

DNL_Bifurcaciones_Codimension2

May 4, 2022

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

WebIO._IJuliaInit()

1 Formas Normales, panorama general

Las bifurcaciones corresponden a cambios cualitativos en la dinámica, o la estructura topológica, de los sistemas dinámicos. Pero cuando nos referimos a “cambios” son cambios desde fuera, al modificar los parámetros del sistema modificamos el sistema dinámico.

Podemos adoptar un punto de vista más general para entender como se producen esos cambios si consideramos a un sistema dinámico como función de un parámetro (hasta ahora todas bifurcaciones que vimos las cruzábamos variando un único parámetro). En ese caso tenemos que considerar ahora los **campos vectores como función de las variables y el parámetro de bifurcación** (μ), es decir que vamos a escribir para flujos 1D:

$$\dot{x} = f(x, \mu)$$

Por ejemplo, para el caso de la forma normal de la pitchfork $f(x, \mu) = \mu x - x^3$.

Y vamos a suponer también que el punto $x = 0$ es un punto fijo y que la bifurcación se produce para el valor de parámetro $\mu = 0$ (esto siempre se puede hacer con un cambio lineal de variables). En ese caso tenemos para el punto fijo $f(0, 0) = 0$ las siguientes condiciones para las bifurcaciones que ocurren sobre una variedad unidimensional:

- la condición de que un autovalor del flujo linealizado se haga cero $\lambda = 0$, lo que equivale a decir que la derivada del campo vector respecto a la variable sea cero, es decir $\partial_x f(0, 0) = 0$. Notar que usamos la derivada parcial porque ahora hicimos depender nuestro campo vector del parámetro de forma explícita. Esta también se conoce como la condición de **no hiperbolicidad**. Recordar que los puntos hiperbólicos eran aquellos con autovalores estrictamente positivos o negativos.
- existen dos condiciones adicionales. La primera se llama condición de **transversalidad** y requiere que en la función $f(x, \mu)$ corte al plano horizontal $f = 0$ de forma transversal, es decir que no se anule también su derivada respecto del parámetro.
- la segunda es la condición de **no degeneración** y requiere que la derivada segunda de f (respecto de x) sea diferente de cero.

Si tomamos un punto fijo de sistema dinámico cualquiera, con un valor arbitrario de μ , **lo genérico es que sea hiperbólico**. También se dice que en un entorno de ese punto fijo el sistema es **robusto**

o **estructuralmente estable**. Si lo perturbamos en μ su dinámica no cambia de forma cualitativa. En ese caso la linealización me determina la dinámica en su entorno y no hay nada novedoso.

Las cosas interesantes empiezan a pasar con la pérdida de genericidad. La primera es la condición de **no hiperbolicidad**. En ese caso aparece la bifurcación. La condición no es genérica porque hay que sintonizar al parámetro para acertarle a la bifurcación. Y en ese punto el sistema no es estructuralmente estable, si lo perturbamos cambia la estructura topológica del flujo.

Si solo permitimos esa perdida de genericidad tenemos una bifurcación de **nodo silla**, siempre. Por eso se explicita que se deben cumplir otras dos condiciones de genericidad: la transversalidad y la no degeneración.

Pero podemos seguir perdiendo genericidad y ver que pasa. Si la posición del punto fijo no varía con el parámetro se viola la condición de transversalidad. Y entonces tenemos la bifurcación **transcrítica** (que es una bifurcación con un punto fijo que no se mueve en el espacio de parámetros).

Si ahora además imponemos una simetría en nuestro sistema, eso viola la condición de no degeneración porque anula la derivada segunda del campo vector, y tenemos la bifurcación de **pitchfork**.

Podemos entonces resumir lo que tenemos hasta ahora para las **bifurcaciones locales** en términos de las condiciones para que se produzcan y la forma normal correspondiente.

Bifurcacion	Descripcion	No hiperbolicidad	No degen- Transversalidad eración	Forma Normal
Saddle Node	un autovalor cero	$\partial_x f(0,0) = 0$	$\partial_\mu f(0,0) \neq 0$	$\dot{x} = \mu - x^2$
Transcritica	un autovalor cero	$\partial_x f(0,0) = 0$	$\partial_\mu f(0,0) = 0$	$\dot{x} = \mu x - x^2$
Pitchfork	un autovalor cero	$\partial_x f(0,0) = 0$	$\partial_\mu f(0,0) = 0$	$\dot{x} = \mu x - x^3$
Hopf	par de autovalores imaginarios	$\lambda_{1,2} = \pm i\omega$	$\partial_\mu \beta(0) \neq 0$	$\dot{\rho} = \beta \rho - \rho^3$

Agregamos tambien la bifurcacion de **Hopf**, si bien las condiciones de transversalidad y no de generacion son un poco diferentes. Si escribimos el autovalor como $\lambda_{1,2}(\mu) = \beta(\mu) \pm i\omega(\mu)$ donde μ es el parámetro de la bifurcación, la condición de transversalidad requiere que la derivada de la parte real del autovalor (en un sistema 2D es la traza del jacobiano) cruce el eje imaginario transversalmente. La condicion de no degeneración es mucho mas complicada de calcular ya que el parametro $l_1(\mu)$ depende de las derivadas de segundo y tercer orden del campo vector.

1.1 Normalizacion y Forma normal inducida.

Pero de donde sale la forma normal?

Recordamos que la forma normal para una bifurcación de un punto fijo es la forma más simple posible del campo vector, es decir con la menor cantidad posible de términos no lineales, que es topológicamente equivalente al flujo original. Esa forma es universal, en el sentido de que si logramos reducir el comportamiento de nuestro sistema a esa forma podemos saber que el flujo se va a comportar de una determinada manera en un entorno del punto fijo.

Pero por que aparecen algunos términos no lineales y no otros?

La respuesta a esa pregunta vamos a darla de forma totalmente informal (la respuesta formal excede este curso). Lo notable es que **es la parte lineal en la bifurcación la que determina cuales términos no lineales aparecen y cuáles no en la forma normal**.

Para llegar a la forma normal el procedimiento es aplicar sucesivos cambios de coordenadas (en general no lineales) para tratar de eliminar la mayor cantidad posible de términos no lineales. Como esos cambios de coordenadas no me alteran la estructura topológica del flujo en un entorno del punto fijo puedo hacer los que sean necesarios para cancelar por ejemplo términos cuadráticos o cúbicos si fuese posible. El procedimiento se conoce como **normalización**

Lo interesante es que si el punto es hiperbólico, podemos, mediante cambios de coordenadas, eliminar **todos** los términos no lineales (esto es equivalente a lo que vimos cuando linealizamos los flujos 2D y es una enunciación ligera del Teorema de Hartman Grobman).

Cuando el punto fijo es no hiperbolico, segun como sea la parte lineal, van a haber términos que no se van a poder eliminar, son los llamados términos **resonantes**. POr ejemplo para la saddle node x^2 es el término resonante, o para primera ecuacion de la Hopf los términos x^3 y xy^2 . Esos términos resonantes son los que hacen a la forma normal.

1.1.1 Ejemplo motivador en 1D [opcional]

El algebra de esta parte parece complicada pero en realidad no es más que aplicar distributiva con ganas y ordenar términos.

Supongamos que tenemos el siguiente sistema 1D con $\lambda \neq 0$:

$$\dot{x} = \lambda x + a_2 x^2 + a_3 x^3 + \dots$$

donde $x = 0$ es el punto fijo y es hiperbólico. Vamos a hacer un cambio de coordenadas “cercano a la identidad” con coeficientes a determinar:

$$x = y + b_2 y^2 + b_3 y^3 + \dots$$

queremos que nuestro sistema en la nueva variable y tenga la menor cantidad posible de términos no lineales y vamos a elegir a los coeficientes b para ello. Comenzamos derivando al expresion anterior con respecto al tiempo (usando la regla de la cadena y sacando factor comun \dot{y}) e igualandola a la ecuación de arriba reemplazando x por el polinomio en y :

$$\dot{x} = \dot{y}(1 + 2b_2 y + 3b_3 y^2 + \dots) = \lambda(y + b_2 y^2 + b_3 y^3 + \dots) + a_2 (y + b_2 y^2 + \dots)^2 + a_3 (y + b_2 y^2 + \dots)^3 + \dots$$

Despejamos \dot{y} y desarrollamos los terminos hasta orden 3 agrupandolos por potencia de y :

$$\dot{y} = \frac{\lambda y + y^2(\lambda b_2 + a_2) + y^3(\lambda b_3 + 2a_2 b_2 + a_3) + \dots}{(1 + 2b_2 y + 3b_3 y^2 + \dots)}$$

y ahora asumimos que $2b_2 y + 3b_3 y^2$ es pequeño cerca del origen y hacemos el desarrollo aproximado para un ϵ pequeño: $(1 + \epsilon)^{-1} \approx 1 - \epsilon$ y obtenemos, luego de distribuir y agrupar:

$$\dot{y} = \lambda y + (a_2 - \lambda b_2) y^2 + (a_3 - 2\lambda b_2^2 - 2\lambda b_3) y^3 + \dots$$

como tenemos la libertad de elegir b_2, b_3 , etc. y $\lambda \neq 0$, entonces podemos hacer $b_2 = a_2/\lambda$ y eliminamos el término cuadrático. Con ese valor de b_2 podemos elegir un valor de b_3 tal que anule

el paréntesis que acompaña al y^3 y no tenemos término cúbico. Y se puede proceder así eligiendo los sucesivos términos del cambio de coordenadas y eliminar todos los términos no lineales.

Con lo cual mostramos (en un caso simplificado 1D) que el sistema no lineal $\dot{x} = \lambda x + a_2 x^2 + a_3 x^3 + \dots$ con un punto fijo no hiperbólico en $x = 0$ es topológicamente equivalente a un sistema lineal $\dot{y} = \lambda y$.

Pero que pasa cuando el punto fijo pierde hiperbolicidad $\lambda = 0$? En ese caso **no podemos eliminar el término cuadrático y tenemos la bifurcación de saddle node**.

1.2 Codimensiones mayores

Volviendo a la tabla de bifurcaciones podríamos preguntarnos que pasaría si siguiésemos perdiendo genericidad. Por ejemplo si tenemos más de un autovalor que se hace cero de forma simultánea, o no se cumple la condición de no degeneración de la Hopf, o el término cuadrático de la Saddle Node se hace cero. Esas bifurcaciones con menos genericidad se conocen como Bogdanov-Takens, Bautin y Cusp respectivamente.

Pero, como vamos a ver, para poder cruzar estas bifurcaciones no basta mover un parámetro sino que se tienen que sintonizar dos parámetros de forma simultánea. Se trata de **bifurcaciones de codimensión 2**. Dicho muy llanamente, la codimensión de una bifurcación se corresponde con la cantidad de parámetros que hay que ajustar de forma simultánea para cruzarla. Todas las que vimos arriba son bifurcaciones de codimensión 1.

Si las bifurcaciones que vimos hasta ahora eran como paredes delgadas que separaban dos regiones topológicamente distintas, las bifurcaciones de codimensión 2 son como las aristas donde varias de estas paredes se juntan.

En efecto, ahora nuestro espacio de parámetros es bidimensional, y en ese espacio de parámetros las bifurcaciones de codimensión 1 son curvas y las de codimensión 2 son puntos.

Lo más interesante de estas bifurcaciones de codimensión mayor es que son como núcleos que organizan las bifurcaciones de codimensión menor, tanto locales como globales. Por ejemplo la Cusp es un punto en el espacio bidimensional de parámetros que da origen a dos bifurcaciones de saddle node, la Bautin a una Hopf supercrítica a una subcrítica y a una saddle node de ciclos límites (global); y la Bogdanov-Takens a una Hopf, una saddle node y a una bifurcación homoclínica.

Con las bifurcaciones de codimensión 2 entramos en un meta-nivel. Si antes estudiábamos las bifurcaciones de codimensión 1 para ver como estaba organizado topológicamente nuestro espacio de fases, con las bifurcaciones de codimensión 2 vamos a entender como está organizado nuestro espacio de parámetros con las distintas bifurcaciones.

2 Bifurcaciones de Codimension 2: Cusp

Para tener una bifurcación de codimensión 2 no es necesario que haya dos variables. El modelo de crecimiento logístico con outbreak presenta como habíamos visto dos bifurcaciones de nodo silla. Esto sucede de forma general si variamos el parámetro P para valores fijos de K y R . Pero en realidad habíamos tenido que ajustar K para que apareciesen esas bifurcaciones.

Que sucede para diferentes valores de K ? No siempre están esas dos bifurcaciones.

$$\dot{x} = Rx(1 - x/K) - P \frac{x^2}{1+x^2}$$

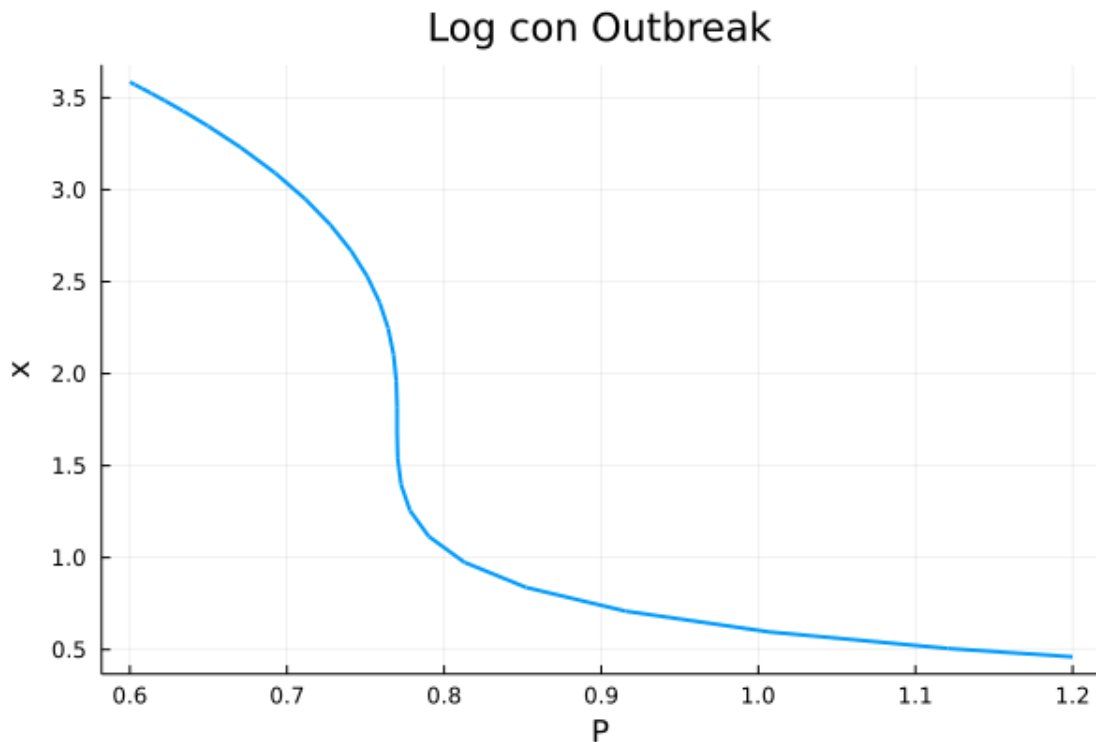
```
[2]: # Ecuacion Logistica con Outbreak
logoutbreak(x,p,t)=p[1]*x*(1.0-x/p[2])-p[3]*x*x/(1+x*x)
```

[2]: logoutbreak (generic function with 1 method)

Probar para valores de K entre 5 y 5.5

```
[6]: F=(x,p) -> @. logoutbreak(x,(p,),0)
opts = ContinuationPar(pMin=0.6,pMax=1.2, detectBifurcation=3)
#este grafico es interactivo en ipynb
K=5.2
#@manipulate throttle=0.1 for K=5.0:0.01:8.0
    br1, = continuation(F,[6.5],[0.5,K,0.6],(@lens _[3]), opts)
    plot(br1,xlabel="P",title="Log con Outbreak",fmt=:png)
#end
```

[6]:



Como se ve, para valores de K mayores a 5.2 (aproximadamente) tenemos los dos puntos de bifurcacion y hay una region en la que tenemos tres puntos fijos. Esa region se va haciendo mas estrecha en torno a $P \approx 0.77$ a medida que nos aproximamos a ese valor por arriba. Pero para $K = 0.52$ y valores menores los dos puntos de bifurcacion desaparecen y tenemos siempre un solo punto fijo para todo valor de P .

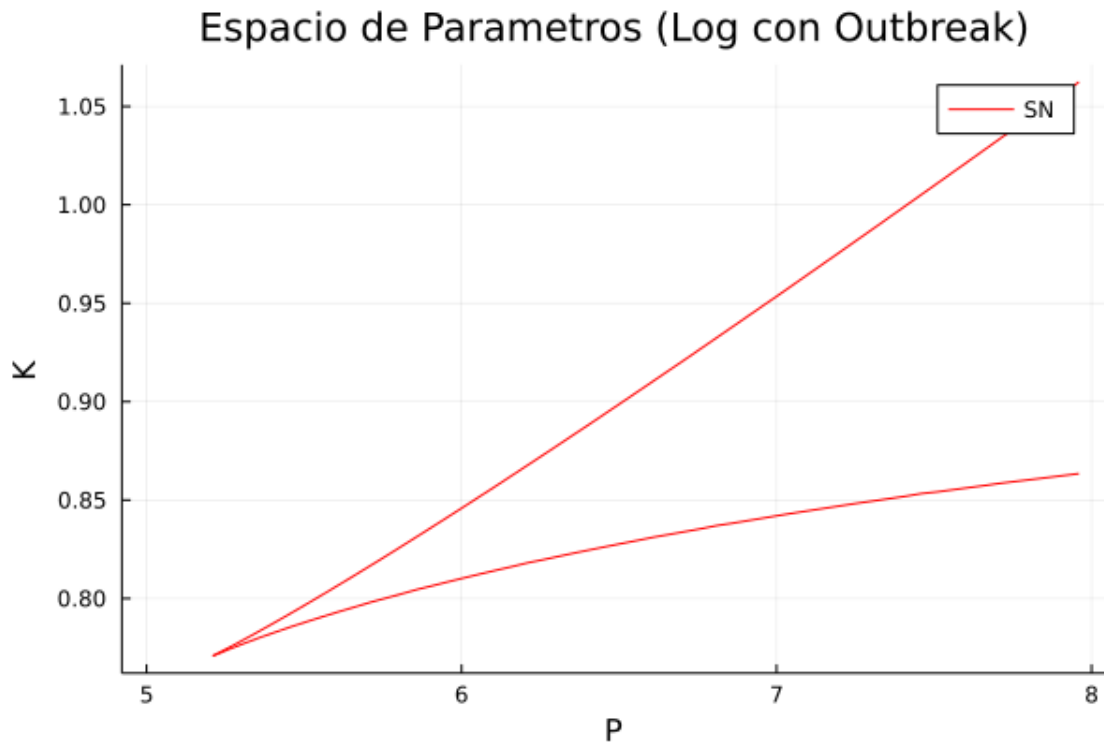
Es decir que para un valor particular de K cambia el diagrama de bifurcaciones 1D. En el modelo anterior dijimos que a pesar de tener varios parametros en realidad la bifurcacion se cruzaba moviendo uno solo de ellos (codimension 1) y esa bifurcacion en un espacio de parametros 2D

correspondia a una curva.

Como va a ser ahora el diagrama de bifurcaciones en el espacio (P, K) ? Ciertamente no va a ser una curva sola porque hay dos bifurcaciones de nodo silla, pero ademas a medida que variamos K esas dos curvas se van juntando (notar que los puntos de bifurcacion estan cada vez mas proximos) y cerca de $K = 5.2$ deberian juntarse en un mismo punto y desaparecer(o colisionar). Atencion que no estamos hablando de puntos fijos lo que colisionan son dos puntos (o en realidad curvas en 2D) de bifurcacion.

```
[7]: k = [8.01-0.05*n for n in 1:60]
opts = ContinuationPar(pMin=0.6,pMax=1.2, detectBifurcation=3)
br1, = continuation(F,[6.5],[0.5,k[1],0.6],(@lens _[3]), opts)
p1 = [br1.specialpoint[1].param]
p2 = [br1.specialpoint[2].param]
x1 = br1.specialpoint[1].x
x2 = br1.specialpoint[2].x
x = [br1.x]
p = [br1.param]
for n=2:60
    opts = ContinuationPar(pMin=0.6,pMax=1.2, detectBifurcation=3)
    br1, = continuation(F,[6.5],[0.5,k[n],0.6],(@lens _[3]), opts)
    if length(br1.specialpoint)>0
        append!(p1,br1.specialpoint[1].param)
        append!(p2,br1.specialpoint[2].param)
        append!(x1,br1.specialpoint[1].x)
        append!(x2,br1.specialpoint[2].x)
    else
        append!(p1,NaN)
        append!(p2,NaN)
        append!(x1,NaN)
        append!(x2,NaN)
    end
    append!(x,[br1.x])
    append!(p,[br1.param])
end
plot(k,p1,color=:red,label="SN",fmt=:png)
plot!(k,p2,color=:red,label="",xlabel="P",ylabel="K",title="Espacio de_
↳Parametros (Log con Outbreak)")
```

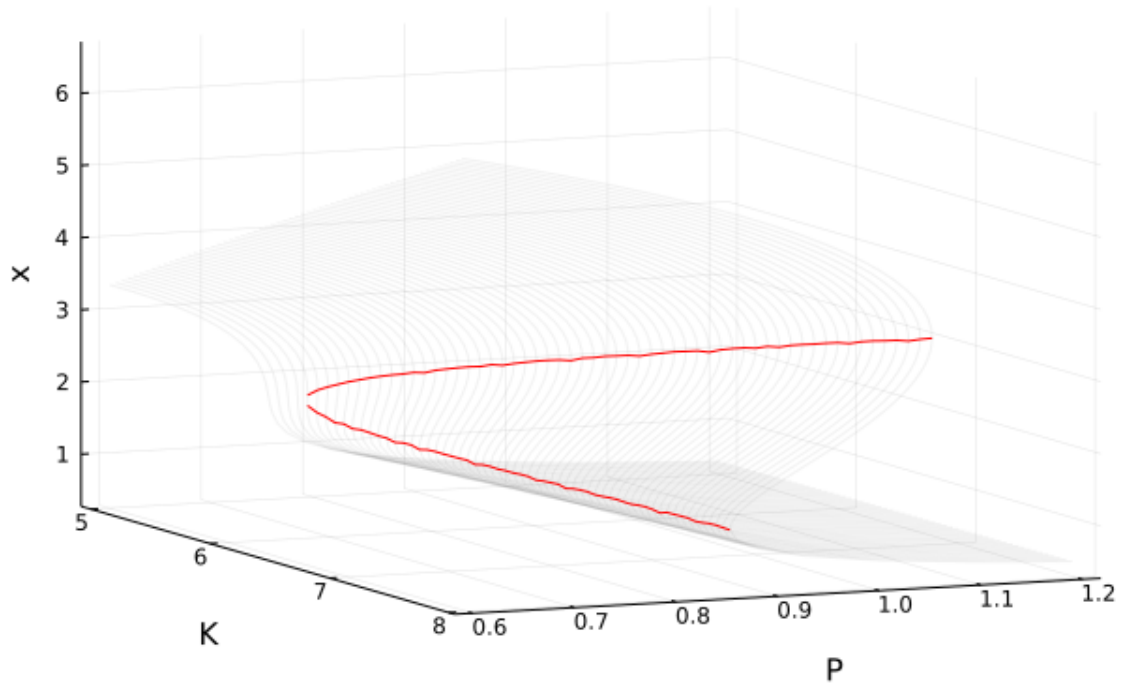
[7]:



```
[9]: pl1=plot()
az = 60; el = 30
#@manipulate throttle=0.05 for az=0:5:90, el=0:5:90
  for n=1:60
    N = length(x[n])
    plot!(pl1,k[n]*ones(N,),p[n],x[n],color=:gray,alpha=0.
↪1,legend=false,camera = (az,el))
  end
  plot!(pl1,k,p1,x1,color=:red,legend=false,xlabel="K",ylabel="P",zlabel="x")
  plot!(pl1,k,p2,x2,color=:red,title="Log con Outbreak",fmt=:png)
#end
```

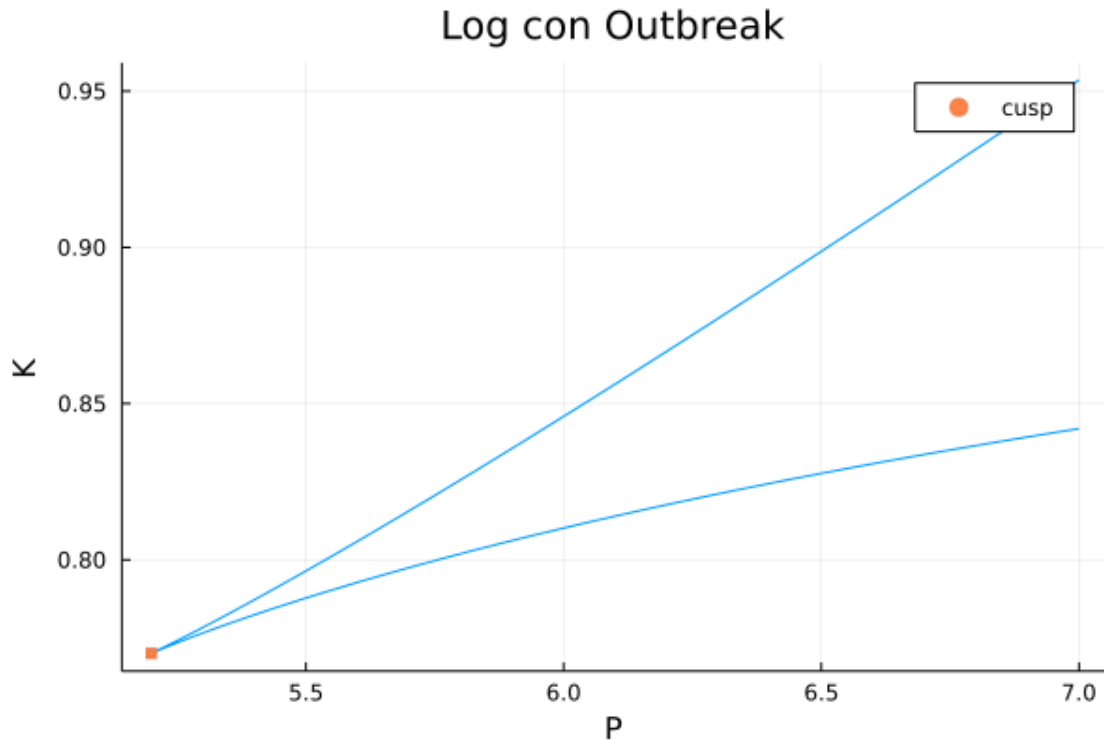
[9]:

Log con Outbreak



```
[10]: FJ = BifurcationKit.getJet(F; matrixfree=false)
      opts = ContinuationPar(pMin=0.6,pMax=1.2, detectBifurcation=3)
      br1, = continuation(FJ[1],FJ[2],[6.5],[0.5,7.0,0.6],(@lens _[3]), opts;␣
        ↪recordFromSolution = (x, p) -> (x = x[1]))
      opts2 = ContinuationPar(opts, pMin=5.0,pMax=7.0, ds = -0.001, dsmax = 0.04)
      br2, = continuation(FJ[1:2]..., br1, 1, (@lens _[2]), opts2;␣
        ↪detectCodim2Bifurcation=2,updateMinAugEveryStep = 1)
      plot(br2,xlabel="P",ylabel="K",title="Log con Outbreak",fmt=:png)
```

[10]:



2.1 Cusp forma Normal

El sistema anterior resulta sumamente arduo de resolver analíticamente, vamos a tomar un ejemplo más sencillo dependiente de dos parámetros (aunque no tiene una aplicación directa) que también experimenta dos bifurcaciones de nodos silla que se juntan en un punto cusp en el espacio de parámetros. En realidad el ejemplo es la “forma normal” de esta bifurcación aunque en el curso no entramos en el detalle de lo que son las formas normales (al menos de manera formal), así que por ahora podemos tomarla como la forma más simple o universal.

El sistema es:

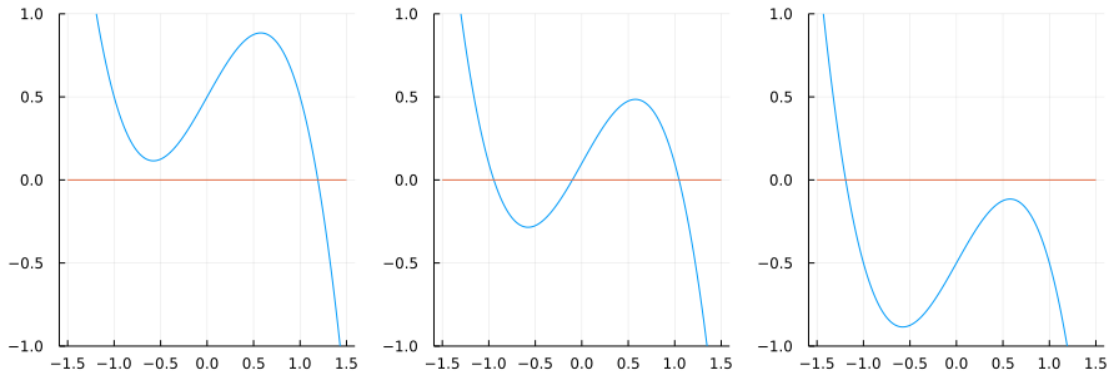
$$\dot{x} = \mu_1 + \mu_2 x - x^3$$

Vamos a evitarnos la tarea de calcular los puntos fijos (que implica resolver la cúbica como en el caso anterior) e interpretar el miembro derecho (campo vector) geoméricamente. Si recordamos la forma de la cúbica (sin término cuadrático) cuando estudiamos los puntos fijos del oscilador de Duffing, podía ser tipo “N” o no dependiendo de si el signo de la parte lineal con respecto a la cúbica. En este caso la diferencia es que μ_1 suma como una constante. Si tenemos $\mu_2 > 0$ la cúbica forma una “N” y podemos tener uno o tres puntos fijos dependiendo del valor de μ_1 que sube o baja la curva con un valor constante:

```
[11]: #ejemplo con mu2=1 y diferentes valores de mu1
x = -1.5:0.01:1.5
p1=plot(x,.5 .+x-x.^3,ylims=(-1,1));plot!(p1,[-1.5,1.5],[0,0],legend=false)
p2=plot(x,.1 .+x-x.^3,ylims=(-1,1));plot!(p2,[-1.5,1.5],[0,0],legend=false)
```

```
p3=plot(x,-.5 .+x-x.^3,ylims=(-1,1));plot!(p3,[-1.5,1.5],[0,0],legend=false)
plot(p1,p2,p3,layout=(1,3),size=(900,300),fmt=:png)
```

[11]:

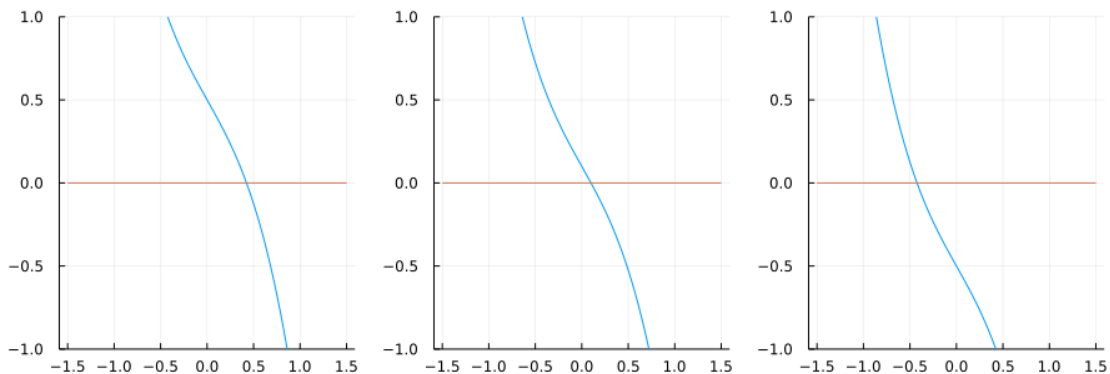


Podemos anticipar que van a ocurrir dos bifurcaciones de nodo silla cuando se pase de 1 a 3 puntos fijos y viceversa. Si en cambio $\mu_2 < 0$ La cubica esta “estirada” hay siempre un punto fijo (atractor, la pendiente es negativa) y no hay bifurcaciones.

[12]:

```
# ahora con mu2=-1
p1=plot(x,.5 .-x-x.^3,ylims=(-1,1));plot!(p1,[-1.5,1.5],[0,0],legend=false)
p2=plot(x,.1 .-x-x.^3,ylims=(-1,1));plot!(p2,[-1.5,1.5],[0,0],legend=false)
p3=plot(x,-.5 .-x-x.^3,ylims=(-1,1));plot!(p3,[-1.5,1.5],[0,0],legend=false)
plot(p1,p2,p3,layout=(1,3),size=(900,300),fmt=:png)
```

[12]:



Por lo tanto podemos inferir que en algun punto las dos bifurcaciones de nodo silla van a tener que colisionar en un punto cusp. No es dificil ver en este caso que eso ocurre para $\mu_2 = 0$, $\mu_1 = 0$

Tampoco es dificil dibujar el diagrama de bifurcaciones. Vamos a dejar fijo μ_2 por ahora y estudiar el diagrama de bifurcaciones en el plano (μ_1, x) como haciamos con la bifurcaciones de codimension 1.

En ese caso recordemos que los puntos fijos corresponden con curvas que estan dadas de forma implicita por $f(x, \mu_1) = 0$ donde f es el miembro derecho de la ecuacion (el campo vector). En

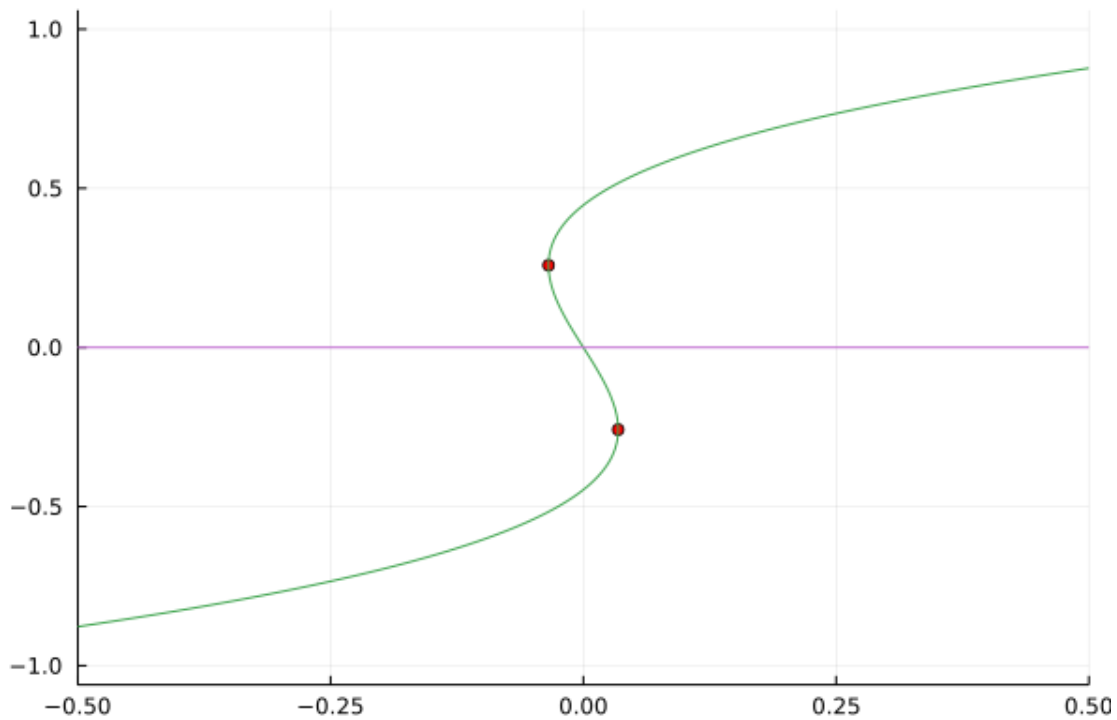
este caso es mucho mas facil despejar de forma explicita μ_1 en funcion de x (para un valor de fijo).

$$\mu_1(x) = x^3 - \mu_2 x$$

que como antes si $\mu_2 > 0$ es una cubica con forma de “N” que se estira para el otro caso, pero acostada:

```
[19]: x = -1:0.01:1
      2 =0.2
      #@manipulate for 2=-1:0.01:1
      plot()
      if ( 2>0)
        x0 = sqrt( 2/3)
        scatter!([-2*x0^3],[x0],color=:red)
        scatter!([2*x0^3],[-x0],color=:red)
      end
      plot!(x.^3- 2*x,x);plot!([-1,1],[0,0],xrange=(-0.5,0.5),legend=false,fmt=:
      ↪png)
      #end
```

[19]:



Ademas podemos calcular para que valores de (μ_1, x) ocurren las bifurcaciones de nodo silla (en el caso de $\mu_2 > 0$). Solo tenemos que encontrar cuando la derivada de μ_1 con respecto a x se hace cero y eso sucede cuando $3x^2 = \mu_2$ o sea para los puntos:

$$x = -\sqrt{\mu_2/3}, \mu_1 = 2\left(\sqrt{\mu_2/3}\right)^3$$

$$x = \sqrt{\mu_2/3}, \mu_1 = -2\left(\sqrt{\mu_2/3}\right)^3$$

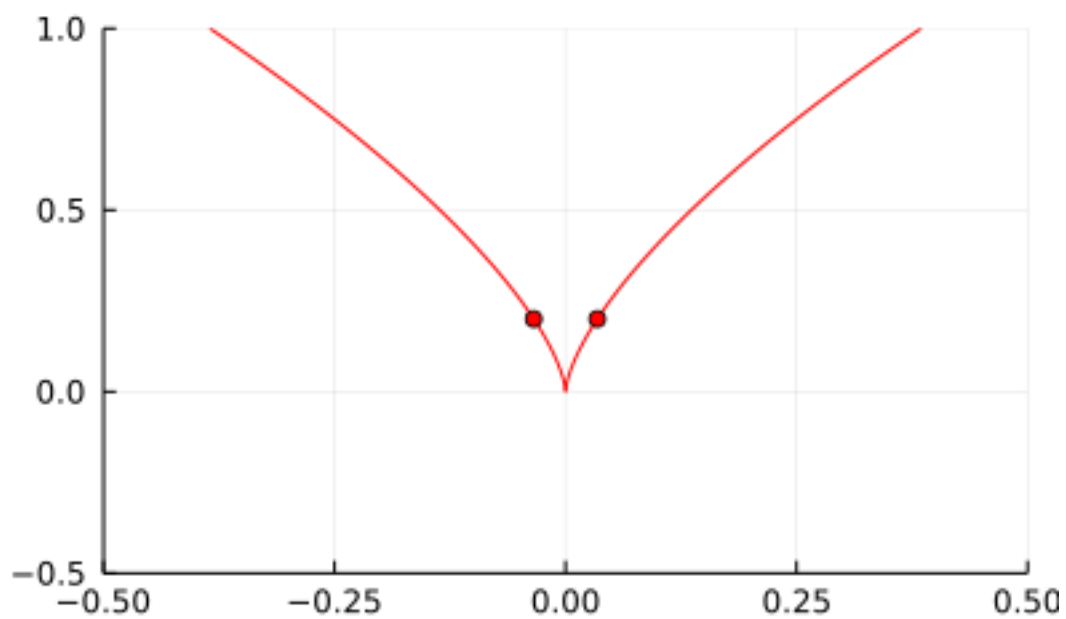
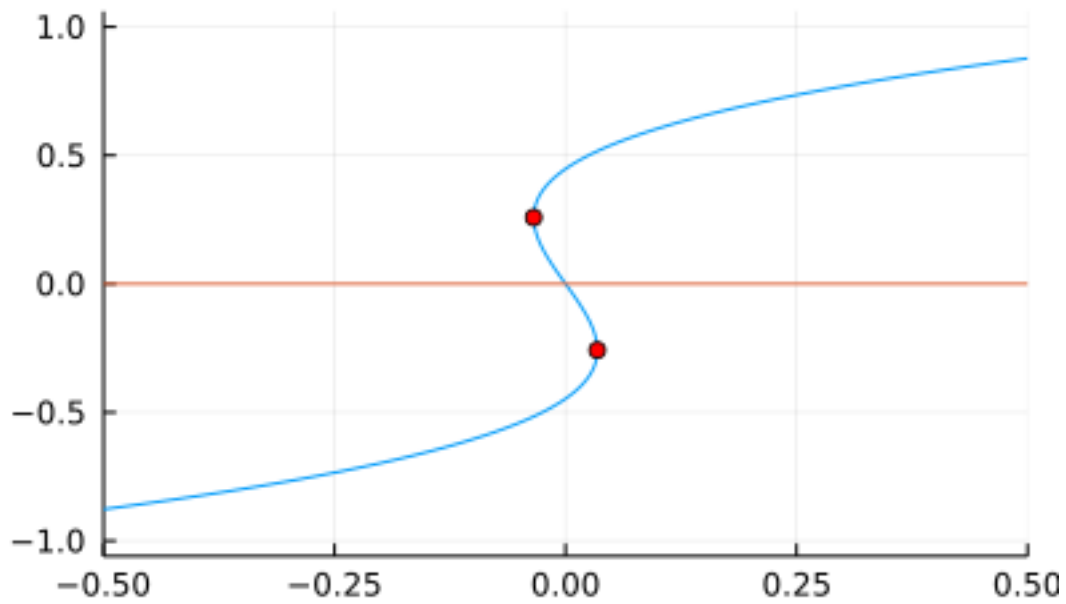
esos puntos estan representados en el grafico interactivo de arriba

Vamos ahora a trazar el diagrama de bifurcaciones completo en el espacio de parametros (μ_1, μ_2) . Las curvas que determinan la bifurcacion de nodo silla en este espacio corresponden a los dos puntos calculados anteriormente pero ahora expresados como μ_1 en funcion de μ_2 en el dominio $\mu_2 > 0$

$$\mu_1 = \pm 2\left(\sqrt{\mu_2/3}\right)^3$$

```
[20]: 2list = 0:0.01:1
#@manipulate throttle=.1 for 2=-0.5:0.01:1
    p1 = plot(x.^3-2*x,x);plot!([-1,1],[0,0],xlims=(-0.5,0.5),legend=false)
    p2 = plot(-2*sqrt.(2list/3).^3, 2list,xlims=(-0.5,0.5),legend=false,color=:
↪red)
    p2 = plot!(2*sqrt.(2list/3).^3, 2list,ylims=(-0.5,1),color=:red)
    if (2>0)
        x0 = sqrt(2/3)
        scatter!(p1,[-2*x0^3],[x0],color=:red)
        scatter!(p1,[2*x0^3],[-x0],color=:red)
        scatter!(p2,[-2*x0^3],[2],color=:red)
        scatter!(p2,[2*x0^3],[2],color=:red)
    else
        scatter!(p2,[0],[2],color=:gray)
    end
    plot(p1,p2,layout=(2,1),size=(400,500),fmt=:png)
end
```

[20]:



3 Bifurcacion Bogdanov-Takens (codimension 2)

Las bifurcaciones (de codimension 1) que vimos hasta ahora se pueden caracterizar en dos grupos:

- Las que suceden cuando un autovalor real se hace cero. Tenemos de forma generica una bifurcacion de nodo silla, pero tambien una pitchfork o una transcritica si se dan otras condiciones de simetria.

- La que sucede cuando la parte real de dos autovalores complejos conjugados se hace cero. En ese caso tenemos una bifurcación de Hopf.

Claramente el primer caso puede suceder en un sistema 1D (o uno de mayor dimensión a lo largo de una dirección particular), mientras que el segundo necesitamos al menos un sistema 2D para tener dos autovalores, pero hace falta solo un parámetro para controlarla (para mover la parte real).

Pero si estamos en un sistema 2D, no podría suceder que **ambos** autovalores se hagan cero de forma simultánea? Claramente si vemos la expresión de los autovalores vamos a necesitar al menos dos parámetros para ajustar este punto. Por otra parte que dos autovalores crucen el cero por el eje real sería algo así como dos nodo-silla sucediendo al mismo tiempo, pero también podría verse como una bifurcación de Hopf con parte imaginaria cero. Es decir este tipo de bifurcación “degenerada” tiene adentro al menos dos nodo silla (como la cusp) y una Hopf.

Esta bifurcación de codimensión 2 se conoce como doble cero o Bogdanov-Takens (o Takens-Bogdanov). La forma normal está caracterizada por tener el siguiente Jacobiano:

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

el cual “induce” que aparezcan los siguientes términos no lineales (en la versión de Bogdanov):

$$\dot{x} = y$$

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

Notar que todavía no introdujimos ningún parámetro, esto es la singularidad “pura”. Para extender esto en el espacio de parámetros (o de sistemas dinámicos para ser más preciso) es necesario hacer un **unfolding**, y acá hay varias posibilidades, vamos tomar la que hace Guckenheimer & Holmes:

$$\dot{x} = y$$

$$\dot{y} = \mu_1 + \mu_2 x + x^2 - xy$$

Vamos a estudiar directamente las bifurcaciones sin preocuparnos todavía por las soluciones porque este sistema (al igual del nodo silla en el plano) tiene trayectorias divergentes.

Como tenemos términos hasta orden cuadrático vamos a poder tener de forma genérica dos puntos fijos o ninguno. Siempre ubicados en el eje horizontal $y = 0$ y con la coordenada x en:

$$x_{\pm} = -\frac{\mu_2}{2} \pm \sqrt{\frac{\mu_2^2}{4} - \mu_1}$$

El signo positivo corresponde al punto fijo de la derecha y el negativo al de la izquierda (cuando estos existan).

la condición para existencia de los puntos fijos es que el interior de la raíz sea positivo lo cual nos da una condición para trazar una curva de bifurcación nodos silla en el plano (μ_1, μ_2) :

$$\text{SN} : \mu_1 = \frac{\mu_2^2}{4}$$

Por otra parte el Jacobiano evaluado en los puntos fijos (conservando el orden \pm) queda escrito

$$\begin{pmatrix} 0 & 1 \\ \pm 2\sqrt{\frac{\mu_2^2}{4} - \mu_1} & \frac{\mu_2}{2} \mp \sqrt{\frac{\mu_2^2}{4} - \mu_1} \end{pmatrix}$$

Lo cual nos da el determinante:

$$\Delta = \mp 2\sqrt{\frac{\mu_2^2}{4} - \mu_1}$$

que para el punto fijo de la derecha es siempre negativo (punto silla) y para el de la izquierda positivo. Para este ultimo evaluamos la traza (solo conservamos el signo de abajo):

$$\tau = \frac{\mu_2}{2} + \sqrt{\frac{\mu_2^2}{4} - \mu_1}$$

como la raiz es positiva cuando $\mu_2 < 0$, la traza se va a hacer cero cuando $\mu_1 = 0$. Esto nos da la condicion para trazar la curva de la bifurcacion de Hopf en el plano (μ_1, μ_2) :

Hopf: $\mu_2 > 0$, $\mu_1 = 0$

Esta curva se junta con la parabola SN en el punto $(0,0)$ asi que tal como habiamos anticipado en este punto singular tenemos una curva SN y una Hopf ocurriendo simultaneamente. El diagrama de bifurcaciones completo (tomado de la Scholarpedia) es:

Notar que hay una curva adicional (en rojo) que corresponde a una bifurcacion global (conexion homoclinica).

Ya vimos esta bifurcación! Y con esta misma forma normal, pero fijando arbitrariamente $\mu_2 = -1$. Se trata de una bifurcación homoclínica o saddle loop. El cambio de la region (3) a la (4) es el que describimos cuando vimos esta bifurcación. El ciclo límite originado en la curva de la Hopf crece hasta tocar el saddle y las variedades del saddle y formar un loop. Del otro lado de la bifurcación no hay ciclo límite y la variedad inestable del saddle alimenta un foco atractor.

Explorar las como se modifican las variedades en el grafico de abajo y tratar de localizar cuando se produce la conexion homoclinica. Como guia se muestra a la derecha el diagrama de bifurcaciones con las curvas SN y Hopf analiticas y en linea punteada la homclinica que ocurre (no vamos a mostrar la deducccion de eso) cuando:

$$\text{HC: } \mu_1 = -\frac{6}{25}\mu_2^2, \mu_2 < 0$$

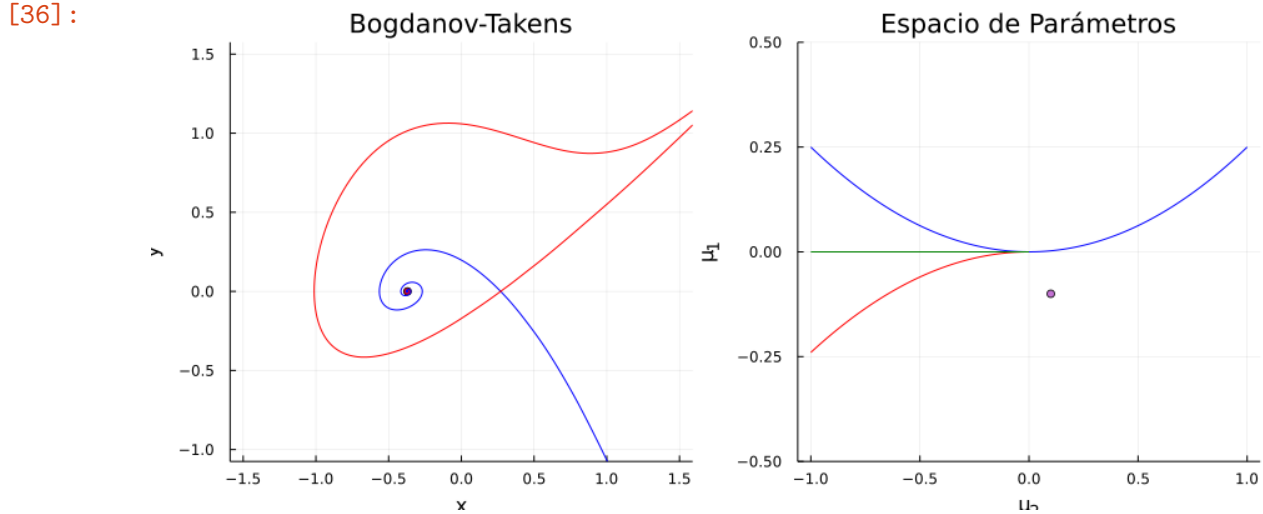
```
[15]: function takens!(du,u,p,t)
      du[1]=u[2]
      du[2]=p[1]+p[2]*u[1]+u[1]*u[1]-u[1]*u[2]
      du
    end
```

```
[15]: takens! (generic function with 1 method)
```

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

```
[16]: takens_jac (generic function with 1 method)
```

```
[36]: 1 = -0.1; 2=0.1
#@manipulate throttle=0.5 for 1=-0.5:0.005:0.5, 2=-1.1:0.005:1
    if 1 < 2^2/4
        u0_array=[[- 2/2-sqrt( 2^2/4- 1);0],[- 2/2+sqrt( 2^2/4- 1);0]]
        p1=flux2d_manifolds(takens!,takens_jac,u0_array,[1, 2];
            ↪5],size=(400,400),title="Bogdanov-Takens")
    else
        p1=plot(title="Bogdanov-Takens")
    end
    p2 = plot(-1:0.01:1,0.25*(-1:0.01:1).^2,color=:blue,legend=false)
    plot!(p2,-1:0.01:0,-6/25*(-1:0.01:0).^2,color=:red,legend=false)
    plot!(p2,[-1,0.0],[0,0],color=:green,xlabel="\mu_2",ylabel="\mu_1")
    scatter!(p2,[2],[1],ylims=(-0.5,0.5),title="Espacio de Parámetros")
    plot(p1,p2,layout=(1,2),size=(900,400),fmt=:png)
end
```



Si bien en este caso pudimos calcular las curvas de bifurcación de forma analítica (al menos SN y Hopf) Vamos a usar BifurcationToolkit para seguir las curvas de bifurcación. Empezamos con la más fácil que es la SN. Fijando $\mu_2 = 1$ barreemos el parámetro μ_1 para cortar la Hopf y la Saddle Node.

Para mejorar la precisión del método armamos el “Jet” (algo así como un array del sistema dinámico y sus derivadas) con la función `getJet` y el Jacobiano calculado numéricamente `jacobian` de3l paquete `ForwardDiff`. Como era esperable para este valor de μ_2 un punto Hopf en $\mu_1 = 0$ y la saddle node en $\mu_1 = 1/4$.

```
[37]: takens(u,p) = takens!(similar(u),u,p,0)
      opts = ContinuationPar(pMin=-0.1, pMax=0.28, ds = 0.001, dsmax = 0.
      ↪02,detectBifurcation = 3)
```

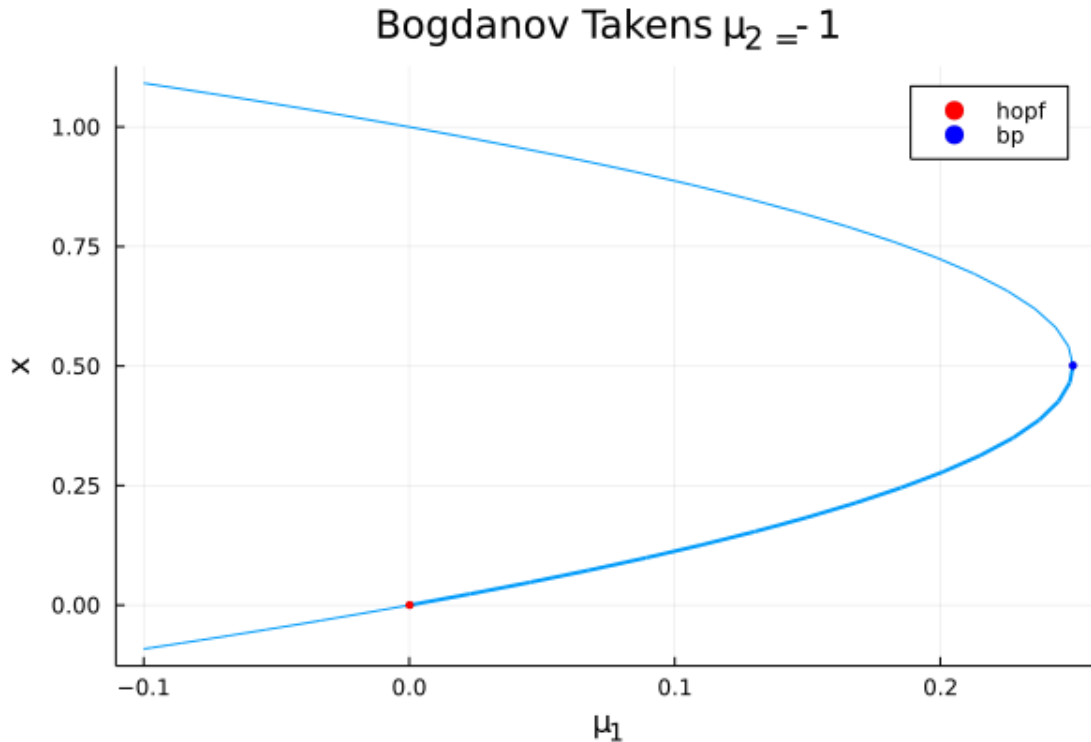


```

FJ = BifurcationKit.getJet(takens,(z,p) -> ForwardDiff.jacobian(x ->
    ↪takens(x,p), z))
br1, = continuation(FJ[1],FJ[2],[-0.9;0],[-0.1,-1.0],(@lens ↪
    ↪_[1]),opts,recordFromSolution = (x, p) -> x[1])
plot(br1,xlabel="\mu_1",title="Bogdanov Takens \mu_2 = -1",fmt=:png)

```

[37]:



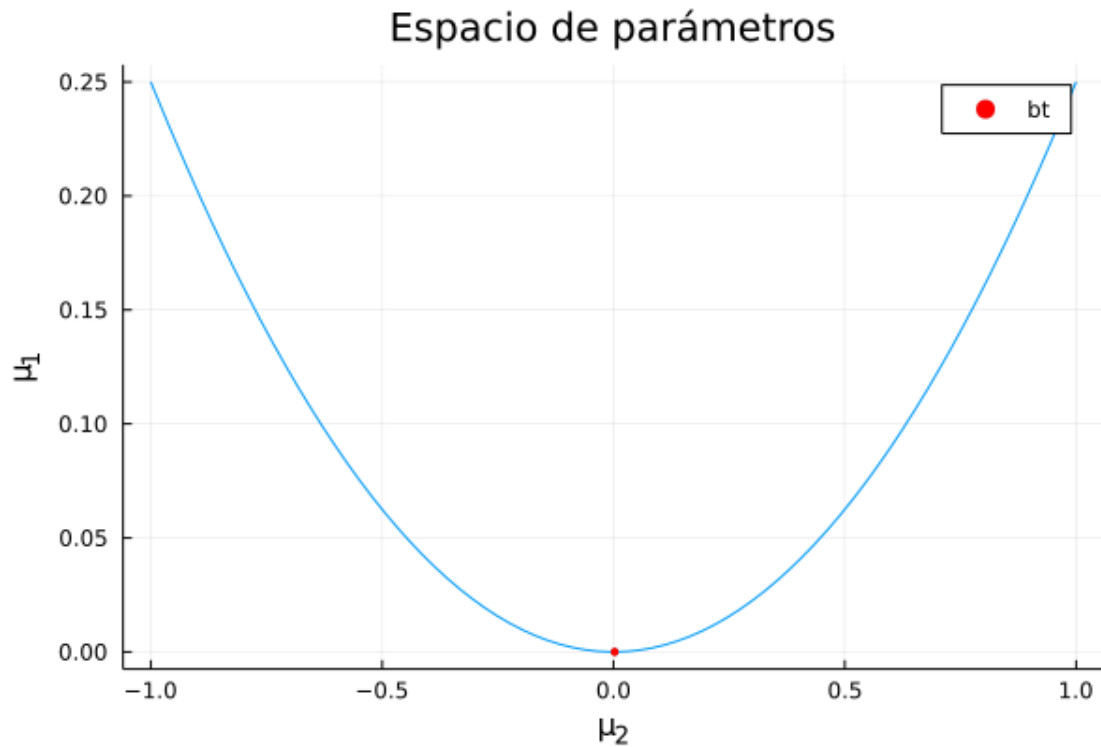
Ahora estamos listos para trazar la curva SN en el espacio de parámetros (μ_1, μ_2) . Para ello tomamos el segundo punto especial de esta rama *br1* que corresponde a la saddle node (indicado bp) y lo continuamos variando μ_1 entre -1 y 1, indicando que detecte bifurcaciones de codimension 2 sobre esa curva.

```

[38]: SN, = continuation(FJ[1],FJ[2],br1,2,(@lens ↪_[2]), ContinuationPar(opts, ↪
    ↪pMin=-1.0,pMax=1.0);
    detectCodim2Bifurcation=2,updateMinAugEveryStep = 1,bdlinsolver = ↪
    ↪MatrixBLS())
plot(SN,xlabel="\mu_2",ylabel="\mu_1",title="Espacio de parámetros",fmt=:png)

```

[38]:

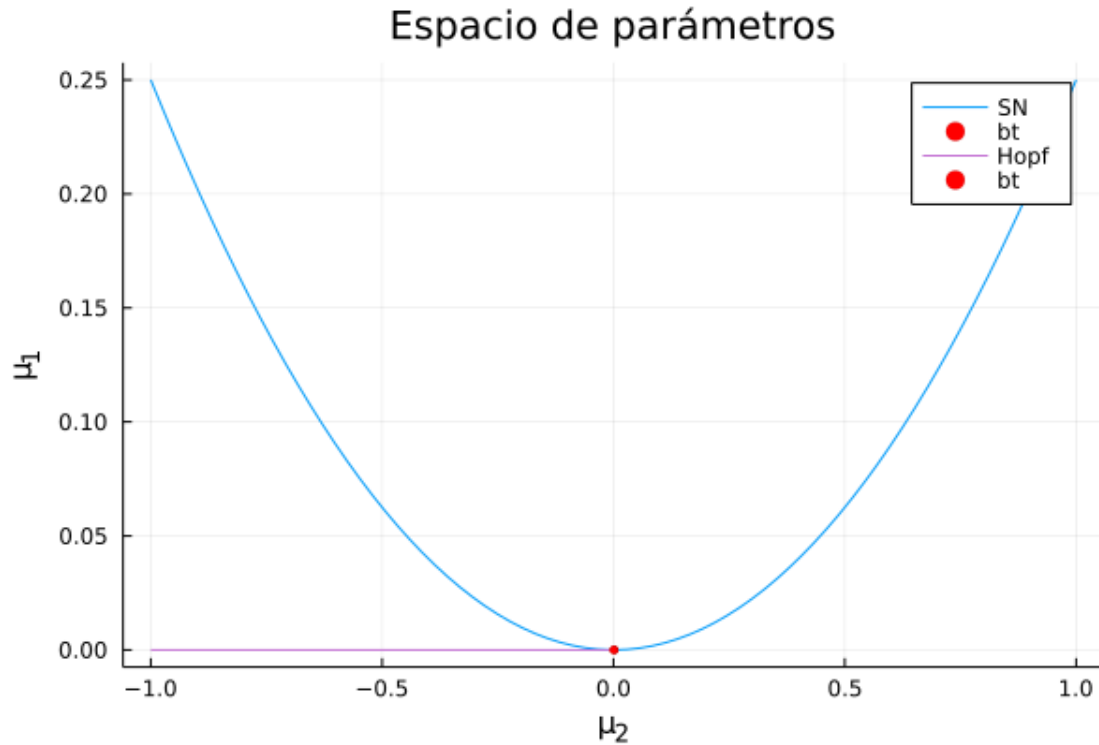


Que tal como esperábamos es una parábola (la condición para la SN era $\mu_1 = \frac{\mu_2^2}{4}$) y además detectó la Bogdanov-Takens en el origen (bt).

La curva de la Hopf la continuamos de forma similar indicando que vamos a seguir el primer punto especial de br1. Graficamos ambas curvas en el espacio de parámetros

```
[39]: H, = continuation(FJ[1],FJ[2],br1,1,(@lens _[2]), ContinuationPar(opts, pMin=-1.
    ↪ 0,pMax=0.1);
    detectCodim2Bifurcation=2,startWithEigen=true,d2F=FJ[3],d3F_
    ↪ FJ[4],updateMinAugEveryStep = 1,
    bdlinsolver = MatrixBLS())
plot(SN,branchlabel="SN")
plot!(H,branchlabel="Hopf",xlabel="\mu_2",ylabel="\mu_1",title="Espacio de_
    ↪ parámetros",fmt=:png)
```

[39]:



Notar que el punto de codimension 2 figura 2 veces porque lo detecto en ambas curvas. BifurcationKit no nos permite calcular bifurcaciones globales (afaik)

3.1 Bogdanov Takens con terminos cubicos

Como puede verse la bifurcacion BT presenta una dinamica muy interesante y variada, con minimas alteraciones en los parametros se puede pasar de comportamientos oscilatorios, creacion de pares de puntos fijos y orbitas de periodo infinito (conexiones HC). El problema del sistema anterior es que tiene trayectorias divergentes, por lo cual necesitamos agregar terminos de orden mas alto (que no van a alterar la bifurcacion de BT aunque si pueden cambiar el diagrama de bifurcaciones fuera de ese punto) que aseguren que las trayectorias no diverjan. De vuelta hay varias alternativas, vamos a seguir la que propone Mindlin:

$$\dot{x} = y$$

$$\dot{y} = \mu_1 + \mu_2 x + x^2 - xy - x^3 - x^2 y$$

En este caso al tener terminos cubicos vamos a tener de forma general uno o tres puntos fijos, como en el caso de la cusp los puntos fijos pasan de ser 1 a 3 a traves de bifurcaciones de nodo silla que ocurren en pares de puntos distintos. De hecho los terminos cubicos lo que hacen es introducir una cusp ademas de la Bogdanov-Takens.

Vamos a ver como se organizan las variedades a partir de estos nuevos términos

```
[40]: function takens3!(du,u,p,t)
      du[1]=u[2]
```

```

du[2]=p[1]+u[1]*(p[2]-u[2]+u[1]*(1-u[1]-u[2]))
du
end

```

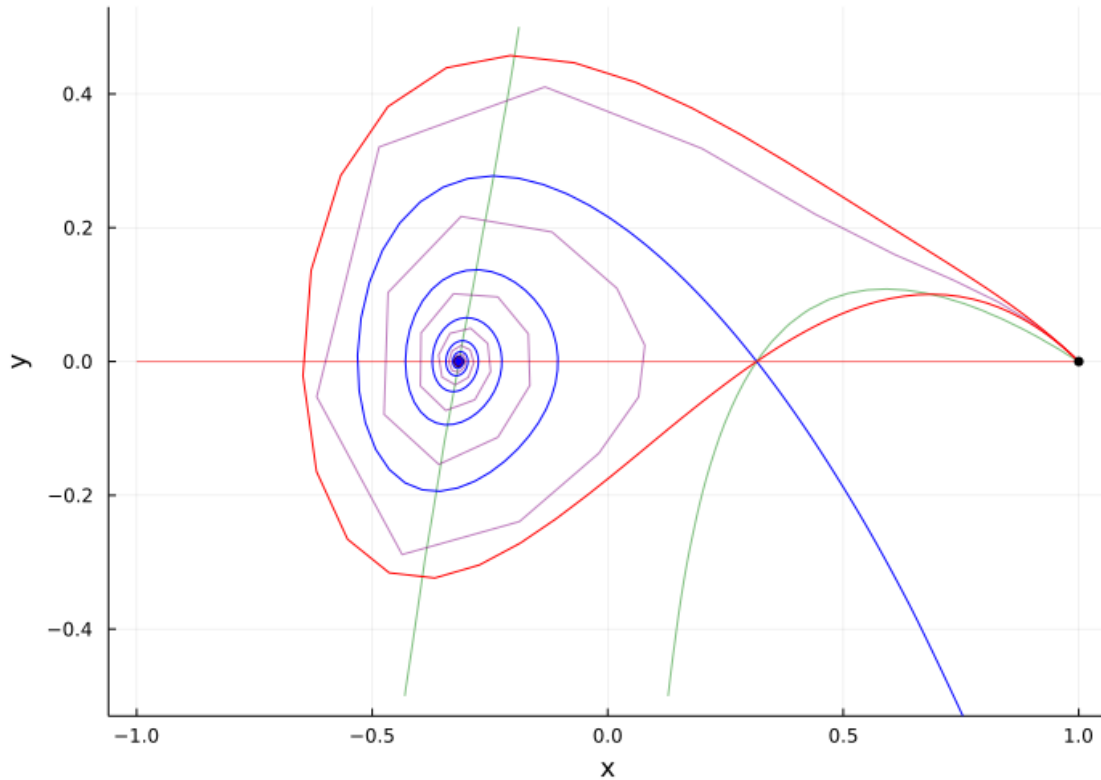
[40]: takens3! (generic function with 1 method)

```

[41]: #@manipulate throttle=0.1 for 1=-0.15:0.001:0.1, 2=-0.4:0.001:0.1
      phase_portrait(takens3!, [1, 2];tmax=300,xlims=[-1,1],ylims=[-0.5,0.5])
      #end

```

[41]:



Este retrato de fase nos puede resultar conocido, sobre todo si fijamos μ_1 en un valor negativo y variamos μ_2 . Para valores crecientes de μ_2 observamos una saddle node sobre el ciclo límite que previo a la bifurcación va formando un “pico” apretado por la nulclina de $\dot{y} = 0$ tal como vimos cuando presentamos esta bifurcación. De hecho el sistema es el mismo.

Esta saddle node es diferente de la otra rama en forma de parábola de la BT original que me aniquila los dos puntos de la izquierda. Por lo tanto es de esperar que tengamos otra curva diferente de SN en el espacio de parámetros, que como vamos a ver se aniquila con una rama de la curva SN parabolica SN original de Bogdanov Takens.

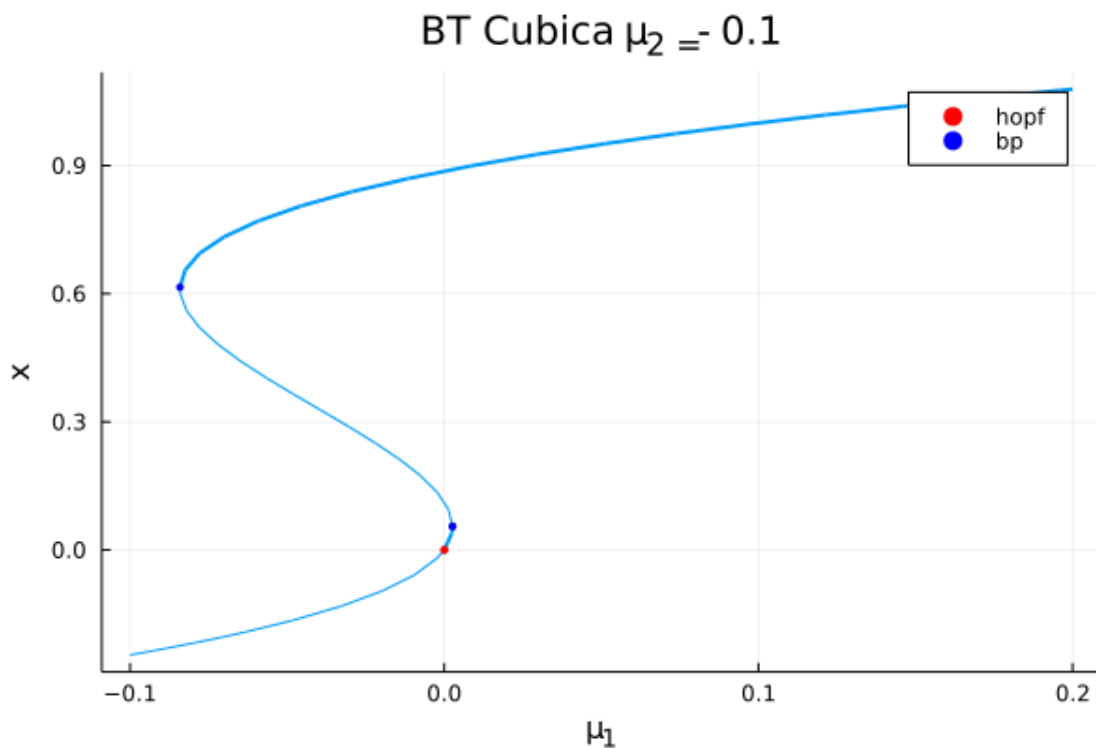
Pero vamos por partes. Vamos a usar BifurcationKit porque en este caso es mas engorroso calcular a mano las curvas. Lo primero es fijar μ_2 en un valor donde estemos seguros que vamos a cortar la SN (y de paso la Hopf) variando μ_1 . Lo mas seguro es a la izquierda de la BT. En este caso

tomamos $\mu_1 = -0.1$ y variamos μ_1 en el rango $(-0.1, 0.2)$.

```
[42]: takes3(u,p) = takes3!(similar(u),u,p,0)
      opts = ContinuationPar(pMin=-0.1, pMax=0.2, ds = 0.001, dsmax = 0.
      ↪02,detectBifurcation = 3)
      FJ = BifurcationKit.getJet(takes3,(z,p) -> ForwardDiff.jacobian(x ->
      ↪takes3(x,p), z))
      br1, = continuation(FJ[1],FJ[2],[-0.9;-0.1],[-0.1,-0.1],(@lens_
      ↪_[1]),opts,recordFromSolution = (x, p) -> x[1])
      print([pt.type for pt in br1.specialpoint])
      plot(br1,xlabel="\mu_1",title="BT Cubica \mu_2 =-0.1",fmt=:png)
```

[:hopf, :bp, :bp]

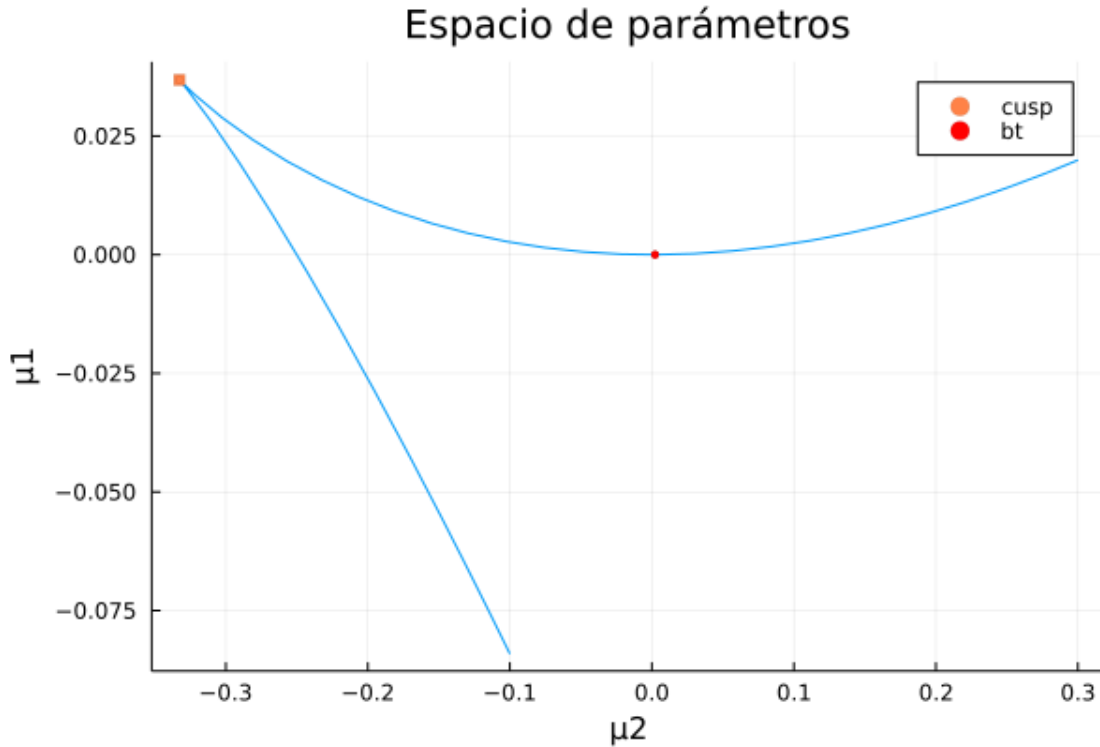
[42]:



Podemos ver que detecto tres puntos especiales, le pedimos que imprima su tipo para luego elegir cual vamos a continuar para calcular las ramas. Tal como anticipamos tenemos dos bifurcaciones SN vamos a continuar una de ellas para ver si se una con la otra en una cusp. Esto lo hacemos en el diagrama de abajo tomando el punto especial 3 y variando μ_2 en un amplio rango.

```
[43]: SN, = continuation(FJ[1],FJ[2],br1,3,(@lens_[2]), ContinuationPar(opts,
      ↪pMin=-0.4,pMax=0.3,ds=-0.001);
      detectCodim2Bifurcation=2,updateMinAugEveryStep = 1,bdlsolver =
      ↪MatrixBLS())
      plot(SN,title="Espacio de parámetros",xaxis=(" 2"),yaxis=(" 1"),fmt=:png)
```

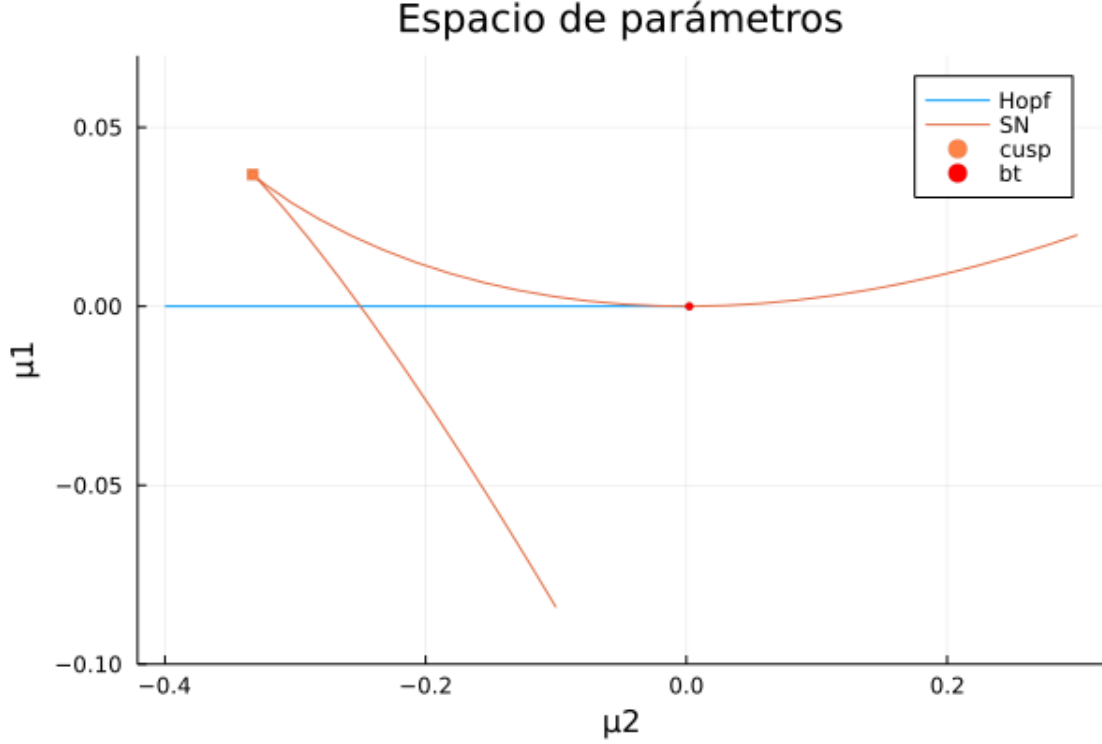
[43]:



Como habíamos sospechado las dos ramas de SN se unen en un punto cusp de codimension 2, por lo tanto no vamos a tener que continuar la otra curva SN. Vamos ahora por la Hopf que quedo inalterada por los nuevos terminos y es una linea horizontal en $\mu_1 = 0$ para valores de μ_2 negativos:

```
[44]: br2, = continuation(FJ[1],FJ[2],[-0.9;0],[-0.1,-0.4],(@lens_
    ↪_[1]),opts,recordFromSolution = (x, p) -> x[1])
H, = continuation(FJ[1],FJ[2],br2,1,(@lens_[2]), ContinuationPar(opts, pMin=-0.
    ↪4,pMax=0.1);
    detectCodim2Bifurcation=2,startWithEigen=true,d2F=FJ[3],d3F_
    ↪=FJ[4],updateMinAugEveryStep = 1,
    bdlinsolver = MatrixBLS())
p1=plot(H,branchlabel="Hopf",plotspecialpoints=false,fmt=:png)
plot!(p1,SN,branchlabel="SN",ylims=(-0.1,0.
    ↪07),xaxis=(" 2"),yaxis=(" 1"),title="Espacio de parámetros")
```

[44]:



En este diagrama nos falta la bifurcación homoclínica que se origina en el punto BT. La presencia de los términos cúbicos la modifica porque tiene que acoplarse a la otra rama de la SN de forma tangencial. No hay una expresión sencilla para esta curva por lo que calculamos de forma numérica. La siguiente subsección se puede saltar y pasar directamente al diagrama final.

3.2 Cálculo analítico de la curva SN y cómputo numérico de la Homoclínica

3.3 [recontra-optativo]

Dijimos que era engorroso el cálculo de la curva SN de forma analítica, acá damos solo la expresión final, pero se puede llegar por argumentos similares a los que utilizamos en el análisis de la función cúbica para la Saddle Node Heteroclínica (si bien en ese caso era un poco más sencillo porque $\mu_2 = 0$).

Las ramas de la curva SN quedan determinadas por:

$$\mu_1 = -\frac{1}{9} [3\mu_2 + 2/3 \pm (2\mu_2 + 2/3)\sqrt{1 + 3\mu_2}]$$

definidas para $\mu_2 > -1/3$. De hecho para $\mu_2 = -1/3, \mu_1 = 1/27$ tenemos el punto cusp.

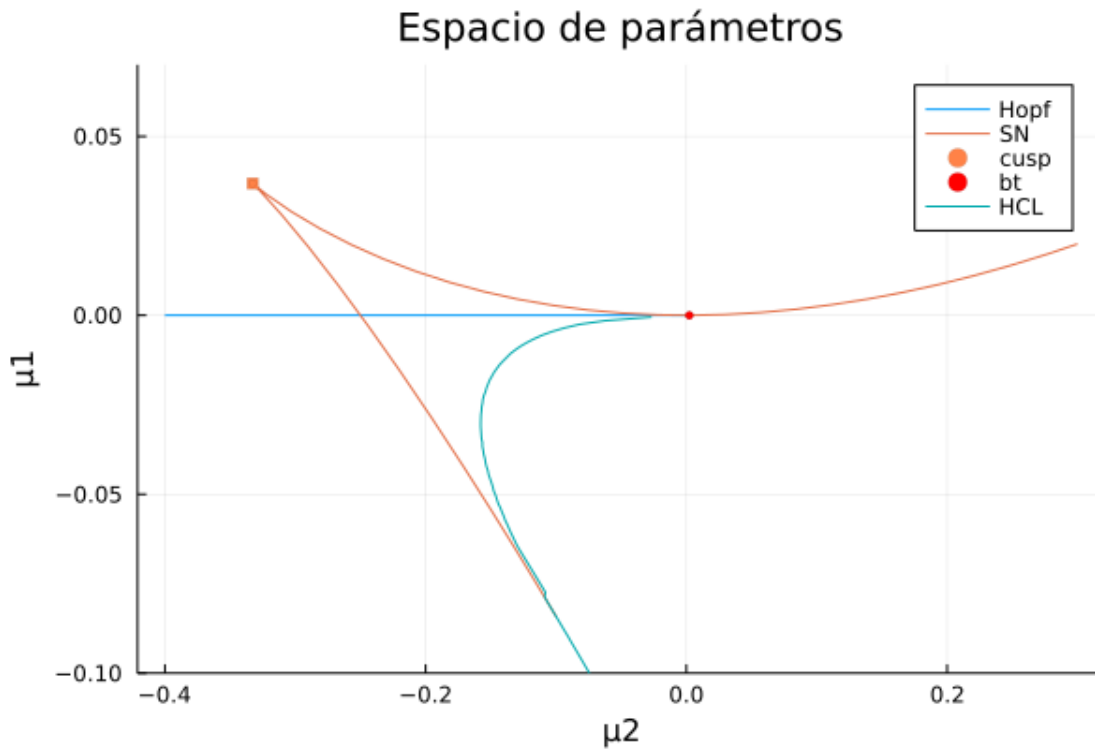
Lo que vamos a hacer es colocarnos sobre la curva SN para un valor de μ_1 definido negativo (que corresponde a la rama con el signo +) e integrar el sistema para determinar si persiste la órbita periódica que está a la izquierda de la curva. Si persiste es porque todavía no cruzamos el saddle loop. Incrementamos entonces el valor de μ_2 un δ pequeño hasta que la órbita periódica desaparece y marcamos ese punto como el valor de (μ_1, μ_2) que corresponde a la bifurcación homoclínica. La forma de detectar la órbita periódica es dejar pasar un transitorio y fijarse en la amplitud de

oscilación de la variable y . Como solo hay atractores y ciclos límites si la amplitud de oscilación (yrange) está por encima de cierto umbral entonces asumimos que tenemos un ciclo límite. Este es un método poco riguroso pero que a veces ofrece mejor resultado que intentar seguir las variedades estable e inestable del punto silla y buscar intersecciones.

```
[45]: # traceamos la homclinica
delta = 0.0001
m2 = -0.249:0.002:0.0
m1a = @. -(3*m2+2/3+(2/3+2.0*m2)*sqrt(1+3.0*m2))/9.0
#m1a = @. -(m2+(2/3+2.0*m2)*(1+sqrt(1+3.0*m2)))/9.0
HCL = []
for n in 1:length(m2)
    yrange = 1.0
    2 = m2[n]
    while yrange>0.0001
        sol = solve(ODEProblem(takens3!, [0.0; 0.0], (0, 300.0), [m1a[n], 2]))
        yrange = maximum(getindex.(sol.u,2)[end-10:end])-minimum(getindex.(sol.
↪u,2)[end-10:end])
        2 += delta
    end
    append!(HCL, 2-delta)
end
```

```
[46]: p1=plot(H,branchlabel="Hopf",plotspecialpoints=false)
plot!(p1,SN,branchlabel="SN",ylims=(-0.1,0.
↪07),xaxis=(" 2"),yaxis=(" 1"),title="Espacio de parámetros")
plot!(p1,HCL,m1a,label="HCL",fmt=:png)
```

[46]:



Y tenemos el diagrama de bifurcaciones en el espacio de parámetros completo!

Vamos a explorarlo ahora de forma interactiva viendo el retrato de fases y la posición en el espacio de parámetros de forma simultanea

```
[47]: #@manipulate throttle=0.1 for 1=-0.25:0.001:0.1, 2=-0.4:0.001:0.1
      p2=phase_portrait(takens3!, [ 1, 2];tmax=300,xlims=[-0.7,0.7],ylims=[-0.5,0.5])
      p1=plot(H,branchlabel="Hopf",plotspecialpoints=false)
      plot!(p1,HCL,m1a,label="HCL")
      plot!(p1,SN,branchlabel="SN",ylims=(-0.1,0.07),xaxis=(" 2"),yaxis=(" 1"))
      scatter!(p1,[ 2],[ 1],label=false)
      plot(p1,p2,layout=(1,2),size=(900,400))
      #end
```

[47]:

