Equação Diferencial Ordinária

Uma equação diferencial é uma equação cuja incógnita é uma função. Essa função aparece na equação sob a forma das suas derivadas.

Veja um exemplo de uma equação diferencial simples:

$$ \frac{dy(t)}{dt}= 2 y(t) $$

Essa equação pode ser lida como “qual é a função y(t) cuja derivada é igual a duas vezes ela mesma?”

Resolver uma equação dessas pode ser bastante trabalhoso.[1] Mas se uma inspiração sobrenatural te disser que “a resposta é $\exp(2x)$”, nós podemos verificar que essa resposta está correta.

Se $y(t)=\exp(2x)$, a derivada de $y(t)$ é $2\exp(2x)$ (lembre da regra da cadeia), que é, de fato, duas vezes a própria $y(t)$.

Normalmente, escrevemos a EDO com a derivada de $y(t)$ do lado esquerdo. Um caso simples de EDO é aquele em que o lado direito não envolve a função y, e portanto depende só de t:

$$ \frac{dy(t)}{dt}= f(t) $$

Podemos solucionar essa equação da seguinte forma:

- $ dy = f(t) dt $
- $ \int{dy} = \int{f(t) dt} $

O que nos dá uma solução geral:

$$ y = \int{f(t) dt} $$

Um caso mais complicado é aquele em que a derivada da função depende tanto de y como de t. Escrevemos:

$$ \frac{dy(t)}{dt}= f(y, t) $$

Vamos retomar esse caso na sessão de soluções numéricas.

Uma EDO simples no Maxima

Vamos usar o Maxima para resolver uma função simples para nós. Lembre-se da EDO da sessão anterior, mas agora vamos trocar a constante 2 por um parâmetro r:

$$ \frac{dy(t)}{dt} = r y(t)$$

Nossa equação verbalmente colocado é: a taxa de variação instantânea da nossa variável de interesse é proporcional a ela própria. Ou seja: quanto maior o valor de y, maior a taxa de crescimento!

Para resolver isso no Maxima, use

```
'diff(y(t),t)=r*y(t);
ode2(%, y(t), t);
```

Essa equação parece familiar? Vamos resgatá-la mais a frente no curso: é a equação do modelo de crescimento populacional exponencial, a estrutura básica de muitos outro modelos. Faça o gráfico dessa função para $ r=0.2 $ e estado inicial igual a 10!

```
plot2d(10*exp(0.2*t),[t,0,20]);
```

Outra função simples

Vamos pensar em outro caso, onde a taxa de variação instantânea é positiva e tende a zero quando o valor da nossa variável aproxima a um. Em outras aumenta muito quando o valor e pequeno e muito pouco quando o valor aproxima-se a um.

$$ f(x)=f(x)* (1-f(x)) $$

```
'diff(f(t),t)=f(t)*(1-f(t));
ode2(%,f(t),t);
```

Talvez não reconheça essa função, faça exponenciação de ambos os lados que ela parecerá mais simpática. Ela é a solução do exemplo anterior multiplicada por uma expressão que funciona como um freio que aperta mais forte conforme chega perto de um. Elá é a base dos modelos logísticos populacionais.

Soluções Numéricas

Essas primeiras equações diferencial foram fáceis de resolver no Maxima . Mas não se acostume, nem sempre é assim. Muitas equações não tem soluções algébricas[2] e precisam ser resolvidas com métodos chamados de “força bruta”. São geralmente computacionalmente intensos, mas com um computador pessoal podemos fazer maravilhas… O processo básico é bem simples, muito parecido com o que fizemos para resolver as derivadas, mas existem muitas outras técnicas mais robustas.

Método de Euler

Ele é bastante simples, e consiste em fazer uma aproximação da curva usando as tangentes em diferentes pontos. Vamos pegar a função:

- $\frac{dN}{dt} = rN $ com r=2 e N(0) =20.

Como vimos:

- $\frac{dN}{dt} \approx \frac{N_{t + \Delta t} - N_t}{\Delta t}$

Podemos usar um intervalo de tempo arbitrário, por exemplo 0.1, o que nós dá :

- $\frac{N_{t + 0.1} - N_t}{0.1}\approx 2N$

Ou seja, sendo que $N(0)=20$, no tempo 0.1, temos que:

- $N_{t + 0.1} - N_t= 2N * 0.1$
- $N_{t + 0.1} = 2N_t*0.1 + N_t$
- $N_{t + 0.1} = 40 * 0.1 + 20 $
- $N(0.1) = 24 $

No tempo 0.2, temos que:

- $N_{0.1 + 0.1} - N_0.1 = 2N_0.1 * 0.1$
- $N_{0.1 + 0.1} = 2 * 24 *0.1 + 24$
- $N_{0.1 + 0.1} = 48 * 0.1 + 24 $
- $N(0.2) = 28.4 $

E assim por diante …. $\lim\limits_{x \to \infty} f(x)$ Note que quanto menor o intervalo de tempo que usamos melhor é a precisão da nossa aproximação, lembrando que $\Delta t \rightarrow 0 $ Podemos repetir isso para os próximos intervalos de tempo no .

```
f <- function (N, t)
{
        return (2*N)
}
# No tempo inicial, N vale 20:
N0 <- 20
# O passo de tempo eh dt. Vamos rodar ateh tmax
dt <- 0.1
tmax <- 2

euler <- function (f, N0, dt, tmax)
{
        # res vai retornar o vetor com todos os Ns
        res <- NULL
        N <- N0
        for (tempo in seq(0, tmax, dt))
        {
                N <- N + f(N,tempo)*dt
                res <- rbind(res, N)
        }
        return (res)
}

# Examine a solucao numerica:
numerica <- euler(f, N0, dt, tmax)
```

```
x<- seq(0, tmax, dt)
# A solucao correta da EDO:
plot(x, 20*exp(2*x), typ='l', col='green')
# Vamos comparar com a solucao numerica
points(x, numerica, col='red', pch=4, ce=0.2)
```

A aproximação foi boa? Tente repetir o mesmo código com dt = 0.01 e 0.001 e compare.

Integração Numérica no R

Não precisamos fazer todo o procedimento anterior para fazer a integração numérica no , existem soluções implementadas previamente que são muito mais eficientes e robusta que a nossa. Vamos integrar numericamente algumas equações usando o pacote *deSolve* e a função *ode*. Antes de tudo precisa instalar e carregar o pacote. Para instalar você pode usar o menu do RGui ou pela linha de comando, digite:

```
install.packages("deSolve")
```

Carregando o pacote e olhando o help da função *ode*:

```
library(deSolve)
?ode
```

Muito bem! Agora que já temos o pacote carregado, vamos integrar numericamente algumas funções, ou seja, calcular o valor para cada tempo infinitesimal dentro de um amplitude de valores.

Uma função simples

Vamos primeiro usar uma função simples:

$$ \frac{dy}{dt} = y-\frac{y^2}{K} $$

- 1. Primeiro criamos a função, com os seguintes parâmetros na função R:
  - o tempo, que depois definiremos com uma sequência numérica
  - a situação inicial da variável independente
  - os parâmetros da equação diferencial

```
  fy1 <- function(time,y,parms)
  {
   n=y[1]
   K=parms[1]
   dy.dt=n-(n^2/K)
   return(list(c(dy.dt)))
  }
```

- 2. Agora precisamos criar os parâmetros:

```
  prmt = 10
  y0 = 1
  st=seq(0.1,20,by=0.1)
```

- 3. Vamos resolver a nossa equação diferencial e graficá-la:

```
  res.fy1= ode(y=y0,times=st, fy1,parms=prmt)
  plot(res.fy1[,1], res.fy1[,2], type="l", col="red",lwd=2, xlab="tempo",
ylab="y")
```

Será que a função *ode* fez mágica? Ela apenas usa um método parecido com o de Euler, que vimos acima, para achar a solução numérica de uma ode.

Uma outra função simples

Agora nossa equação é:

$$\frac{dy}{dt}= y(ay^2 + by + r)$$

Veja o código abaixo:

```
fy2 <- function(time,y,parms)
  {
   n=y[1]
   a=parms[1]
   b=parms[2]
   r=parms[3]
   dy.dt=n*(a*n^2 + b*n + r)
   return(list(c(dy.dt)))
  }
prmt = c(a=-1,b=4, r=-1)
y0 = 1
st=seq(0.1,20,by=0.1)
res.fy2= ode(y=y0,times=st, fy2,parms=prmt)
plot(res.fy2[,1], res.fy2[,2], type="l", col="red",lwd=2, xlab="tempo",
ylab="y")
```

Agora é com vc.

- modifique a condição inicial e veja o que acontece: 0,5; 0,3; 0,01

Agora é realmente com vc.

Faça a integração numérica para as seguintes funções:

- 1. $\frac{dy}{dt} = y-y^2*f(t)$
    - sendo: $f(t)= 0.01 + 0.01 \sin(2\pi Mt)$;

    - M=15
- 2. $\frac{dn}{dt} = r(t)*n $
    - sendo: $r(t)= 0.1 - 100 sin(2 \pi t)$

maxima, equação diferencial, R

1)

existem pelo menos umas 3 matérias no IME relacionadas a isso!

2)

outras tem mais de uma solução

From:
http://ecovirtual.ib.usp.br/ -

Permanent link:
http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:math:eq_difr

Last update: **2016/05/10 07:19**