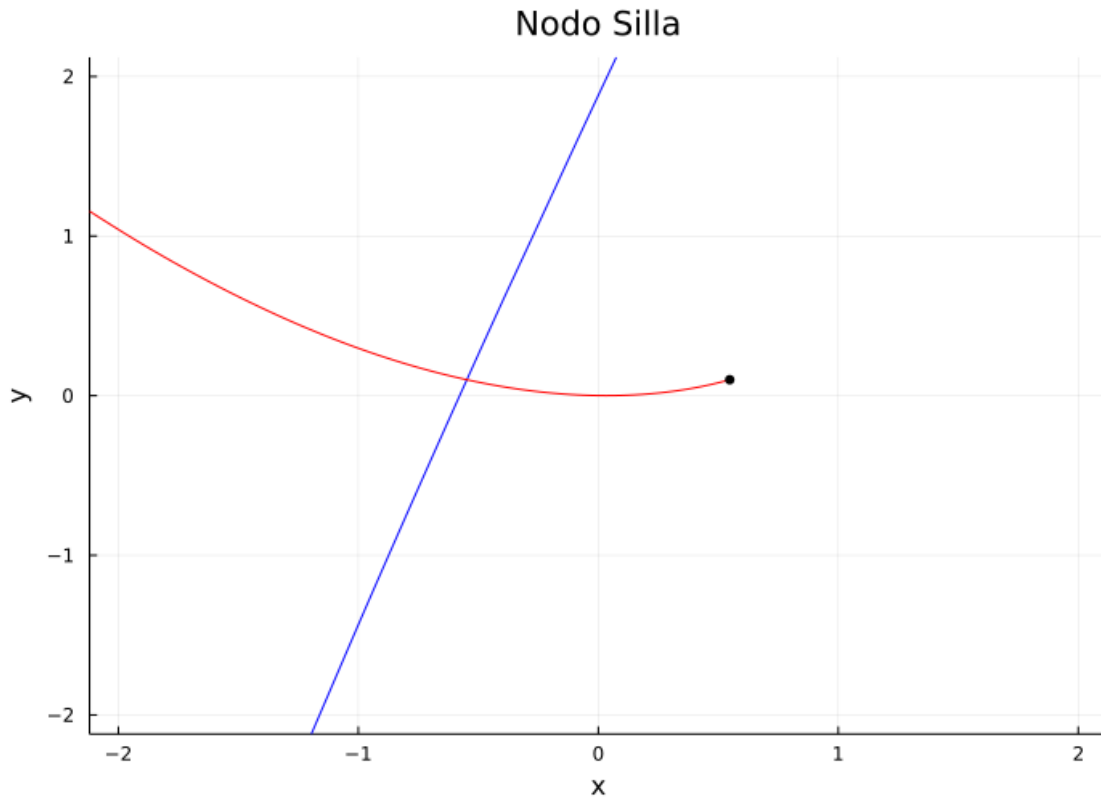
```
[7]: function nodosilla2d_jac(u,p)
    J = Array{Float64, 2}(undef, 2, 2)
    J[1,1] = 0
    J[1,2] = -1.0
    J[2,1] = 2*u[1]
    J[2,2] = -p[2]
    return J
```

```
end
```

```
[7]: nodosilla2d_jac (generic function with 1 method)
```

```
[29]: # Esto es interactivo en ipynb
a=0.1;b=3.0
#@manipulate for a=0.0:0.01:0.2,b=0.0:0.1:3.0
    u0_array=[[-sqrt(a*b);a],[sqrt(a*b);a]]
    flux2d_manifolds(nodosilla2d!,nodosilla2d_jac,u0_array,[a,b];
        tmax=300,delta=1e-7,xlims=[-2,2],ylims=[-2,2],title="Nodo Silla")
#end
```

```
[29]:
```



La cuenca de atracción del punto fijo estable es lo que queda del lado de la variedad estable del punto silla (curva azul) donde está el punto fijo. Notar como afecta el parámetro b la forma de la cuenca de atracción.

1.1 Estabilidad de los puntos fijos y autovalores

Los puntos fijos para $a > 0$ en el sistema original están en $(x_* = \pm\sqrt{ab}, y_* = a)$ y la matriz jacobiana evaluada en ese punto es:

$$\begin{pmatrix} 0 & -1 \\ \pm 2\sqrt{ab} & -b \end{pmatrix}$$

Que la traza sea siempre negativa ($-b$) implica que ninguno de ellos es un repulsor, solo puede ser un punto silla o un atractor. Esto es logico porque hay una direccion en la que los dos son atractores y es la direccion en la que converge rapidamente a la parabola. Mientras que el determinante es positivo para el punto a la derecha (atractor) y negativo para el de la izquierda (punto silla).

$$\Delta = 2\sqrt{ab} \quad , \text{ para } x_* = \sqrt{ab} \quad (\text{atractor})$$

$$\Delta = -2\sqrt{ab} \quad , \text{ para } x_* = -\sqrt{ab} \quad (\text{punto silla})$$

A medida que nos acercamos a la bifurcacion ambos puntos tienen la misma traza y el determinante se va aproximado a cero por los negativos y los positivos y colisionan en el eje $\Delta = 0$.

Es interesante que en el sistema transformado (w, z) si el flujo cambio y el jacobiano da diferente, el determinante y la traza y por ende sus autovalores son los mismos.

En efecto, los puntos fijos ahora son $(x_1^*, y_1^*) = (-\sqrt{ab}, \sqrt{a})$ y $(x_2^*, y_2^*) = (\sqrt{ab}, -\sqrt{a})$. La matriz jacobiana de forma general es:

$$\begin{pmatrix} 0 & 2z \\ -\sqrt{b} & -b \end{pmatrix}$$

cuya traza es como antes $-b$ y su determinante es

$$\Delta = 2\sqrt{ab} \quad , \text{ para } x_1^* \quad (\text{atractor})$$

$$\Delta = -2\sqrt{ab} \quad , \text{ para } x_2^* \quad (\text{punto silla})$$

Nota que la separación entre los puntos es la misma a pesar de que la transformación invirtió la orientación (en este caso el atractor está a la derecha).

Otra forma mas interesante de verlo es recordar que los autovalores de la matriz se podian escribir como

$$\lambda_1 = \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2} \quad \lambda_2 = \frac{\tau - \sqrt{\tau^2 - 4\Delta}}{2}$$

Para el atractor estos autovalores serian:

$$\lambda_1 = \frac{-b + \sqrt{b^2 - 8\sqrt{ab}}}{2} \quad \lambda_2 = \frac{-b - \sqrt{b^2 - 8\sqrt{ab}}}{2}$$

Ambos autovalores son negativos (y reales porque recordemos que b es un valor grande y estamos estudiando el imite de la bifurcacion de a acercandose a cero desde los positivos) porque le estamos sumando y restando a $-b$ un numero que es mas chico en valor absoluto (la raiz es mas chica).

y para el punto silla:

$$\lambda_1 = \frac{-b + \sqrt{b^2 + 8\sqrt{ab}}}{2} \quad \lambda_2 = \frac{-b - \sqrt{b^2 + 8\sqrt{ab}}}{2}$$

Ahora la raiz es mas grande que b en valor absoluto asi que el primer autovalor es positivo y tiene asociado un autovector que apunta en la direccion repulsora del punto silla (se puede calcular que para b grande es tangente a la parabola).

Lo interesante es que a medida que nos aproximamos a la bifurcacion el segundo autovalor tiende en ambos casos a $-b$, pero el primero tiende a cero (por los negativos y por los positivos) y en la bifurcacion es estrictamente cero. Esta condicion es necesaria en general para que ocurra la bifurcacion de nodo silla en cualquier numero de dimensiones. Podemos tener varias direcciones con autovalores diferentes de cero, pero si en una de las direcciones uno de los autovalores se hace cero (o mas estrictamente cruza el cero por el eje real para distinguirla de otras) entonces

tenemos una bifurcación de nodo silla. De un lado de la bifurcación tendremos dos puntos fijos (con autovalores positivos y negativos en esa dirección) y del otro lado esos dos puntos fijos no existen más. Hay algunas condiciones que se tienen que cumplir para distinguirla de la transcritical y la pitchfork (y de otras que se van a ver más adelante), pero por ahora nos vamos a quedar con esa condición

1.1.1 Detección de la Nodo Silla usando BifurcationKit

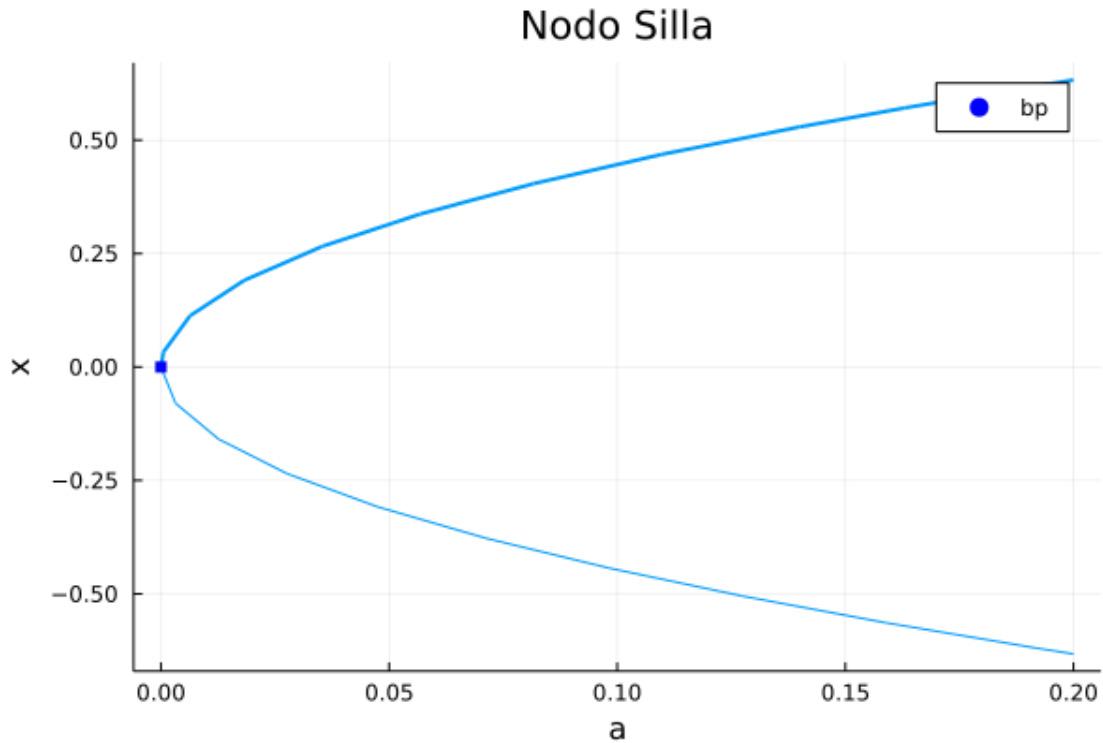
Vamos a usar BifurcationKit para barrer esta bifurcación. Como antes tenemos que definir una versión de la función “out of place” y sin dependencia temporal, eso lo hacemos en la primera línea.

Luego barremos de -0.1 a 0.2 pero **hacia atrás** porque tenemos que partir de un valor de a positivo para detectar puntos fijos. Eso se indica con un valor de ds negativo en las opciones de ContinuationPar (mientras que pMax pMin son los valores máximo y mínimo y dsmax el máximo paso permitido).

La función continuation recibe como argumentos obligatorios: - la función del campo vector - un “guess” para el punto fijo para el valor inicial de los parámetros. No es necesario que sea la posición exacta del punto fijo sino un valor cercano para el método numérico. - los valores iniciales de parámetros (aquí usamos $b = 0$ y colocamos el primer valor de a en pMax). - el argumento @lens_[1] indica que vamos a barrer el primer parámetro. - las opciones de continuación que definimos anteriormente

```
[9]: nodosilla2d(u,p) = nodosilla2d!(similar(u),u,p,0)
      opts = ContinuationPar(pMin=-0.1,pMax=0.2, ds = -0.01, dsmax = 0.
      ↪04,detectBifurcation=3,nInversion=8)
      br1, = continuation(nodosilla2d,[0.1,0.1],[0.2,2.0],(@lens_[1]), opts;
      ↪recordFromSolution = (x, p) -> x[1])
      plot(br1,xlabel="a",title="Nodo Silla",fmt=:png)
```

[9]:



2 Bifurcacion Pitchfork en el plano

Vamos a volver al oscilador de Duffing con disipación:

$$\dot{x} = y$$

$$\dot{y} = -\gamma y + \beta x - x^3$$

Con puntos fijos en:

Para $\beta < 0$:

$$x_1^* = 0 \text{ atractor}$$

Para $\beta > 0$:

$$x_1^* = 0 \text{ punto silla}$$

$$x_{2,3}^* = \pm\sqrt{\beta} \text{ atractores}$$

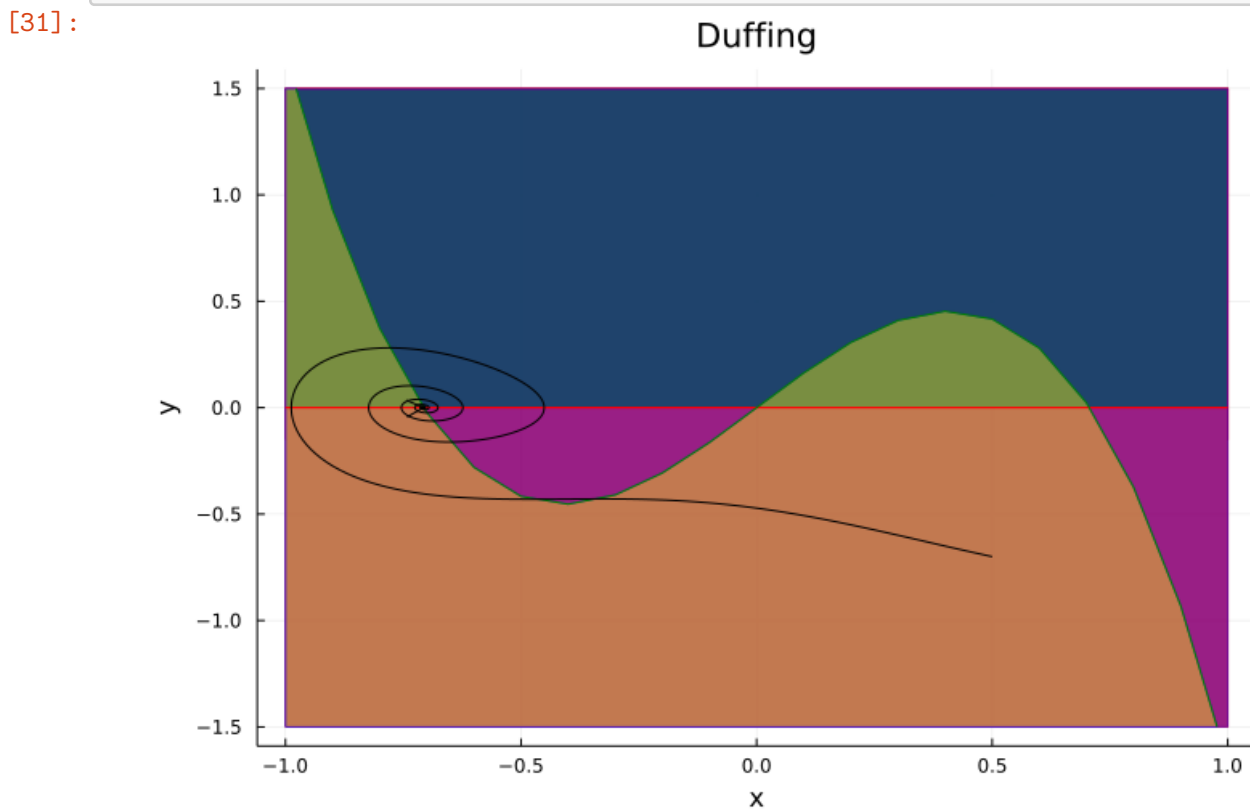
Cuando β cruza el cero el punto fijo x_1^* cambia de estabilidad de (deja de ser atractor) y al mismo tiempo aparecen otros dos puntos fijos en posiciones simétricas. Esto es una bifurcacion de Pitchfork.

Pero notar que el cambio de estabilidad es de un nodo atractor a un punto silla (no un repulsor). Eso es porque hay una direccion (transversal a la parabola) que es siempre atractora, y ese punto fijo luego de la bifurcacion pasa a tener una direccion inestable (punto silla) y esa direccion inestable es en la direccion tangente a la parabola. Verificar eso en el grafico interactivo.

Si permitiésemos a la disipación γ tomar valores elevados tendríamos una situación similar al caso anterior donde la bifurcación está ocurriendo sobre una curva (variedad) 1D mientras que hay una dirección transversal atractora. Sin embargo en el caso del oscilador de Duffing el interés por lo general está en valores intermedios o bajos de γ

```
[31]: #la Ed para el oscilador de Duffing definida in-place
function duffing!(du,u,p,t)
    du[1] = u[2]
    du[2] = -p[1]*u[2]+u[1]*(p[2]-u[1]*u[1])
    du
end

# Esto es interactivo en ipynb
xini=0.5;yini=-0.7;=0.3;=0.5
#@manipulate for xini=-1:0.02:1,yini=-1.5:0.2:1.5,=0.0:0.01:2.0,=-0.5:0.1:0.5
    flux2d_nullclines(duffing!, [xini;yini], 50.0, [ , ]; ylims=[-1.5,1.
↪5], title="Duffing")
#end
```



2.0.1 Detección de la Pitchfork usando BifurcationKit

Vamos a usar BifurcationKit para barrer esta bifurcación. A diferencia del caso anterior esta bifurcación no tiene una sola rama (se considera la SN una sola rama que “pega la vuelta” en la

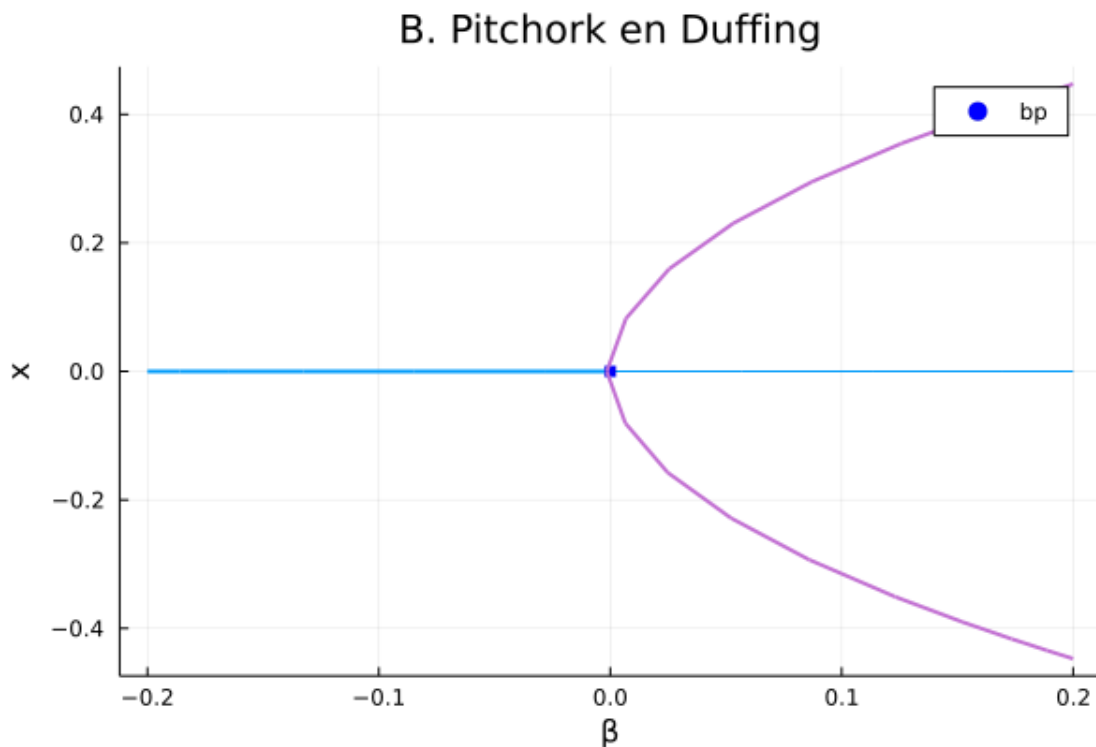
bifurcacion). Por ello tenemos que detectar dos ramas y eso lo podemos hacer solo para valores de β positivo donde tengo tres puntos fijos.

La estrategia es utilizar dos puntos iniciales diferentes como guess del punto fijo, uno muy cerca de cero para el punto silla y otro en un valor grande para tender hacia el atractor (tener en cuenta que el algoritmo detecta puntos fijos de cualquier tipo por lo tanto es irrelevante que sea un atracto un punto silla o un repulsor pero si tiene que estar cerca)

Notar que en este caso estamos barriendo el segundo parámetro, de allí `@lens _[2]` y que tambien tenemos que barrer hacia los negativos en la rama parabolica (br1). En la otra rama (br1) da igual porque existe para valores positivos y negativos de β .

```
[27]: duffing(u,p) = duffing!(similar(u),u,p,0)
      opts = ContinuationPar(pMin=-0.2,pMax=0.2, ds = 0.01, dsmax = 0.
      ↪04,detectBifurcation=3,nInversion=8)
      br1, = continuation(duffing,[0.1,0.1],[0.1,-0.2],(@lens _[2]), opts)
      opts = ContinuationPar(pMin=-0.2,pMax=0.2, ds=-0.01, dsmax=0.04,
      ↪detectBifurcation=3)
      br2, = continuation(duffing,[-0.5,0.1],[0.1,0.2],(@lens _[2]), opts;
      ↪recordFromSolution = (x, p) -> x[1])
      plot(br1)
      plot!(br2,xlabel="\\beta",title="B. Pitchork en Duffing",fmt=:png)
```

[27]:



3 Bifurcacion de Hopf

En Flujos 2D existe otra bifurcacion que no existe en 1D. Cuando un punto fijo pierde estabilidad, pasa de ser atractor a repulsor lo hace a traves de un centro (foco atractor que se transforma en repulsor). Esto sucede cuando el Determinante del Jacobiano es positivo y la Traza cambia de signo pasando por cero. Recordar que en este caso los autovalores eran complejos (foco) y la parte real era la traza asi que otra forma de caracterizar esta bifurcacion es cuando para un punto fijo **ambos** autovalores cruzan el eje imaginario (parte real cero).

En un sistema lineal eso significa unicamente un cambio de direccion en el flujo del foco (espiral atractor a repulsor). Pero en los sistemas no lineales los terminos de orden mas alto dan de forma generica origen a un ciclo limite que es expelido del foco y se produce una **Bifurcacion de Hopf** con el cambio de estabilidad.

Hay dos tipos de Bifurcación de Hopf, **supercritica** y **subcritica**. Por ahora nos va a interesar mas la supercritica y para este tipo la historia es asi: de un lado de la bifurcacion (en la figura para $\beta < 0$) tenemos un foco atractor, el Jacobiano tiene traza negativa. En la bifurcacion el foco se transforma en un centro ($\beta = 0$) y las trayectorias se siguen aproximando (por los terminos no lineales) pero no convergen nunca al centro. Y luego de la bifurcacion el punto fijo pierde estabilidad y se transforma en repulsor.

Pero ahora bien, como es una bifurcacion LOCAL el flujo lejos “no se entera” del cambio de estabilidad y sigue siendo atractor (dominan los terminos no lineales). Por lo tanto el flujo sigue siendo globalmente atractor y se acerca a la región del centro. Pero como al mismo tiempo es expelido de el por el foco repulsor, en alguna parte se tiene que acumular el flujo y se forma un ciclo limite estable. Es decir que del otro lado de la bifurcacion (para $\beta > 0$) tengo un foco repulsor que alimenta a un ciclo limite atractor.

La forma normal de la bifurcacion de Hopf (supercritica) es la siguiente

$$\dot{x} = \beta x - \omega y - x(x^2 + y^2)$$

$$\dot{y} = \omega x + \beta y - y(x^2 + y^2)$$

Parece complicado pero veamos primero que el punto fijo esta en $(0,0)$ y si queremos calcular el Jacobiano va a tener un monton de terminos pero solo van a sobrevivir los que no tengan ni x ni y porque esos van a ser cero. Asi que el Jacobiano evaluado en el punto fijo $(0,0)$ es solo el de la parte lineal y es:

$$\begin{pmatrix} \beta & -\omega \\ \omega & \beta \end{pmatrix}$$

La traza es 2β y el determinante siempre positivo $\beta^2 + \omega^2$. Por lo tanto cuando β cruza el cero cambia el signo de la Traza y el foco pasa de ser atractor a repulsor como muestra la figura.

Por otro lado los terminos no lineales tienen una expresion que equivale al radio al cuadrado ($x^2 + y^2$) con la variable cambiada de signo por lo tanto siempre van a ser atractores lejos.

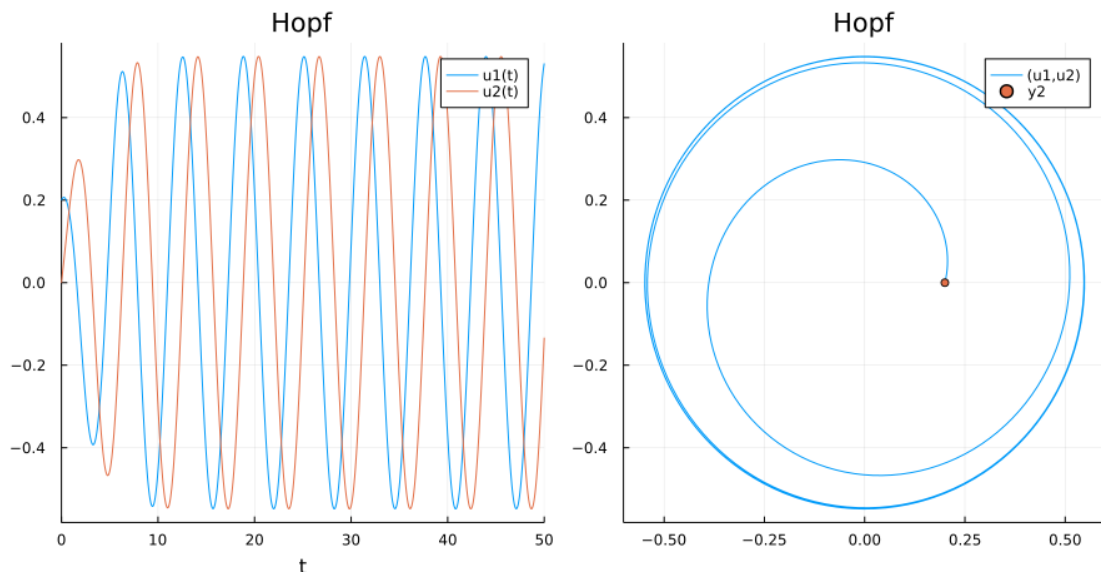
```
[12]: function hopf!(du,u,p,t)
      ( , ) = p
      du[1] = *u[1] - *u[2] - u[1]*(u[1]*u[1]+u[2]*u[2])
      du[2] = *u[1] + *u[2] - u[2]*(u[1]*u[1]+u[2]*u[2])
      du
```

```
end
```

[12]: hopf! (generic function with 1 method)

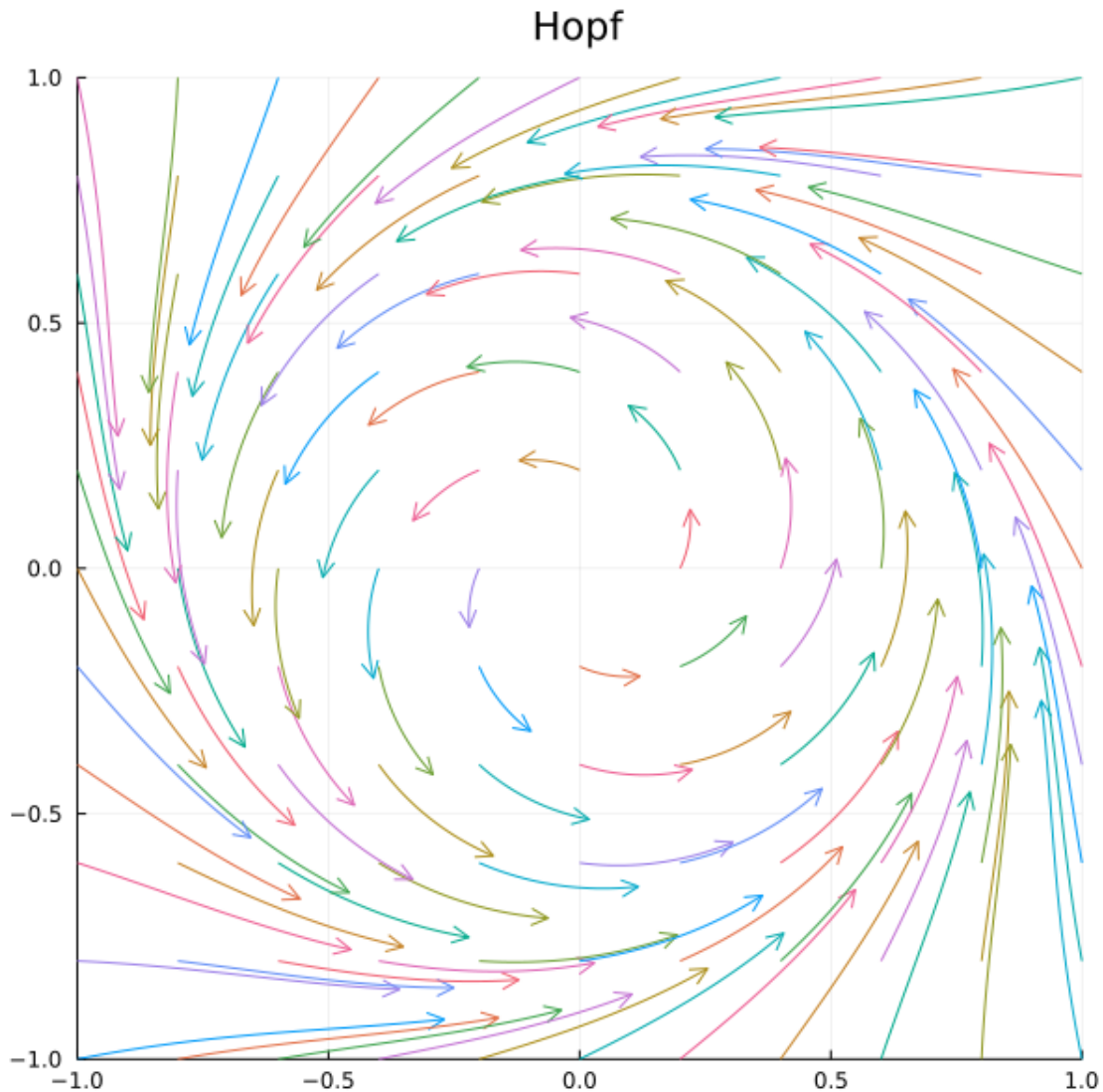
```
[32]: # Esto es interactivo en ipynb
xini=0.2;yini=0.0;tmax=50.0;=1.0;=0.3
#@manipulate throttle=.05 for xini=-1:0.1:1,yini=-1:0.1:1,tmax=10:10:50,=0.1:0.
↪1:1,=-0.5:.02:0.5
    prob = ODEProblem(hopf!, [xini; yini], (0, tmax), [], [])
    sol = solve(prob);
    p1 = plot(sol)
    p2 = plot(sol,vars=(1,2))
    scatter!(p2,[prob.u0[1]], [prob.u0[2]])
    plot(p1,p2,layout=(1,2),size=(900,450),title="Hopf",fmt=:png)
end
```

[32]:



```
[14]: # multiples trayectorias
u0_arr = vec([0.2*i-1,0.2*j-1 for i=0:10, j=0:10])
prob = ODEProblem(hopf!,u0_arr[1],(0,0.5),[0.5,1.0])
ensemble_prob = EnsembleProblem(prob,prob_func=(prob,i,repeat;
↪u0=u0_arr)->(remake(prob,u0=u0[i])))
sol = solve(ensemble_prob,EnsembleThreads(),trajectories=length(u0_arr))
plot(sol,vars=(1,2),arrow=true,size=(600,600),title="Hopf",fmt=:png)
```

[14]:

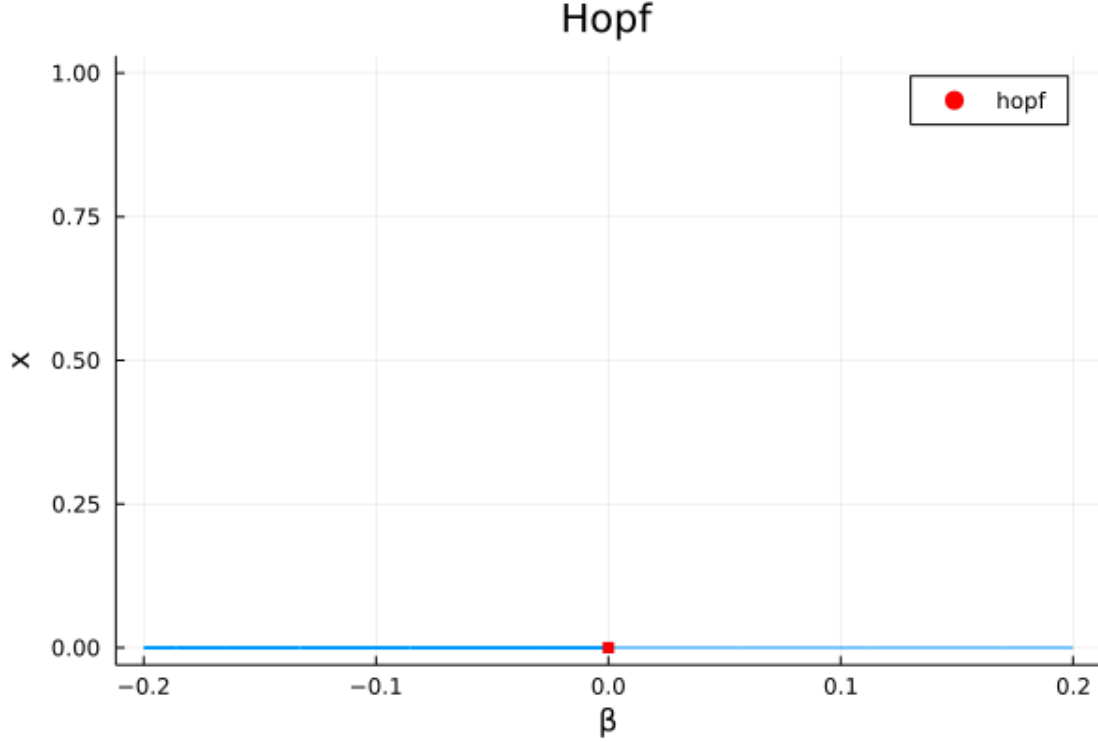


3.0.1 Detección de la Hopf usando BifurcationKit

Vamos a usar BifurcationKit para detectar esta bifurcación. Este caso es relativamente sencillo, ya que el punto fijo existe a ambos lados de la bifurcación y tenemos una sola rama. Notar que, a diferencia de los casos anteriores en los que marcaba los puntos de bifurcación en el diagrama únicamente como “bp” (bifurcation point), ahora llevan la etiqueta de una Hopf.

```
[16]: hopf(u,p) = hopf!(similar(u),u,p,0)
      opts = ContinuationPar(pMin=-0.2,pMax=0.2, ds = 0.01, dsmax = 0.
      ↪04,detectBifurcation=3,nInversion=8)
      br1, = continuation(hopf,[0.1,0.1],[-0.2,1.0],(@lens _[1]), opts)
      plot(br1,xlabel="\\beta",title="Hopf",fmt=:png)
```

[16]:



3.1 Forma normal de la Hopf en polares

Podemos reescribir la forma normal de Hopf como:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \beta & -\omega \\ \omega & \beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - (x^2 + y^2) \begin{pmatrix} x \\ y \end{pmatrix}$$

Vamos a cambiar ahora a coordenadas polares:

$$\rho = \sqrt{x^2 + y^2}, \quad \theta = \arctan\left(\frac{y}{x}\right)$$

con la transformación inversa:

$$x = \rho \cos(\theta), \quad y = \rho \sin(\theta)$$

Luego de operar algebraicamente y usando extensivamente la relación $\cos^2(\theta) + \sin^2(\theta) = 1$ llegamos a un sistema mucho más simple en coordenadas polares:

$$\dot{\rho} = \beta\rho - \rho^3$$

$$\dot{\theta} = \omega$$

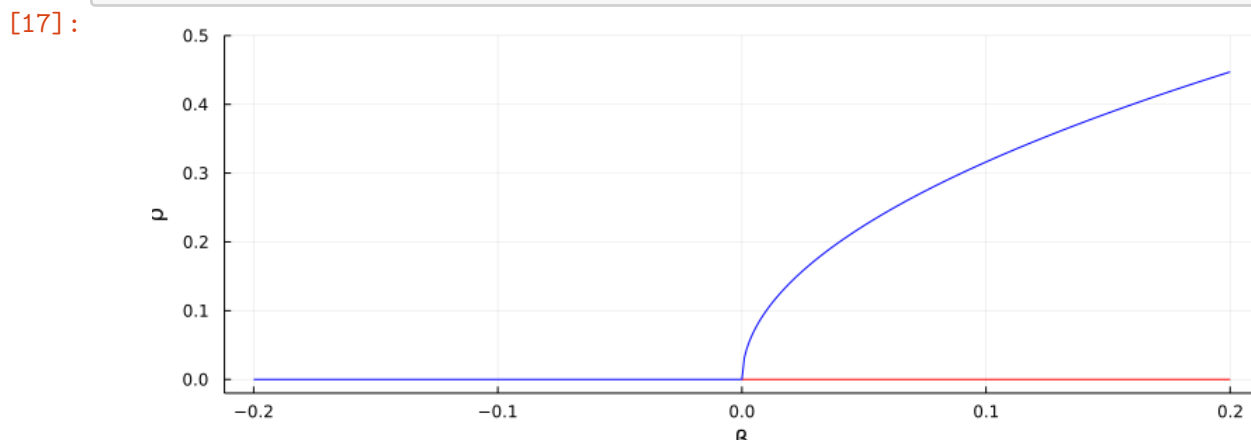
que es la forma normal de la bifurcación de Hopf en coordenadas polares.

Lo notable es que la segunda ecuación (la variación angular) es invariante y lo único que indica es que el flujo gira en torno al origen con velocidad angular constante ω (algo que se puede verificar en el flujo). Mientras que la primera ecuación para la variación en el radio tiene la misma forma que la bifurcación de pitchfork en 1D! Si bien hay que tener en cuenta que por definición $\rho \geq 0$.

Por lo tanto, al igual que en la pitchfork supercrítica vamos a tener, para $\beta < 0$ un atractor en el origen en la dirección radial (que junto con la variación angular a velocidad constante dan lugar a una espiral atractora), mientras que para $\beta > 0$ el punto fijo en $\rho = 0$ se vuelve repulsor y aparece un punto fijo atractor en $\rho = \sqrt{\beta}$ que corresponde a un ciclo límite atractor con radio constante.

El diagrama de bifurcaciones en la coordenada radial quedaría entonces para la Hopf (atención que lo hacemos a mano y no usando BifurcationKit) :

```
[17]: neg = -0.2:0.001:0.0; pos = 0.0:0.001:0.2
      2 = sqrt.(pos)
      plot(neg, 0*neg, c=:blue, legend=false)
      plot!(pos, 0*pos, c=:red)
      plot!(pos, 2, c=:blue, ylims=(-0.02, 0.
      ↪5), xlabel="\beta", ylabel="\rho", size=(800, 300), fmt=:png)
```



3.1.1 Amplitud de la oscilación de Hopf

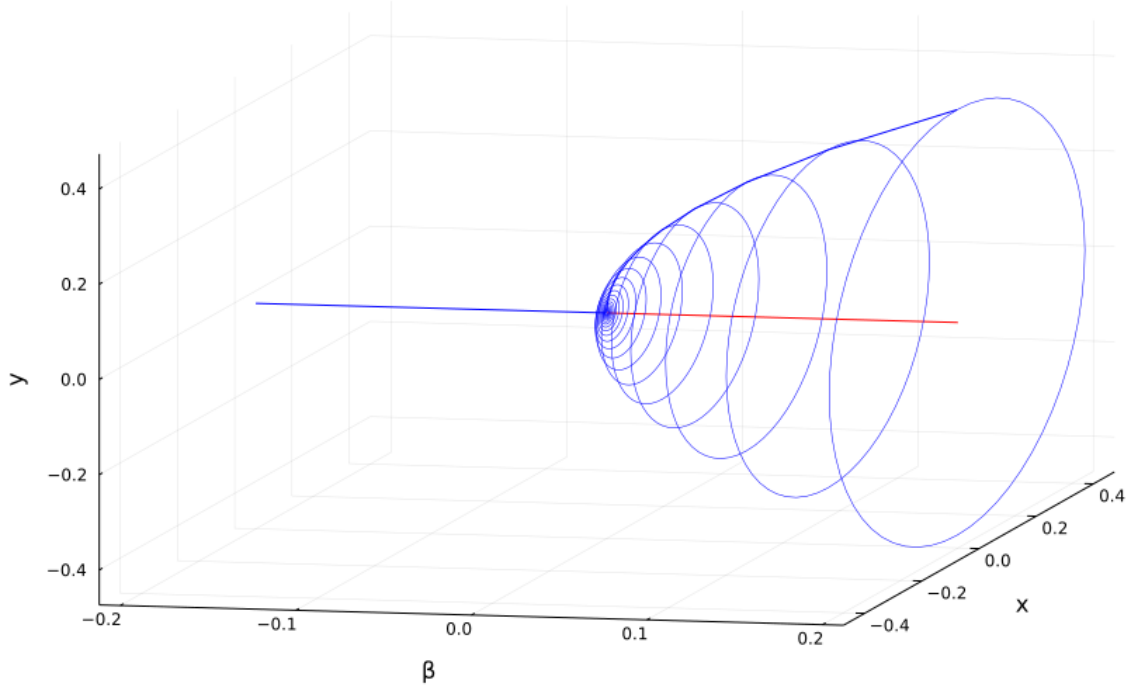
Recordamos que habíamos definido el diagrama de bifurcaciones en el espacio producto del (o los) parámetro(s) y las variables y graficábamos los conjuntos límite (hasta ahora puntos fijos) como ramas (curvas) coloréandolos según su estabilidad.

En el caso de la Hopf, al tratarse de una bifurcación que involucra si o si dos variables (no tenemos como en el caso de la SN la relajación a una variedad unidimensional porque la variedad sobre la que ocurre la Hopf tiene que ser bidimensional), nuestro diagrama de bifurcaciones tendría que ser 3D, graficando los conjuntos límite en el espacio de fases (x, y) en función de nuestro parámetro de bifurcación β .

Afortunadamente en el caso de la forma normal de Hopf es muy sencillo porque en coordenadas polares ya tenemos el diagrama de bifurcaciones en la variable radial y la variable angular lo único que hace es rotar el diagrama anterior en torno al origen. Como resultado, la rama estable de $\rho = \sqrt{\beta}$ determina un ciclo límite atractor cuya amplitud crece con la raíz cuadrada del parámetro y cuya velocidad angular ($\dot{\theta}$) es exactamente igual a 1:


```
[18]: list = 0: /30:2*
      list = 10 .^(-4.7:0.2:-0.7)
      p1 = plot( neg,0* neg,0* neg,c=:blue,legend=false)
      plot!(p1, pos,0* pos,0* neg,c=:red)
      plot!(p1, list,0* list,sqrt.( list),c=:blue)
      for in list
          = sqrt( )
          plot!(p1, .+0* list, *cos.( list), *sin.( list),c=:blue,linewidth=0.5)
      end
      plot(p1;
           ↪xlabel="\\beta",ylabel="x",zlabel="y",size=(800,500),camera=(20,35),fmt=:png)
```

[18]:



3.1.2 Frecuencia de la oscilación de Hopf

Volviendo a la forma normal de la Hopf:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \beta & -\omega \\ \omega & \beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} - (x^2 + y^2) \begin{pmatrix} x \\ y \end{pmatrix}$$

Si calculamos los autovalores de la parte lineal resulta que en la bifurcación ω corresponde a la parte imaginaria del autovalor y β es la parte real que se anula. En efecto de la matriz lineal tenemos que el determinante es $\Delta = \beta^2 + \omega^2$ y la traza es $\tau = 2\beta$, por lo tanto:

$$\lambda_{\pm} = \frac{\tau \pm \sqrt{\tau^2 - 4\Delta}}{2} = \beta \pm \sqrt{-\omega^2} = \beta \pm i\omega$$

En la bifurcación $\beta = 0$ y los autovalores son puramente imaginarios

$$\lambda_{\pm} = \pm i\omega$$

Cuando pasamos a coordenadas polares resulta evidente el rol que cumple ω :

$$\dot{\rho} = \beta\rho - \rho^3$$

$$\dot{\theta} = \omega$$

es decir que es precisamente la frecuencia angular del ciclo límite que surge en la Hopf.

Es importante notar que esta separación de variables puede hacerse para la forma normal pero no de forma general. Sin embargo, al igual que con el resto de las bifurcaciones, **todo sistema genérico que anule la parte real de un par de autovalores conjugados es topológicamente equivalente a la forma normal de la Hopf**, por lo tanto para esos sistemas también tendremos el nacimiento de un **ciclo límite que aparece con amplitud proporcional a la raíz cuadrada del parámetro de bifurcación y con una frecuencia de oscilación aproximada por la parte imaginaria del autovalor**.

3.1.3 Transformación de polares a cartesianas

Cuando pasamos la forma de la Hopf a polares evitamos los pasos intermedios porque son bastante engorrosos. Sin embargo ir en el sentido inverso es bastante sencillo y nos va a servir luego para estudiar modelos simples de bifurcaciones de ciclos límites. Supongamos que partimos de un sistema escrito en coordenadas polares especificando $(\dot{\rho}, \dot{\theta})$. Como pasar a las ecuaciones diferenciales escritas en cartesianas (\dot{x}, \dot{y}) ? Partimos de:

$$x = \rho \cos(\theta),$$

$$y = \rho \sin(\theta)$$

Tenemos que derivar estas expresiones respecto al tiempo (usando la regla de la cadena y la derivada del producto:

$$\dot{x} = \dot{\rho} \cos(\theta) - \rho \sin(\theta) \dot{\theta} = \frac{\dot{\rho}x}{\rho} - \frac{\rho \dot{\theta}y}{\rho}$$

$$\dot{y} = \dot{\rho} \sin(\theta) + \rho \cos(\theta) \dot{\theta} = \frac{\dot{\rho}y}{\rho} + \frac{\rho \dot{\theta}x}{\rho}$$

donde en el ultimo paso usamos la conversión de variables para sacarnos de encima los senos y cosenos. Vamos a suponer ahora por simplicidad que la dinámica ocurre solo en la variable radial y que $\dot{\theta} = \omega$. De esta forma nuestro sistema de ecuaciones queda expresado de una forma bastante elegante como:

$$\dot{x} = -\omega y + x(\dot{\rho}/\rho)$$

$$\dot{y} = \omega x + y(\dot{\rho}/\rho)$$

Dependiendo de la elección de $\dot{\rho}$ vamos a tener diferentes dinámicas y eso se va a poder trasladar fácil a un sistema (\dot{x}, \dot{y}) . Por ejemplo si reemplazamos $\dot{\rho} = \beta\rho - \rho^3$:

$$\dot{x} = -\omega y + \beta x - x\rho^2$$

$$\dot{y} = \omega x + \beta y - y\rho^2$$

que es la expresión original de la Hopf ($\rho^2 = x^2 + y^2$).

Como ejemplo simple podemos usar $\dot{\rho} = \beta\rho + \rho^3$ que corresponde a la bifurcación de Pitchfork **subcrítica** en el que se crean puntos fijos para $\beta < 0$ y son inestables. Si reemplazamos obtenemos:

$$\dot{x} = -\omega y + \beta x + x\rho^2$$

$$\dot{y} = \omega x + \beta y + y\rho^2$$

donde lo unico que cambio es el signo del término no lineal y corresponde a la forma normal de una bifurcación de **Hopf subcritica** donde el ciclo límite es creado para $\beta < 0$, es repulsor y convive con un foco atractor, con el cual colapsa a $\beta = 0$ convirtiéndolo en un foco repulsor.

3.2 Ejemplos simples de bifurcaciones de Hopf

3.2.1 Lengüeta de Rayleigh

Como un ejemplo simple consideremos el modelo de Rayleigh de la lengüeta:

$$\dot{x} = y$$

$$\dot{y} = \mu y - y^3 - kx$$

con un único punto fijo en el origen. El Jacobiano de forma general es:

$$\begin{pmatrix} 0 & 1 \\ -k & \mu - 3y^2 \end{pmatrix}$$

evaluado en el punto fijo $x = 0, y = 0$ es

$$\begin{pmatrix} 0 & 1 \\ -k & \mu \end{pmatrix}$$

que tiene determinante siempre positivo $\Delta = k$ y la traza igual al parámetro μ . El punto fijo es atractor para $\mu < 0$ y repulsor para $\mu > 0$. Cuando $\mu = 0$ la traza se anula y los autovalores son puramente imaginarios:

$$\lambda_1 = \frac{\tau \pm \sqrt{\tau^2 - 4\Delta}}{2} = \pm \frac{\sqrt{-4k}}{2} = \pm i\sqrt{k}$$

Tenemos una bifurcación de Hopf y encontramos la relación entre el parámetro de elasticidad k y la frecuencia de oscilación en la bifurcación: $k = \omega^2$ al igual que en el oscilador armónico. Sabemos además que la amplitud de las oscilaciones va a crecer de forma suave al principio como $\sqrt{\mu}$

3.2.2 van der Pol sesgado

Hay una condición más para que se produzca la bifurcación de Hopf y es que en la bifurcación persistan términos no lineales (en realidad están vinculados con la segunda derivada del campo vector pero no vamos a entrar en ese detalle técnico).

Por ese motivo en el oscilador de van der Pol no tenemos una bifurcación de Hopf, si bien cuando $\mu = 0$ los autovalores tienen parte real cero, ya que para este valor de parámetro también se anula el único no lineal $-\mu x^2 y$. Como consecuencia, no hay ciclos límites “pequeños” cerca de la bifurcación sino que el ciclo límite ya nace “grande” como una oscilación de relajación.

Sin embargo, basta con perturbar el sistema con una variable constante (lo que se conoce como sistema de van der Pol sesgado) y tenemos una bifurcación de Hopf:

$$\dot{x} = y$$

$$\dot{y} = \mu(1 - x^2)y - x + a$$

Este sesgo corre el punto fijo del origen al punto $(x_*, y_*) = (a, 0)$. Notar que este corrimiento no es un mero cambio de coordenadas porque afecta al flujo global. Esto último se puede probar con un cambio de coordenadas llevando el punto fijo al origen (definiendo una nueva variable $x_1 = x - a$) y viendo que aparecen otros términos no lineales y que la dependencia con la perturbación a queda incorporada en el parámetro para el cual se produce la bifurcación. Pero vamos a ver lo mismo en el sistema escrito ahí arriba de forma más simple. La matriz Jacobiana queda de forma general:

$$\begin{pmatrix} 0 & 1 \\ -1 - 2\mu xy & \mu(1 - x^2) \end{pmatrix}$$

que evaluada con $x = a$ y $y = 0$ resulta en la matriz de linealización en un entorno del punto fijo:

$$\begin{pmatrix} 0 & 1 \\ -1 & \mu(1 - a^2) \end{pmatrix}$$

Como en el caso anterior el determinante es siempre positivo $\Delta = 1$ pero la traza incluye a ambos parámetros $\mu(1 - a^2)$. Entonces haciendo $\mu \neq 0$ evadimos la degeneración del van der Pol original y obtenemos una bifurcación de Hopf cuando a cruza el valor 1 o -1.

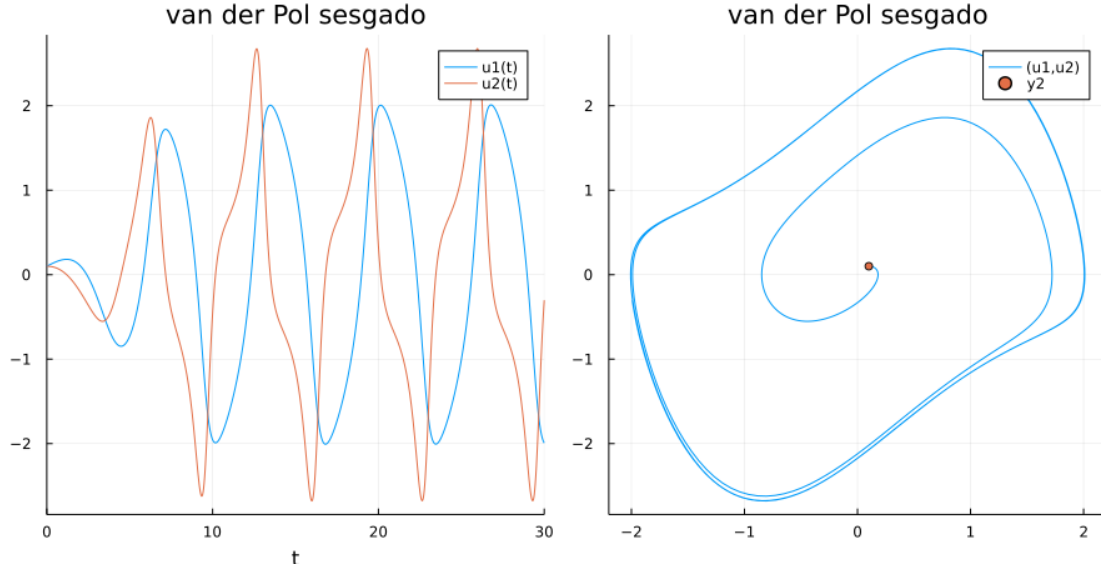
Esto se puede explorar en el gráfico interactivo de abajo

```
[19]: function vdp_biased!(du,u,p,t)
      ( ,a) = p
      du[1] = u[2]
      du[2] = *(1.0-u[1]*u[1])*u[2]-u[1]+a
      du
    end
```

```
[19]: vdp_biased! (generic function with 1 method)
```

```
[33]: # Esto es interactivo en ipynb
xini=0.1;yini=0.1;tmax=30.0;a=0;
#@manipulate throttle=.05 for xini=-1:0.1:1,yini=-1:0.1:1,tmax=10:10:50,a=-1.5:
↪0.01:1.5
    prob = ODEProblem(vdp_biased!, [xini; yini], (0, tmax), [1.0,a])
    sol = solve(prob);
    p1 = plot(sol)
    p2 = plot(sol,vars=(1,2))
    scatter!(p2,[prob.u0[1]], [prob.u0[2]])
    plot(p1,p2,layout=(1,2),size=(900,450),title="van der Pol sesgado",fmt=:png)
  end
```

```
[33]:
```



3.3 Bifurcación de Hopf en Modelo de Predador Presa

Vamos a ver un ejemplo de sistema que sufre una bifurcación de Hopf supercrítica y da origen a autooscilaciones. Se trata de un modelo poblacional de dos especies, una que predica a la otra. A diferencia del modelo de Lotka-Volterra que no presentaba **estabilidad estructural** en este caso para ciertos valores de parámetros vamos a tener un ciclo límite atractor que va a dar lugar a oscilaciones robustas en ambas poblaciones: si el sistema está oscilando en este ciclo y realizamos una perturbación razonable que no lleve a la extinción luego de un tiempo el sistema vuelve a la misma órbita periódica (y no salta de órbita como pasaba en el LV).

3.3.1 [OPCIONAL] descripción y adimensionalización del modelo

Podemos plantear un modelo de Predador-Presa más realista y ver si podemos obtener auto oscilaciones. Para empezar podemos reemplazar el término de crecimiento exponencial bx de la presa por un crecimiento logístico con una capacidad K . Luego en lugar de asumir que el impacto de la predación en la población de la presa es siempre proporcional al producto de la presa por el predador podemos asumir que satura en un momento determinado (el predador no es insaciable). Si N es la población de la presa y P la del predador la ecuación para la variación de la población de la presa sería:

$$\dot{N} = RN \left(1 - \frac{N}{K}\right) - PNf(N)$$

Con $f(N)$ una función que satura, vamos a tomar una forma simple $f(N) = A/(N + C)$.

Por otro lado podemos suponer simplemente para la población del predador P un crecimiento logístico donde la capacidad es proporcional a la población de la presa con lo cual la segunda ecuación quedaría:

$$\dot{P} = SP \left(1 - \frac{P}{BN}\right)$$

Tenemos seis parametros (R, S, K, A, B, C) , pero se pueden reducir si adimensionalizamos las poblaciones: $x = N/K$ y $y = P/(BK)$. Al principio las ecuaciones se complican (dividimos la de arriba por K y la de abajo por BK):

$$\frac{d(N/K)}{dt} = \frac{RN}{K} \left(1 - \frac{N}{K}\right) - \frac{PAN}{K^2(N/K + C/K)}$$

$$\frac{d(P/(BK))}{dt} = \frac{SP}{BK} \left(1 - \frac{P}{BN}\right)$$

reemplazando por la definición de las variables adimensionalizadas, dividiendo ambas ecuaciones por R y reescalando el tiempo por $\tau = tR$:

$$\frac{dx}{d\tau} = x(1 - x) - \frac{ABxy}{R(x + C/K)}$$

$$\frac{dy}{d\tau} = \frac{S}{R}y \left(1 - \frac{y}{x}\right)$$

si redefinimos parametros: $a = AB/R$, $b = S/R$, $c = C/K$ las ecuaciones quedan escritas de una forma mas compacta (ahora la derivada temporal es con respecto a τ).

3.4 Modelo adimensionalizado

El modelo simplificado de las poblaciones adimensionalizadas de presa (x) y predador (y) queda entonces:

$$\dot{x} = x(1 - x) - \frac{axy}{x + c}$$

$$\dot{y} = by \left(1 - \frac{y}{x}\right)$$

que depende solo de tres parametros a, b, c . La contra es que ahora los parametros son mas dificiles de interpretar. Por ejemplo b representa como es la tasa de crecimiento de la poblacion de predadores **relativa** a la tasa de crecimiento de la poblacion de presas, c esta vinculada a la fraccion de la poblacion maxima de presa a la que satura la predacion, pero a es mas dificil de interpretar ya que crece tanto con la accion de la predacion como con la capacidad de la poblacion de los predadores relativa a la capacidad de la poblacion de la presa.

Para un valor fijo de $a > 0.5$ las dos poblaciones llegan a un punto fijo con un mismo valor $x = y$ para valores de b y c altos, pero si esto parametros disminuyen las poblaciones comienzan a oscilar. A diferencia del modelo de Lotka-Volterra, esta oscilacion es estructuralmente estable, todas las trayectorias convergen al **ciclo limite**.

Si bien las ecuaciones parecen mas complicadas las nulclinas no lo son tanto restringidas al dominio no negativo. Ademas de una nulclina en $y = 0$ tenemos la recta $y = x$ para la segunda nulclina (linea diagonal verde) y una cuadratica que sale de la primera ecuacion que puede despejarse como:

$$y = (x + c)(1 - x)/a$$

que es una parabola invertida que corta el eje horizontal en los puntos $x = 1$, $x = -c$ (fuera del dominio), y corta el eje vertical en $y = c/a$

En el dominio tenemos dos puntos fijos:

- donde se corta la parabola con el eje horizontal $x = 1, y = 0$ que es siempre un punto silla (esto se puede ver calculando el Jacobiano que en ese punto fijo tiene determinante igual a $-b$)
- donde se corta la parabola con la diagonal $x = y$ vamos a calcular la ubicacion de es punto fijo que es el que da lugar a la bifurcacion de Hopf

Haciendo $y = x$ en la ecuacion del a nulclina de la parabola y tomando solo la solucion positiva dentro del dominio tenemos:

$$x_* = \frac{1-a-c}{2} + \sqrt{\left(\frac{1-a-c}{2}\right)^2 + c}$$

aun cuando $1 - a - c$ es negativo la raiz es mayor en modulo por lo tanto siempre hay una raiz positiva.

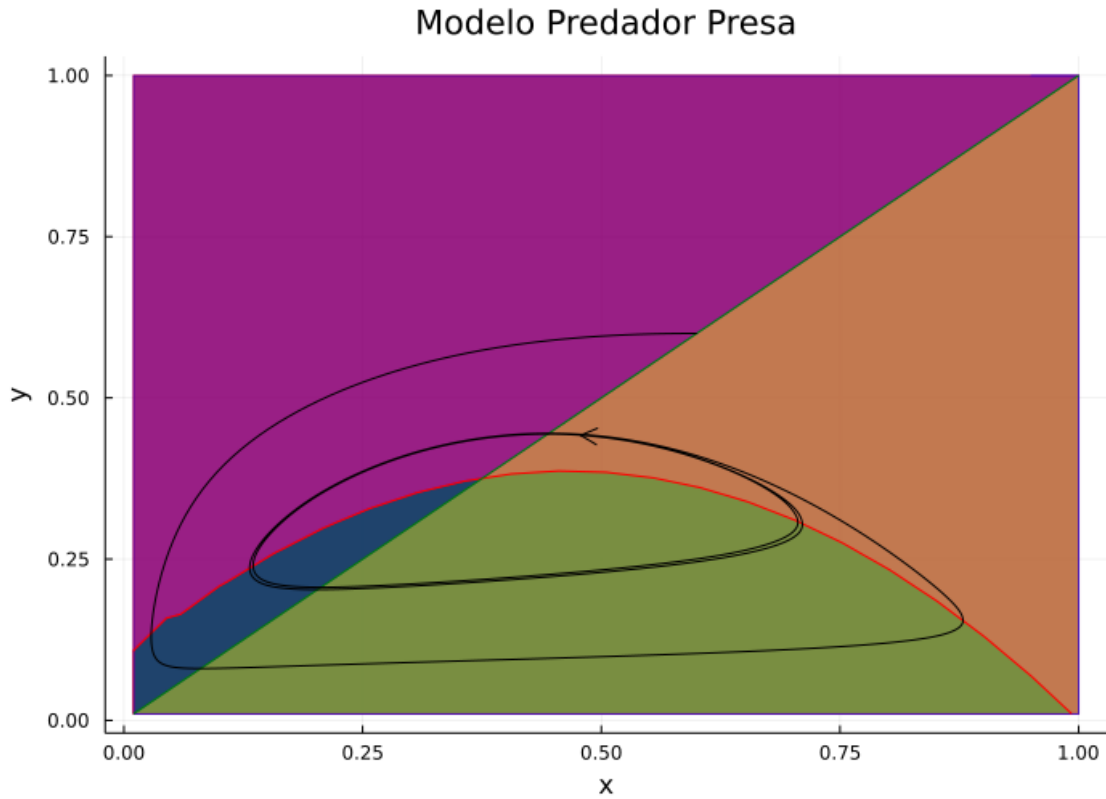
Notar que tanto el punto fijo como las nulclinas no dependen del parametro b sin embargo este si influencia en la estabilidad.

```
[21]: function predatorprey!(du,u,p,t)
      (a,b,c) = p
      du[1] = u[1]*((1.0-u[1]) - a*u[2]/(c+u[1]))
      du[2] = b*u[2]*(1.0-u[2]/u[1])
      du
    end
```

```
[21]: predatorprey! (generic function with 1 method)
```

```
[34]: # Esto es interactivo en ipynb
xini=0.6;yini=0.6;a=0.74;b=0.11;c=0.07
#@manipulate throttle=0.2 for xini=0:0.1:1, yini=0:0.1:1,a=0.01:0.01:2,b=0.01:0.
↪01:1,c=0.01:0.01:1
      flux2d_nullclines(predatorprey!, [xini;yini], 100.0, [a,b,c];
        xlims=[0.01,1.0], ylims=[0.01,1.0], title="Modelo Predador Presa")
    #end
```

```
[34]:
```



3.5 Estabilidad de los puntos fijos

Calcular el jacobiano y evaluarlo los puntos fijos puede ser un poco arduo pero fijando $a = 1$ y luego de algunas cuentas el Jacobiano queda expresado:

$$\begin{pmatrix} 1 - 2x - \frac{cy}{(x+c)^2} & -\frac{x}{x+c} \\ b\frac{y^2}{x^2} & -b \end{pmatrix}$$

el punto fijo es ahora

$$x_* = -\frac{c}{2} + \sqrt{\frac{c^2}{4} + c}$$

tras un poco de algebra la matriz Jacobiana queda:

$$\begin{pmatrix} \frac{c^2}{2} + \frac{5c}{2} + 1 - (c+3)\sqrt{\frac{c^2}{4} + c} & \sqrt{\frac{c^2}{4} + c} - \frac{c}{2} - 1 \\ b & -b \end{pmatrix}$$

El determinante queda un poco mas simple:

$$\Delta = b \left((c+2)\sqrt{\frac{c^2}{4} + c} - \frac{c^2}{2} - 2c \right)$$

que con un poco de trabajo (o graficando) se puede ver que es siempre negativo.

Mientras que la traza puede ser positiva o negativa y se hace cero cuando (condicion de la bifurcacion de Hopf):

$$b = \frac{c^2}{2} + \frac{5}{2}c + 1 - (c + 3)\sqrt{\frac{c^2}{4} + c}$$

En la seccion siguiente vamos a ver el significado de esa condicion expresada como b en funcion de c que determina una curva en el espacio de parametros (b, c) .

4 Codimension de una bifurcacion

Si bien en el modelo anterior tuvimos que mover tres parametros para buscar la bifurcacion de Hopf (probar por ejemplo con $a = 1$, $b = 0.25$ y bajar el parametro c cruzando el valor 0.1), en realidad la bifurcacion de Hopf, al igual que la de nodo-silla, se cruza moviendo un solo parametro. Mas adelante vamos a ver bifurcaciones donde es necesario sintonizar dos parametros de forma simultanea para “pegarle” a la bifurcacion.

De manera terriblemente informal, la cantidad de parametros que es necesaria ajustar de forma simultanea para cruzar una bifurcacion corresponde a la **codimension** de la bifurcacion (hay una definicion mas rigurosa pero no es relevante ahora). Todas las bifurcaciones que vimos hasta ahora son de codimension 1.

Por ejemplo en el caso anterior podemos fijar $a = 1$, $b = 0.25$ y buscar la bifurcacion de Hopf usando solo el parametro c . En este caso nuestro “espacio de parametros” es 1D. Pero tambien podemos fijar a y c y usar b (queda como ejercicio la otra posibilidad de usar a como parametro de la bifurcacion).

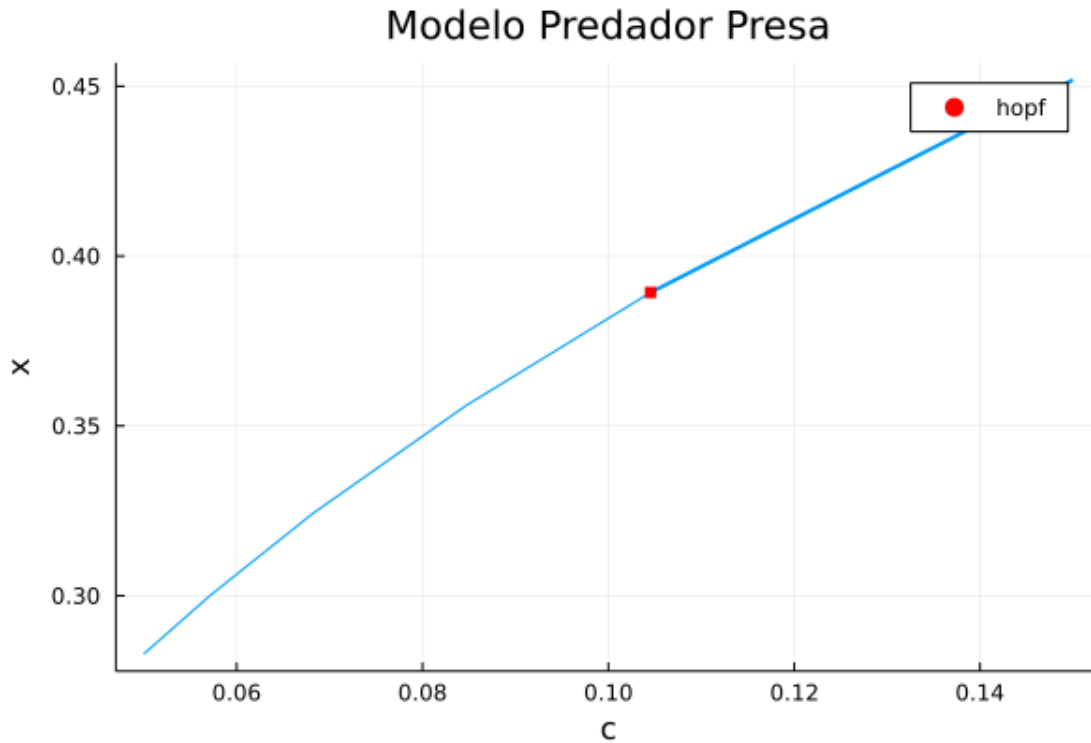
Que sucede si fijamos $a = 1$ y usamos un espacio de parametros 2D formado por los restantes parametros?

Esto es absolutamente legitimo. En ese caso la bifurcacion de Hopf no ocurre en un punto sino en una curva en ese espacio bidimensional que se puede cruzar en ambas direcciones (es decir no es paralela a uno de los ejes). Este tipo de diagrama de bifurcación es útil porque si bien la bifurcacion puede ser de codimension 1 nos puede interesar como esa bifurcacion depende de dos parametros de control. La curva que derivamos en la seccion anterior como condicion de Hopf (la traza nula con el determinante positivo), corresponde exactamente a esa curva, pero vamos a usar primero el paquete BifurcationKit para calcular numericamente la bifurcacion.

Vamos a estudiar primero la bifurcacion en un espacio de parametros 1D (variando solo c) y luego vamos a tratar de esbozar el diagrama en el espacio (b, c)

```
[23]: predatorprey(u,p) = predatorprey!(similar(u),u,p,0)
      opts = ContinuationPar(pMin=0.05,pMax=0.15, ds = 0.01, dsmax = 0.
      ↪04,detectBifurcation=3,nInversion=8)
      br1, = continuation(predatorprey,[0.3,0.3],[1.0,0.25,0.05],(@lens _[3]), opts)
      plot(br1,xlabel="c",title="Modelo Predador Presa",fmt=:png)
```

[23]:



La rama del foco (br) además de poder graficarse tiene la información del punto especial hopf. Los puntos especiales que encuentra en la rama los guarda en un array `specialpoint`

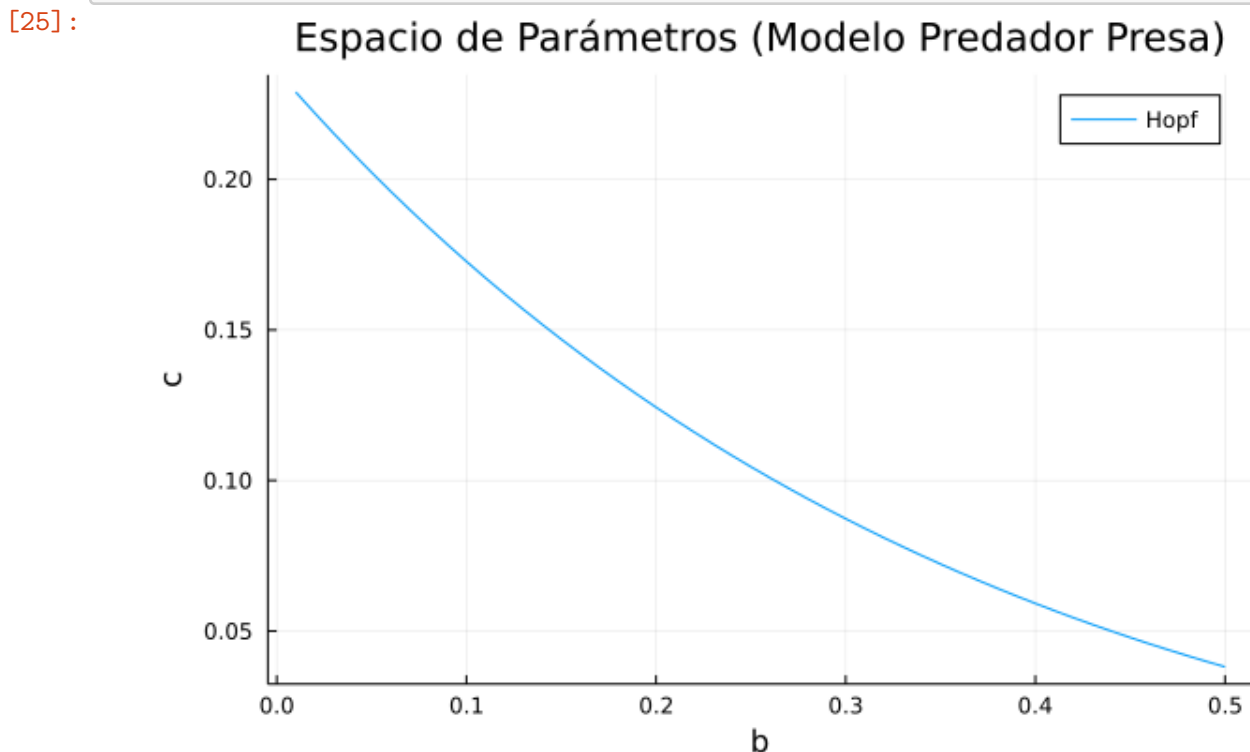
```
[24]: br1.specialpoint
```

```
[24]: 1-element Vector{BifurcationKit.SpecialPoint{Float64, NamedTuple{(:x,),
Tuple{Float64}}, Vector{Float64}}}:
  BifurcationKit.SpecialPoint{Float64, NamedTuple{(:x,), Tuple{Float64}},
Vector{Float64}}
    type: Symbol hopf
    idx: Int64 5
    param: Float64 0.10454184616960667
    norm: Float64 0.3892712221834026
    printsol: NamedTuple{(:x,), Tuple{Float64}}
    x: Array{Float64}((2,)) [0.2752563209266592, 0.2752563209266592]
      : BorderedArray{Vector{Float64}, Float64}
    ind_ev: Int64 2
    step: Int64 4
    status: Symbol guess
      : Tuple{Int64, Int64}
    precision: Float64 1.0994387272256212e-6
    interval: Tuple{Float64, Float64}
```

que en este caso tiene un solo elemento. Ese punto tiene el valor del parametro que se barrio (c) para el cual se produce la bifurcacion almacenado en `param`. Por lo tanto ahora podemos barrer el parametro b en un loop y para cada valor de b calculamos la rama igual que antes y extraemos el valor de c que corresponde a la bifurcacion con `br1.specialpoint[1].param`.

De esta forma podemos armar un array de valores de c (`clist`) para los cuales el sistema bifurca, en funcion de los valores de b (`blist`) y eso es una curva en el espacio de parámetros (b, c) que corresponde a la bifurcación de Hopf. Recordar que a esta fijo en el valor 1.

```
[25]: blist = [0.01*n for n in 1:50]
      opts = ContinuationPar(pMin=0.15,pMax=0.25, ds = 0.01, dsmax = 0.
      ↪04,detectBifurcation=3,nInversion=8)
      br1, = continuation(predatorprey,[0.4,0.4],[1.0,blist[1],0.15],(@lens _[3]),
      ↪opts)
      c = br1.specialpoint[1].param
      clist = [c]
      for n=2:50
          opts = ContinuationPar(pMin=0.8*c,pMax=1.2*c, ds = 0.01, dsmax = 0.
          ↪04,detectBifurcation=3,nInversion=8)
          br1, = continuation(predatorprey,[0.4,0.4],[1.0,blist[n],0.8*c],(@lens _[3]),
          ↪opts)
          c = br1.specialpoint[1].param
          append!(clist,c)
      end
      plot(blist,clist,xlabel="b",ylabel="c",title="Espacio de Parámetros (Modelo
      ↪Predador Presa)",label="Hopf",fmt=:png)
```



Anteriormente deducimos una expresión teórica para la curva de Hopf $b = \frac{c^2}{2} + \frac{5}{2}c + 1 - (c+3)\sqrt{\frac{c^2}{4} + c}$ a partir de igualar a cero la traza del Jacobiano. Podemos comparar el valor teórico de la curva con el resultado numérico de BifurcationKit (BK). En la figura de abajo se muestra la curva teórica como una línea continua y los valores de BK con marcadores azules.

```
[26]: scatter(blist,clist,label="BK")
      bteo = @. 0.5*clist^2+2.5*clist+1-(clist+3)*sqrt(abs2(clist)/4+clist)
      plot!(bteo,clist,xlabel="b",ylabel="c",title="Espacio de Parámetros (Modelo_
      ↪Predador Presa)",label="Teorico",fmt=:png)
```

