Maple 高级应用 www.maplesoft.com.cn

B3: Maple中的偏微分方程求解

©西希安工程模拟软件（上海）有限公司，2008

11.0 Maple中的微分方程求解器介绍

Maple中微分方程求解器使用领先的算法求解以下问题：

- 常微分方程 (ODEs): dsolve 命令用于求解线性和非线性ODEs, 初始值问题 (IVP), 以及边界值问题 (BVP)，可以通过参数项选择求符号解 (解析解) 或数值解。ODE Analyzer Assistant 微分方程分析器助手提供一个交互式用户界面方便用户求解 ODE 以及显示结果的图形。了解更多信息，参考帮助系统中的 dsolve, dsolve/numeric, 和 ODE Analyzer.

- 偏微分方程 (PDEs): pdsolve 命令用于求 PDEs 和含边界值问题的 PDEs 的符号解或数值解。使用Maple的PDE工具可以完成对PDE系统的结构分析和指数降阶处理。了解更多信息，参考帮助系统中的 pdsolve and pdsolve/numeric.

- 微分-代数方程 (DAEs)：dsolve/numeric 命令是符号-数值混合求解器，使用符号预处理和降阶技术，让Maple能够求解高指数的DAE问题。Maple内置三个求解器用于处理DAEs：1）修正的 Runge-Kutta Fehlberg 方法，2）Rosenbrock 方法，以及 3）修正的拓展后向差分隐式方法。

11.1 求解偏微分方程PDE问题(BVP和IVP)

Maple 求解经典力学难题的能力是非常著名的，它的数值和符号偏微分方程求解器是其中的重要工具。

例子：在不同的边界条件下，求波动方程的数值解、解析解、和图形解。

11.1.1 初始化

```
> restart;
  with(plots) :
```

下面的Maple代码定义了一个名为PX的程序，生成函数的周期展开。

```
> PX := proc(h::{algebraic,procedure},g::{range,name=range})
    local L, D, var;
    if type(g,'range') then L := lhs(g); D := rhs(g) - L;
```

```
    if not type(h,'procedure') then var := indets(h,'name');
    if nops(var) <> 1 then error "need to specify a variable"; end if;
    var := op(var); end if;
    else L := lhs(rhs(g));D := rhs(rhs(g)) - L;var := lhs(g); end if;
    if type(h,'procedure') then proc(x::algebraic) h(x - floor((x-L)/D)*D); end;
    else proc(x::algebraic) eval(h, var = x - floor((x-L)/D)*D); end;
    end if;
end:
```

11.1.2 数值解和图形解

一个空间变量的波动方程是：

> $pde := \frac{\partial^2}{\partial t^2}\, u(x,t) = \frac{\partial^2}{\partial x^2}\, u(x,t)$

$$pde := \frac{\partial^2}{\partial t^2}\, u(x,t) = \frac{\partial^2}{\partial x^2}\, u(x,t) \qquad (1)$$

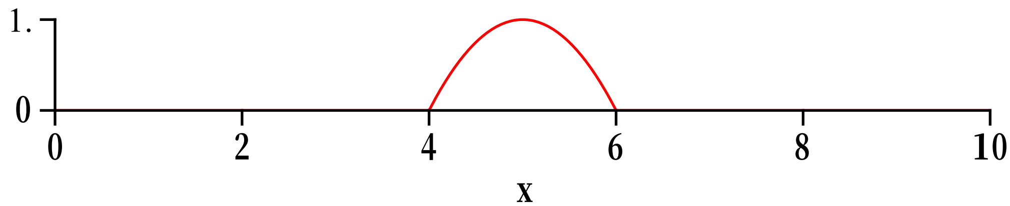
假设初始形状由下面的函数给出：

> $f := x \rightarrow piecewise(x < 4, 0, x < 6, (x-4)*(6-x), 0) :$
> $'f'(x) = f(x);$

$$f(x) = \begin{cases} 0 & x < 4 \\ (-4+x)\,(6-x) & x < 6 \\ 0 & otherwise \end{cases} \qquad (2)$$

对应的图形如下：

> $plot\big(f(x), x = 0..10, scaling = constrained, tickmarks = ([6, 2])\big);$



假设波动方程的初始条件和边界条件如下：

> ibc := [u(0, t) = 0, u(10, t) = 0, u(x, 0) = f(x), (D[2](u))(x, 0) = 0];

$$ibc := \left[u(0,t)=0,\ u(10,t)=0,\ u(x,0)=\begin{cases}0 & x<4\\ (-4+x)\,(6-x) & x<6\\ 0 & otherwise\end{cases},\ D_2(u)(x,0)=0\right] \tag{3}$$

Maple中的命令pdsolve将求解单变量演化方程（双曲和抛物）的数值解（有限差分）。

> SOL := pdsolve(pde, ibc, numeric, time = t, range = 0 ..10, spacestep = 0.1);

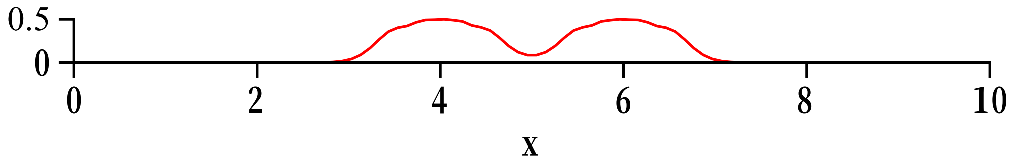
$$SOL := \textbf{module}()\ \textbf{export}\ plot, plot3d, animate, value, settings;\ ...\ \textbf{end module} \tag{4}$$

这个命令创建了一个模块，可以看到模块的输出函数是 plot, plot3d, animate, 和 value。

时间 $t = 1$ **的图形：**

> $SOL\text{:-}plot(t = 1, numpoints = 100, scaling = constrained, tickmarks = ([6, 2]), title = "图 2");$

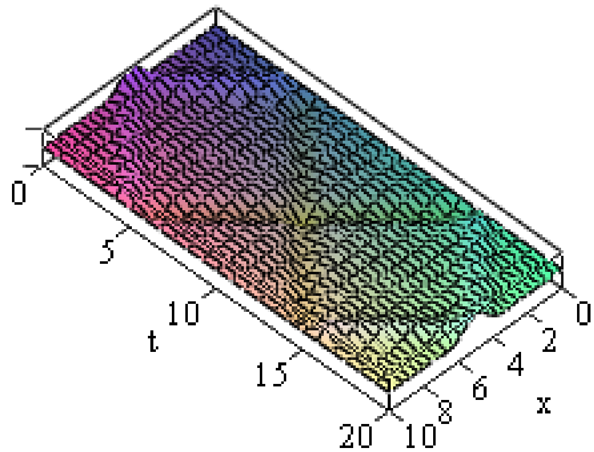
图 2



解的表面图：

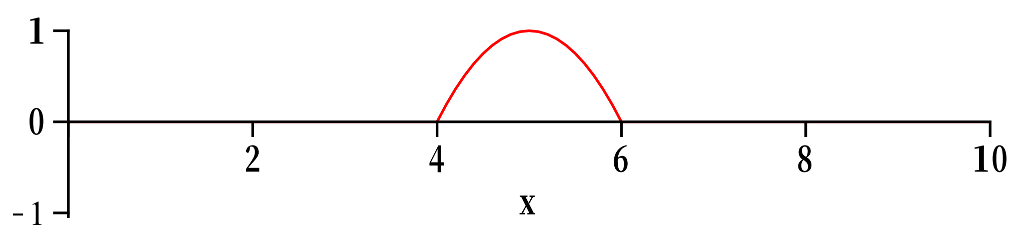
> $SOL\text{:-}plot3d(t = 0 ..20, x = 0 ..10, axes = box, scaling = constrained, tickmarks = ([6, 5, 2]), title = "Figure 3");$

Figure 3



动画：

```
> SOL:-animate(t = 20, frames = 60, scaling = constrained, tickmarks = ([6, 3]), title
      = "time = %f");
```



给出计算值 $U(x,t)$ 的函数：

> $U := rhs\left(\left((SOL :\text{-}\ value)(output = listprocedure)\right)_3\right)$

$$U := \textbf{proc}(\ )\ \ldots\ \textbf{end proc} \quad (5)$$

例如，$U(2.5, 3)$ 的解：

> $U(2.5, 3)$

$$0.364667328524451 \quad (6)$$

11.1.3 D'Alembert解析解

对于这个问题，D'Alembert 解的形式是：

$$u(x,t) = \frac{f(x+t) + f(x-t)}{2}$$

因此 $u(2.5, 3)$ **的精确值是：**

> $\frac{f(2.5+3)+f(2.5-3)}{2}$

$$0.3750000000 \tag{7}$$

在 $f(x)$ **区间** $[-10, 10]$ **上的奇次展开如下：**

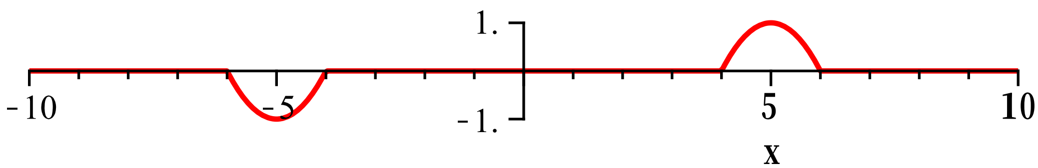
> $FOE := simplify\left(piecewise\left(x < 0, -f(-x), x < 10, f(x)\right)\right);$

$$FOE := \begin{cases} 0 & x \le -6 \\ (4+x)\,(6+x) & x \le -4 \\ 0 & x < 4 \\ -(-4+x)\,(x-6) & x < 6 \\ 0 & 6 \le x \end{cases} \tag{8}$$

通过【图4】验证。

> $plot(FOE, x = -10..10, color = red, thickness = 2, scaling = constrained, ytickmarks = 2,$
> $title = \text{"图 4"});$

图 4



$f(x)$ **的奇次周期展开如下：**

> $F := PX(FOE, x=-10..10)$

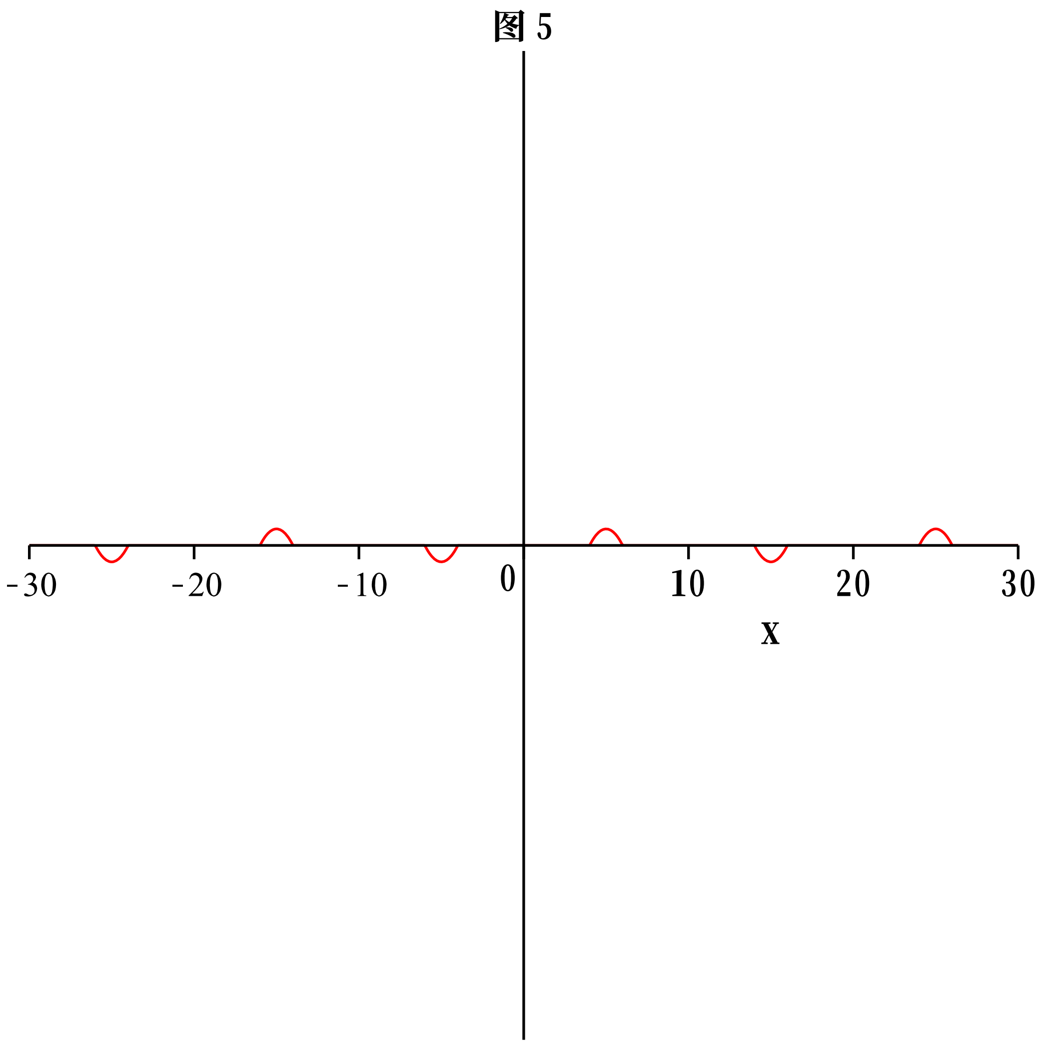
$F :=$ **proc**$(x::algebraic)$ (9)

$eval\big(piecewise(x <= -6, 0, x <= -4, (4+x)*(6+x), x < 4, 0, x < 6, -(-4+x)*(x-6), 6 <= x, 0), var = x - \text{floor}((x-L)/\text{D})*\text{D}\big)$

end proc

我们注意到输出是一个程序体，忽略Maple输出的细节，我们再次使用图形来验证工作，得到【图5】。

> $plot(F, -30..30, color=red, scaling=constrained, tickmarks=([6, [0]]), labels=([x, ``]), labelfont=([TIMES, ITALIC, 12]), view=([-30..30, -30..30]), title=$"图 5")
;

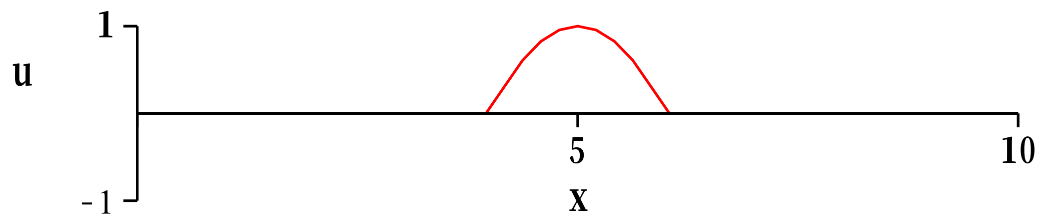


最后，通过下式组合解$u(x, t) = \frac{1}{2}\left(F(x+t)+F(x-t)\right)$

> $UDAL := \frac{F(x+t)+F(x-t)}{2}$:

U**的输出是一个分段函数的复杂组合，但是下面的动画显示了在指定区间内波的相应运动。**

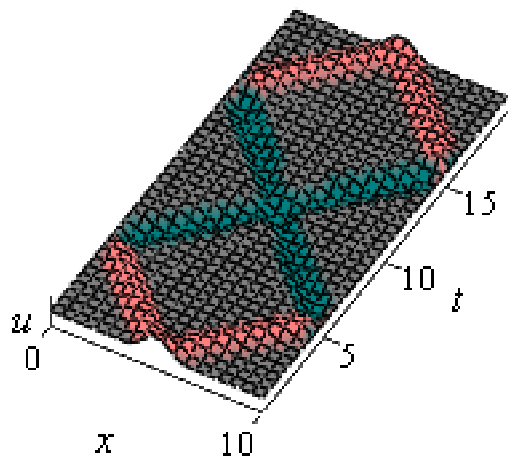
> $animate(UDAL, x=0..10, t=0..20, frames=61, color=red, scaling=constrained, tickmarks=([3,2]), labels=([x, u\ \ ]), labelfont=([TIMES, ITALIC, 12]));$



【图6】是解的表面，与动画提供的信息一致。

$> plot3d\left(UDAL, x=0..10, t=0..20, axes=frame, color=\left(\left[\frac{1+\mathrm{signum}(UDAL)}{2}, 0.5, 0.5\right]\right), labels=([x, t, u]), tickmarks=([2, [5, 10, 15], 0]), grid=([20, 40]), orientation=([-60, 45]), scaling=constrained, labelfont=([TIMES, ITALIC, 12]), title=\text{"图 6"}\right);$

图 6



11.1.4 扩展（可选）：传播和Klein-Gordon方程

10.2.4.1 *数值解*

波和Klein-Gordon方程分别表示为：

> $wave := pde$

$$wave := \frac{\partial^2}{\partial t^2} u(x, t) = \frac{\partial^2}{\partial x^2} u(x, t) \quad \textbf{(10)}$$

> $KG := \frac{\partial^2}{\partial t^2} u(x, t) = \frac{\partial^2}{\partial x^2} u(x, t) - u(x, t)$

$$KG := \frac{\partial^2}{\partial t^2} u(x, t) = \frac{\partial^2}{\partial x^2} u(x, t) - u(x, t) \quad \textbf{(11)}$$

两者之间的区别是Klein-Gordon方程中有$u(x, t)$项。

为了说明两个方程解之间的查表，考虑下面的初始扰动：

> $L := 40:$
$f := piecewise(x < L/2-1, 0, x < L/2+1, (x-(L/2-1))*(L/2+1-x), 0);$

$$f := \begin{cases} 0 & x < 19 \\ (-19+x)\,(21-x) & x < 21 \\ 0 & otherwise \end{cases} \tag{12}$$

传播长度设定为2 0，端点固定。初始条件和边界条件的方程是：

> $ibc := [u(0,t)=0, u(L,t)=0, u(x,0)=f, D[2](u)(x,0)=0];$

$$ibc := \left[u(0,t)=0, u(40,t)=0, u(x,0) = \begin{cases} 0 & x < 19 \\ (-19+x)\,(21-x) & x < 21 \\ 0 & otherwise \end{cases}, \mathrm{D}_2(u)(x,0)=0 \right] \tag{13}$$

方程的数值解分别是：

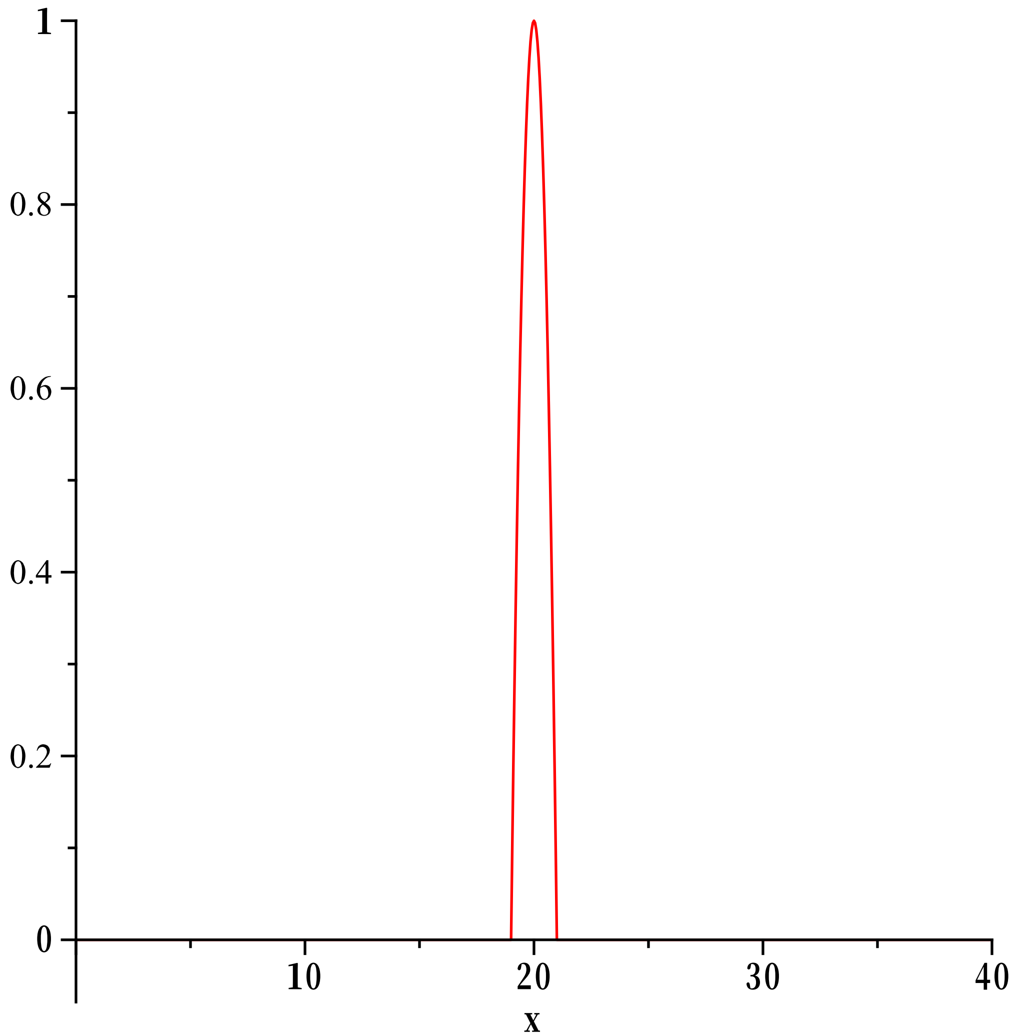
> $SW := pdsolve(wave, ibc, numeric, time=t, spacestep=.05):$
$SKG := pdsolve(KG, ibc, numeric, time=t, spacestep=.05):$

方程的动画分别是：

> $p1 := SW\text{:-}animate(x=0..L, t=0..L/2, frames=50, color=red):$
$p2 := SKG\text{:-}animate(x=0..L, t=0..L/2, frames=50, color=blue):$

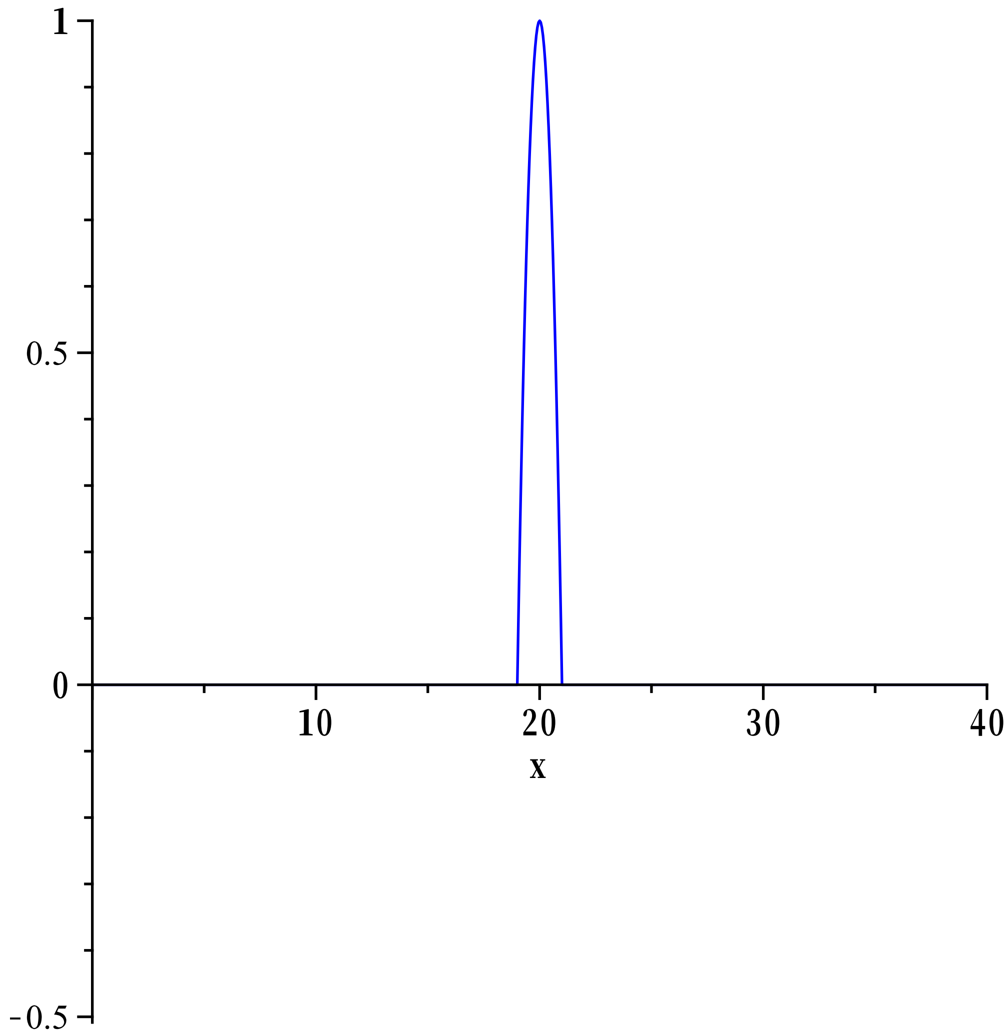
波动方程的动画是：

> $p1;$



Klein-Gordon传播方程的动画是：

```
> p2;
```



10.2.4.2 *解析解*

Klein-Gordon方程产生分离的变量。因此，假设解的形式如下：

> $U := X(x)\,T(t):$

因此Klein-Gordon方程，这里 $c=1$**，变为：**

> $q1 := KG\Big|_{u(x,t)=U};$

$$q1 := X(x)\left(\frac{d^2}{dt^2}T(t)\right) = \left(\frac{d^2}{dx^2}X(x)\right)T(t) - X(x)\,T(t) \tag{14}$$

除以 $u(x,t)=X(x)\,T(t)$ **得到：**

> $q2 := expand\left(\frac{q1}{U}\right)$

$$q2 := \frac{\frac{d^2}{dt^2} T(t)}{T(t)} = \frac{\frac{d^2}{dx^2} X(x)}{X(x)} - 1 \tag{15}$$

-1 移到左边：

> $q3 := q2 + (1 = 1)$

$$q3 := \frac{\frac{d^2}{dt^2} T(t)}{T(t)} + 1 = \frac{\frac{d^2}{dx^2} X(x)}{X(x)} \tag{16}$$

基于以前的经验，我们引入Bernoulli分离常数 $-\lambda^2$，这两个常微分方程变为：

> $q4 := rhs(q3) = -\lambda^2$

$$q4 := \frac{\frac{d^2}{dx^2} X(x)}{X(x)} = -\lambda^2 \tag{17}$$

> $q5 := lhs(q3) = -\lambda^2$

$$q5 := \frac{\frac{d^2}{dt^2} T(t)}{T(t)} + 1 = -\lambda^2 \tag{18}$$

在不同的假设下，得到边界条件：

$$X(0) = X(l) = 0$$

使用其中的第一个得到：

> $dsolve(\{q4, X(0) = 0\}, X(x))$

$$X(x) = \_C1 \sin(\lambda x) \tag{19}$$

使用其中的第二个得到：

> $Xn := \sin\left(\frac{n \pi x}{l}\right)$

$$Xn := \sin\left(\frac{n \pi x}{l}\right) \tag{20}$$

第二个常微分方程变为：

> $q6 := eval(q5, \lambda = n * \pi / l);$

$$\tag{21}$$

$$q6 := \frac{\frac{d^2}{dt^2} T(t)}{T(t)} + 1 = -\frac{n^2 \pi^2}{l^2} \qquad (21)$$

初始条件$u_t(x,0)=0$得到条件$T'(0)=0$，然后得到解。

```
> q7 := rhs(dsolve({q6, D(T)(0)=0}, T(t)));
```

$$q7 := \_C2 \cos\left(\frac{\sqrt{l^2+n^2\pi^2}\; t}{l}\right) \qquad (22)$$

可以写为：

```
> Tn := eval(q7, _C2=1);
```

$$Tn := \cos\left(\frac{\sqrt{l^2+n^2\pi^2}\; t}{l}\right) \qquad (23)$$

使用常用的技术，前一段落中的BVP的数值解可以表示为级数：

$$u(x,t) = \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{l}\right)\cos\left(\sqrt{1+\left(\frac{n^{\pi}}{l}\right)^2}\; t\right)$$

这里系数b_n由下面给出：

```
> qb := eval(2/l*Int(f*Xn, x=0..l), l=L);
```

$$qb := \frac{1}{20}\int_0^{40} \left(\begin{cases} 0 & x < 19 \\ (-19+x)(21-x) & x < 21 \\ 0 & otherwise \end{cases}\right) \sin\left(\frac{1}{40} n\pi x\right) dx \qquad (24)$$

最终得到：

```
> b := value(qb) assuming n :: integer;
```

$$b := -\frac{1}{n^3\pi^3}\left(160\left(n\pi \sin\left(\frac{19}{40}n\pi\right) - 40\cos\left(\frac{19}{40}n\pi\right) + n\pi\sin\left(\frac{21}{40}n\pi\right) + 40\cos\left(\frac{21}{40}n\pi\right)\right)\right) \qquad (25)$$

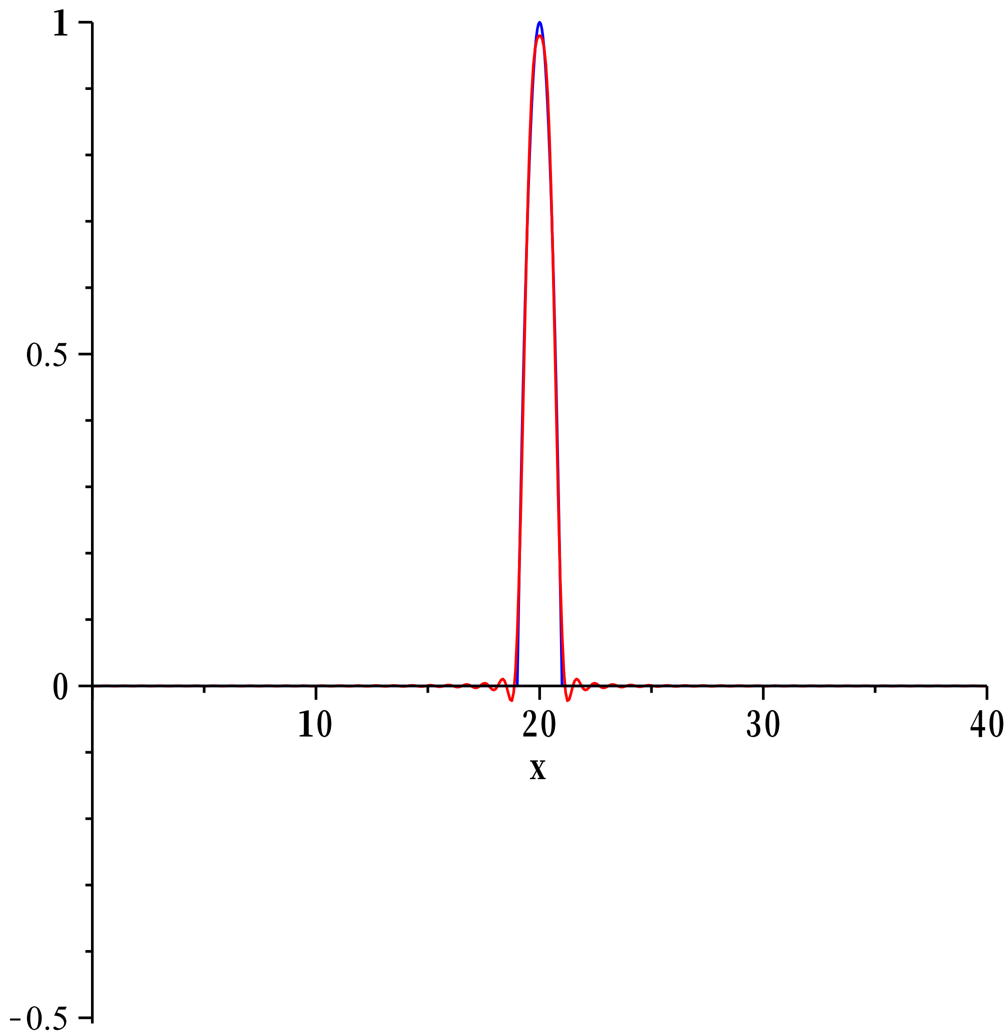
得到一个近似部分和：

```
> Uxt := eval(sum(b*Xn*Tn, n=1..100), l=L) :
```

叠加解后的动画：

```
> p3 := animate(Uxt, x=0..L, t=0..L/2, frames=50, color=red, numpoints=500) :
```

display([*p2*, *p3*]);



可以发现数值解和级数近似值几乎等同。

11.1.5 求解初始值偏微分方程问题（热传导方程）

数值解

```
> pde := diff(u(x, t), t) = diff(u(x, t), x, x);
```

$$pde := \frac{\partial}{\partial t} u(x, t) = \frac{\partial^2}{\partial x^2} u(x, t) \tag{26}$$

初始条件：

```
> ibc := [(D[1](u))(0, t)+a*u(0, t) = h(t), u(x, 0) = 0];
```

$$ibc := \left[D_1(u)(0, t) + a\, u(0, t) = h(t), u(x, 0) = 0\right] \tag{27}$$

可以得到数值解，但是我们需要定义 a 和 h 的值，以及提供第二个边界条件：

```
> ibc1:=[D[1](u)(0,t)+2*u(0,t)=cos(t),u(x,0)=0,u(0,t)=1];
```

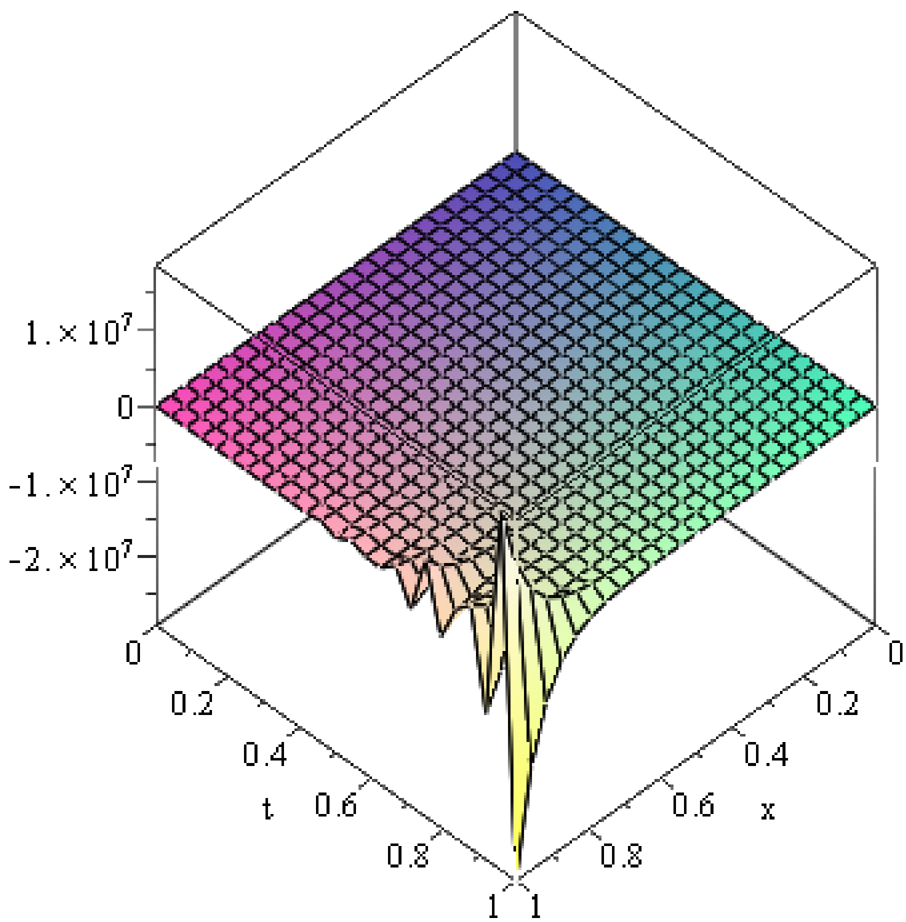
$$ibc1 := [D_1(u)(0,t) + 2\,u(0,t) = \cos(t),\ u(x,0) = 0,\ u(0,t) = 1] \quad (28)$$

```
> pds:=pdsolve(pde,ibc1,numeric,time=t,range=0..1);
```

pds := **module**( ) **export** *plot, plot3d, animate, value, settings;* ... **end module** (29)

我们可以做其他计算，例如生成解的三维图形：

```
> pds:-plot3d(t=0..1,x=0..1,axes=boxed);
```



关于偏微分方程组数值解的更多信息，请参阅下面的帮助文件：

```
> ?pdsolve,numeric
```

上面的帮助页中包含了处理热传导方程的示例。

解析解

我们可以使用 pdsolve （无需参数项）命令计算偏微分方程的解析解：

```
> gsol:=pdsolve(pde);
```

$$gsol := \left(u(x,t) = \_F1(x)\,\_F2(t)\right) \&where \left[\left\{\frac{d^2}{dx^2}\_F1(x) = \_c_1\,\_F1(x), \frac{d}{dt}\_F2(t) = \_c_1\,\_F2(t)\right\}\right] \tag{30}$$

结果可以拆分为方程（eq）和条件（conds）：

```
> eq:=op(gsol)[1];
  conds:=op(op(gsol)[2]);
```

$$eq := u(x,t) = \_F1(x)\,\_F2(t)$$

$$conds := \left\{\frac{d^2}{dx^2}\_F1(x) = \_c_1\,\_F1(x), \frac{d}{dt}\_F2(t) = \_c_1\,\_F2(t)\right\} \tag{31}$$

我们可以用 dsolve 命令求解常微分方程组，然后使用结果对 eq 求值：

```
> conditions:=dsolve(conds);
```

$$conditions := \left\{\_F1(x) = \_C2\, e^{\sqrt{\_c_1}\,x} + \_C3\, e^{-\sqrt{\_c_1}\,x}, \_F2(t) = \_C1\, e^{\_c_1 t}\right\} \tag{32}$$

```
> solution:=eval(eq,conditions);
```

$$solution := u(x,t) = \left(\_C2\, e^{\sqrt{\_c_1}\,x} + \_C3\, e^{-\sqrt{\_c_1}\,x}\right)\_C1\, e^{\_c_1 t} \tag{33}$$

我们可以判定结果是否满足偏微分方程：

```
> pdetest(solution,pde);
```

$$0 \tag{34}$$

然后我们可以调整常数 _c1, _C1, _C2 和 _C3 满足期望的条件。

关于 pdsolve 命令的更多信息，请参考下面的帮助页：

```
> ?pdsolve
```

其他帮助

pdetest, pdsolve