

José Eduardo Soubhia Natali

Doutorando em Fisiologia Geral, Instituto de Biosciências, USP. O título de minha tese é: “Modelagem do Controle Motor Associado com o Sistema Ventilatório em Mamíferos”, orientado pelo ZéGui.

Exercícios

[exec](#)

Proposta de Trabalho Final

Plano A:

Fazer uma função que peça valores iniciais dos parâmetros de uma equação diferencial e devolva a simulação numérica e o gráfico do espaço de fases.

Para tanto usaria um pacote capaz de simular equações diferenciais (deSolve) e uma equação que apresente o comportamento de bifurcação (sendo um dos casos um ciclo limite).

Caso a entrada de valor inicial seja um vetor (e não só um valor), a equação será feita para cada valor e será gerado, adicionalmente, um diagrama de bifurcação.

Este projeto seria bem interessante pois usaria conceitos de programação, parte gráfica e simulação de equações diferenciais, podendo juntar interesses pessoais com os conceitos aprendidos na disciplina. Essa função seria ainda importante para biólogos que procuram entender mais de conceitos básicos de sistemas dinâmicos, já que será possível observar os efeitos das condições iniciais no comportamento do sistema.

Resumidamente penso em algo assim:

```
minha.sim=sim.eq(par=4,par=3, par=6, par= (1:20), graf.ss=TRUE, graf.bif=TRUE)
```

A resposta, dependendo da entrada, será uma matriz (ou data.frame) com os valores de cada variável de estado e dois gráficos.

Comentários

A função apenas resolveria um conjunto pré-determinado de equações ou o usuário poderia alterá-las? Acho que seria interessante permitir que fossem mudadas, mas nesse caso não sei o quanto diferente ficaria das funções já existentes.

— [Fabio de A. Machado](#) 2011/04/06 18:34

Paulo: fiquei com a mesma dúvida que o Fabio, pois vc parece ter já escolhido a função, embora o deSolve (veja tb o odeSolve) seja genérico. Mas a proposta em si seria de fato muito útil para uma exploração rápida do comportamento de equações diferenciais. Outra aperfeiçoamento (se der tempo) seriam gráficos que exploram mudanças no comportamento com a variação do valor de uma parâmetros, mantidos outros constantes.

Zé: Escolhi uma função específica para facilitar a manipulação por outros alunos do curso. O objetivo da (minha) função seria então familiarizar o usuário com conceitos interessantes de sistemas dinâmicos (como os gráficos que são gerados) e dar a ele a liberdade de explorar mudanças nos parâmetros e nas condições iniciais.

Pensei, ainda, em deixar a escolha ou não de uma função como opção, mas ao ver que existe uma série de passos um pouco chatos para construir essa função, preferi me focar na parte de exploração do comportamento.

No final peguei um sistema de equações bem didático (no meu ponto de vista) que apresenta um ciclo limite quando um parâmetro cruza o 0.

Plano B

Uma função capaz de calcular se é mais benéfico usar alcool ou gasolina em carros.

A entrada poderia ser algo como:

```
teste.ga(preço.alcool= , preço.gasolina= , km.por.dia= , carro="" ,)
```

o parâmetro carro terá algumas opções de consumo para alguns carros (penso agora, que carros populares), se for deixado em branco o valor usado será uma média padrão (consumo 70% maior para alcool, se não me engano)

O resultado será dado como um data frame com:

Melhor escolha	Economia Mensal (R\$)^	Economia anual (R\$)
Alcool / Gasolina	45	540

Paulo: bem mais simples, mas também cabível para um trabalho da disciplina. Vc pode tb incluir gráficos que mostrem a partri de que diferença de preço é vantajoso adotar um combustível e outro. Outra coisa legal é se vc incluir a possibilidade de misturar os dois combustíveis (supondo efeitos aidtivos, que não sei se valem :)).

Comentários

Bem interessante. Acho que especificar alguns tipos de carros faria um uso interessante de operadores lógicos e etc. Seria interessante no output dizer se o carro constava na base de dados ou se foi usada a média padrão.

— [Fabio de A. Machado](#) 2011/04/06 18:34

Trabalho final

Página de Ajuda

ode.exp

package:nenhum

R Documentation

Gráficos exploratórios de uma equação diferencial

Description

Análise gráfica do efeito das condições iniciais e alterações de um parâmetro em um sistema de equações diferenciais. A função retorna os valores da simulação, um gráfico do espaço de fases e, caso sejam utilizados vários valores do mesmo parâmetro, um diagrama de bifurcação.

Usage:

```
ode.exp=function(a,tempo,x.inicial,y.inicial,plots=FALSE)
```

Arguments:

a: Valores do parâmetro a, pode ser um número ou um vetor

tempo: Tempo final da simulação sendo utilizado um passo de 0.01

x.inicial: Valor inicial da variável de estado X

y.inicial: Valor inicial da variável de estado Y

plots: Um valor lógico indicando se as figuras deve ser feitas.

Details:

As simulações são feitas, utilizando a função ode do pacote deSolve, em cima do seguinte sistema de equações diferenciais:

$$dX/dt = -Y + X(a - X^2 - Y^2)$$

$$dY/dt = X + Y(a - X^2 - Y^2)$$

Onde X e Y são as variáveis de estado e a o parâmetro do sistema. Caso o parâmetro a seja um vetor, são realizadas múltiplas simulações sendo que cada simulação resulta em um gráfico. O gráfico do estado de fases usa a função plot para mostrar a relação entre os valores de X e Y, ao longo do tempo. Por outro lado, os valores de Y obtidos (eliminando um transiente inicial) com cada parâmetro são utilizados no diagrama de bifurcação.

Values:

Caso o parâmetro `a` seja um número, uma matriz é gerada com as seguintes colunas:

`time`: tempo (passos) da simulação

`X`: Valores da variável `X` em cada tempo

`Y`: Valores da variável `Y` em cada tempo

Caso o parâmetro `a` seja um vetor, uma lista é gerada contendo uma matriz (com as mesmas colunas descritas acima) para cada valor do vetor. Adicionalmente, até dois gráficos são gerados das entradas da função.

Authors:

José Eduardo Soubhia Natali

jesnatali@gmail.com

References:

Karline Soetaert, Thomas Petzoldt, R. Woodrow Setzer (2010). Solving Differential Equations in R: Package `deSolve` Journal of Statistical Software, 33(9), 1--25. URL <http://www.jstatsoft.org/v33/i09/>.

Monteiro, L. H. A. (2006). Sistemas Dinâmicos. Editora Livraria da Física, São Paulo, 2a edição.

See Also

'ode' do pacote `deSolve` para as simulações numéricas de equações diferenciais.

Examples

```
exemplo.1=ode.exp(a=-1,tempo=100,x.inicial=5,y.inicial=10,plots=TRUE) #Para  
um valor de a
```

```
exemplo.2=ode.exp(a=(-5:5),tempo=100,x.inicial=10,y.inicial=5,plots=TRUE)  
#Para múltiplos valores
```

Código da Função

```
### Projeto
```

```

ode.exp=function(a,tempo,x.inicial,y.inicial,plots=FALSE)
{
  require(deSolve)
  state=c(X=x.inicial,Y=y.inicial)
  tempos=seq(0,tempo,by=0.01)
  if(length(a)==1)#caso a seja um numero
  {
    #criando a funcao
    c.lim=function(t, state, a) {
      with(as.list(c(state,a)),{
        #taxa de mudanca
        dX= -Y+X*(a-X^2-Y^2)
        dY= X+Y*(a-X^2-Y^2)
        #retorna a taxa de mudanca
        list(c(dX,dY))
      })
    }
    dados=ode(y=state,times=tempos,func=c.lim,
parms=a)
    dados.f=na.omit(dados) #Quando o numero chega
    muito perto de 0 ele vira NaN,
    #esse omit tira esses valores sem modificar os resultados.
    Interessante que o na.omit tira os NaN
    if(plots==TRUE)
    {
      X11()
      par(bty="l")
      plot(dados[, "X"], dados[, "Y"], pch = "o",
xlab="Valores de X",ylab="Valores de Y", col="red", main="Espaco de fases")
    }
  }
  else#caso a seja um vetor
  {
    k=1
    for(i in a)
    {
      b=i
      c.lim=function(t, state, b) { #criando a funcao
        with(as.list(c(state,b)),{
          #taxa de mudanca
          dX= -Y+X*(b-X^2-Y^2)
          dY= X+Y*(b-X^2-Y^2)
          #retorna a taxa de mudanca
          list(c(dX,dY))
        })
      }
      dados=ode(y=state,times=tempos,func=c.lim,
parms=b)
      dados=na.omit(dados)
      if(k==1)
      {

```

```
        lista.dados=list(dados)
        min.x=min(dados[, "X"]) # Para que todos os
plots tenham as dimensoes corretas
        min.y=min(dados[, "Y"])
        max.x=max(dados[, "X"])
        max.y=max(dados[, "Y"])
    }
    else
    {
        lista.dados[[k]]=dados
        min.x[k]=min(dados[, "X"])
        min.y[k]=min(dados[, "Y"])
        max.x[k]=max(dados[, "X"])
        max.y[k]=max(dados[, "Y"])
    }
    k=k+1
}
minimo.x=floor(min(min.x))
minimo.y=floor(min(min.y))
maximo.x=ceiling(max(max.x))
maximo.y=ceiling(max(max.y))
if(plots==TRUE)
{
    X11()
    par(bty="l")
    for(i in 1:length(a))#plots multiplos
    {
        par(mfrow = c(1,2))
        par(ask=TRUE)
        plot(lista.dados[[i]][, "X"], lista.dados[[i]][, "Y"],
pch = "o", xlab="Valores de X",ylab="Valores de Y",
col="red",xlim=c(minimo.x,maximo.x),ylim=c(minimo.y,maximo.y), main="Espaco
de fases")

        temp.d=dim(lista.dados[[i]])[1]
        plot(rep(a[i],
times=length((temp.d-500):temp.d)),lista.dados[[i]][(temp.d-500):temp.d,"X"]
, xlab="Valores do parametro de entrada", ylab="Valores da variavel de
estado X", xlim=range(a), ylim= c(minimo.y,maximo.y), main="Diagrama de
Bifurcacao")

    } # fim do for
}# fim do if plots
dados.f=lista.dados
} # Fim do else length a > 1
par(mfrow=c(1,1))
par(ask=FALSE)
return(dados.f)
} # Fim da funcao
```

Arquivo da Função

[ode.exp](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2011:alunos:trabalho_final:jose:start

Last update: **2020/08/12 09:04**

