

符号计算又称计算机代数,通俗地说就是用计算机推导数学公式,如对表达式进行因式分解、化简、微分、积分、解代数方程、求解常微分方程等。与数值计算不同,符号计算对操作对象不进行离散化和近似化处理,因此其计算结果是完全准确而没有误差的。

MATLAB 是通过调用第三方软件进行符号计算的,在 MATLAB R2008b(MATLAB 7.7)之前,MATLAB 符号计算引擎是基于 Maple 内核的,从 MATLAB R2008b 开始,符号计算引擎开始采用 MuPAD 内核。对于普通的 MATLAB 用户,符号计算引擎的改变并没有带来什么不同,用户只需要调用 MATLAB 函数进行符号运算即可。本章结合具体案例介绍 MATLAB 符号计算。

5.1 符号对象和符号表达式

符号对象和符号表达式是 MATLAB 进行符号计算的基本元素,要进行符号计算首先要创建符号对象,对符号对象进行数学上的运算操作便得到符号表达式。

5.1.1 创建符号对象

MATLAB 中提供了 `sym` 和 `syms` 函数,用来创建符号对象,其用法参见下例。

【例 5.1-1】 调用 `sym` 和 `syms` 函数创建符号对象。

```
>> a = sym('6.01');           % 定义符号常数
>> b = sym('b','real');       % 定义实数域上的符号变量
>> A = [1, 2; 3, 4];          % 定义数值矩阵
>> B = sym(A);                 % 把数值矩阵转为符号矩阵
>> C = sym('c%d%d',[3,4])     % 定义 3 行 4 列的符号矩阵
C =
[ c11, c12, c13, c14]
[ c21, c22, c23, c24]
[ c31, c32, c33, c34]
>> syms x y                    % 同时定义多个复数域上的符号变量
>> syms z positive              % 定义正实数域上的符号变量
>> syms f(x,y)                 % 定义符号函数
>> f(x,y) = x + y^2;           % 指定符号函数表达式
```

```

>> c = f(1, 2) % 计算符号函数在(1,2)点处的函数值
c =
    5
>> zv = solve(z^2 == 1, z) % 求方程 z^2 = 1 的解(只有大于零的解)
zv =
    1
>> syms z % 撤销对符号变量取值域的限定,将其恢复为复数域上的符号变量
>> zv = solve(z^2 == 1, z) % 求方程 z^2 = 1 的解(有两个解)
zv =
    1
   -1

```

由例 5.1-1 可知, `sym` 函数可用来创建单个符号对象, 而 `syms` 函数可用来创建多个符号对象。关于 `sym` 和 `syms` 函数的详细用法, 读者可以参考其帮助文档。

5.1.2 符号变量取值域的限定

从例 5.1-1 可以看出, MATLAB 中定义的符号变量有取值域的限定, 不同的取值域会对符号计算的结果造成不同影响, 默认情况下均为复数域。在调用 `sym` 或 `syms` 函数定义符号变量时, 可通过附加 `real` 或 `positive` 改变符号变量的取值域, 除此之外, 还可调用 `assume` 和 `assumeAlso` 函数对符号变量取值域实行更为精细的控制。

【例 5.1-2】 解不等式 $x^2 > 12$, 其中 x 为整数变量, 并满足 $0 < x < 5$ 。

```

>> syms x % 定义符号变量
>> assume(x > 0 & x < 5); % 对符号变量的取值域进行限定, 0 < x < 5
>> assumeAlso(x, 'integer'); % 对符号变量的取值域增加别的限定, x 取整数
>> assumptions(x) % 查看符号变量取值域的限定
ans =
[ in(x, 'integer'), 0 < x, x < 5]
>> result = solve(x^2 > 12) % 求解不等式
result =
    4
>> syms x % 重新定义符号变量, 以撤销对符号变量取值域的限定

```

若需撤销对符号变量取值域的限定, 将其恢复为复数域上的符号变量, 只需用命令“`syms 变量名`”重新定义符号变量即可。

【说明】 在 MATLAB R2018a 及其以前的版本中, 若需撤销对符号变量取值域的限定, 将其恢复为复数域上的符号变量, 可用命令“`syms 变量名 clear`”实现, 单纯用命令“`clear 变量名`”只是从 MATLAB 工作区中删除了变量, 并没有改变符号计算引擎 MuPAD 中变量取值域的限定。

5.1.3 创建符号表达式

对符号对象进行各种运算(算术运算、关系运算、逻辑运算), 即可创建符号表达式。

1. 符号计算中的算术运算与转置

由于 MATLAB 采用了重载(overload)技术, 使得用来构成符号表达式的运算符, 无论在拼写还是在使用方法上, 都与数值计算中的运算符完全相同。常用的算术运算符与转置符如

表 5.1-1 所示。

表 5.1-1 符号计算中的算术运算符与转置符

运 算 符	说 明	运 算 符	说 明
+	加	. *	点乘
-	减	. /	右点除
*	乘	. \	左点除
/	右除	. ^	点乘方
\	左除	'	共轭转置
^	乘方	. '	非共轭转置

【例 5.1-3】 在符号对象的基础上通过算术运算创建符号表达式。

```
>> syms a b c x y z           % 定义多个符号变量
>> f1 = a*x^2+b*x-c;         % 创建符号表达式 f1
>> f2 = sin(x)*cos(y);       % 创建符号表达式 f2
>> f3 = (x+y)/z;             % 创建符号表达式 f3
>> f4 = [x+1, x^2; x^3, x^4] % 创建符号表达式矩阵 f4
f4 =
[ x + 1, x^2]
[ x^3, x^4]
>> f5 = f4'                  % 符号表达式矩阵的共轭转置(')
f5 =
[ conj(x) + 1, conj(x)^3]
[ conj(x)^2, conj(x)^4]
>> f6 = f4.'                % 符号表达式矩阵的转置('.')
f6 =
[ x + 1, x^3]
[ x^2, x^4]
```

2. 符号计算中的关系运算

符号计算中的关系运算符如表 5.1-2 所示。

表 5.1-2 符号计算中的关系运算符

运 算 符	说 明	运 算 符	说 明
==	等于	<=	小于或等于
>=	大于或等于	<	小于
>	大于	~=	不等于

3. 符号计算中的逻辑运算

符号计算中的逻辑运算符如表 5.1-3 所示。

表 5.1-3 符号计算中的逻辑运算符

运 算 符	说 明
	逻辑或
&	逻辑与
~	逻辑非
xor	逻辑异或

【例 5.1-4】 在符号对象的基础上通过算术运算、关系运算和逻辑运算创建符号表达式。

```
>> syms x y                % 定义符号变量
>> f1 = abs(x) >= 0        % 创建符号表达式
f1 =
0 <= abs(x)
>> f2 = x^2 + y^2 == 1     % 创建符号表达式
f2 =
x^2 + y^2 == 1
>> f3 = ~(y - sqrt(x) > 0) % 创建符号表达式
f3 =
~0 < y - x^(1/2)
>> f4 = x > 0 | y < -1     % 创建符号表达式
f4 =
0 < x | y < -1
>> f5 = x > 0 & y < -1     % 创建符号表达式
f5 =
0 < x & y < -1
```

通过算术运算、关系运算和逻辑运算创建的符号表达式是由逻辑运算符连接的一系列等式或不等式。利用 `isAlways` 函数或 `logical` 函数可判断等式或不等式是否成立,利用 `isequaln` 函数可判断两个符号表达式是否相等,它们均返回逻辑型结果。

【例 5.1-5】 `isAlways`、`logical` 和 `isequaln` 函数的用法示例。

```
>> syms x                % 定义符号变量
>> f = abs(x) >= 0;      % 创建符号表达式
>> result1 = isAlways(f) % 判断不等式 |x|>=0 是否成立
result1 =
1
>> result2 = isequaln(abs(x), x) % 判断 |x| 是否等于 x
result2 =
0
>> assume(x > 0);        % 限定 x > 0
>> result3 = isequaln(abs(x), x) % 重新判断 |x| 是否等于 x
result3 =
1
>> syms x                % 撤销对符号变量取值域的限定
```

5.1.4 符号表达式的常用运算

这里要介绍的符号表达式的常用运算包括:因式(或因子)分解、合并同类项、对指定项展开、提取符号多项式系数、提取分式的分子和分母、对符号表达式进行化简、对符号分式进行约分、复合函数、嵌套多项式等,用到的 MATLAB 函数如表 5.1-4 所示。

表 5.1-4 符号表达式的常用运算函数

函 数 名	说 明	函 数 名	说 明
<code>factor</code>	因式(或因子)分解	<code>horner</code>	嵌套多项式
<code>collect</code>	合并同类项	<code>simplify</code>	对符号表达式进行化简
<code>expand</code>	对指定项展开	<code>simplifyFraction</code>	对符号分式进行约分

续表

函 数 名	说 明	函 数 名	说 明
combine	把相同的代数结构结合在一起	compose	复合函数
coeffs	提取符号多项式系数	finverse	反函数
numden	提取分式的分子和分母	pretty	美化显示符号表达式

【例 5.1-6】 因式(或因因子)分解。

(1) 对 $f = x^3 - y^3$ 进行因式分解; (2) 对符号数 12345678901234567890 进行因子分解。

```
>> syms x y % 定义符号变量
>> f = factor(x^3 - y^3) % 对符号表达式进行因式分解
f =
[ x - y, x^2 + x*y + y^2]
>> fa = factor(sym('12345678901234567890')) % 对符号数进行因子分解
fa =
[ 2, 3, 3, 5, 101, 3541, 3607, 3803, 27961]
```

【例 5.1-7】 把 $f = (x + y)(x^2 + y^2 + 1)$ 按变量 y 合并同类项。

```
>> syms x y % 定义符号变量
>> f = (x + y) * (x^2 + y^2 + 1); % 定义符号表达式
>> collect(f, y) % 对符号表达式按变量 y 合并同类项
ans =
y^3 + x*y^2 + (x^2 + 1)*y + x*(x^2 + 1)
```

【例 5.1-8】 把符号表达式展开。

(1) $f_1 = \cos(x + y)$; (2) $f_2 = (a + b)e^{(a-b)^2}$ 。

```
>> syms x y a b % 定义符号变量
>> f = [cos(x + y); (a + b) * exp((a - b)^2)]; % 定义符号表达式向量
>> expand(f) % 把符号表达式展开
ans =
cos(x) * cos(y) - sin(x) * sin(y)
a * exp(a^2) * exp(b^2) * exp(-2 * a * b) + b * exp(a^2) * exp(b^2) * exp(-2 * a * b)
```

【例 5.1-9】 对符号表达式进行化简。

(1) $f_1 = \sqrt{\frac{4}{x^2} + \frac{4}{x} + 1}$; (2) $f_2 = \cos(3\arccos(x))$ 。

```
>> syms x % 定义符号变量
>> f1 = sqrt(4/x^2 + 4/x + 1); % 定义符号表达式
>> g1 = simplify(f1) % 按默认设置进行化简
g1 =
((x + 2)^2/x^2)^(1/2)
>> g2 = simplify(f1, 'IgnoreAnalyticConstraints', 1) % 忽略分析约束进行化简
g2 =
(x + 2)/x
>> pretty(g2) % 把符号表达式显示为数学公式形式
x + 2
-----
```

```

x
>> f2 = cos(3 * acos(x));           % 定义符号表达式
>> g3 = simplify(f2, 'Steps', 4)    % 进行4步化简
g3 =
    4 * x^3 - 3 * x

```

【说明】 pretty 函数用来美化符号表达式在屏幕上的显示样式,把符号表达式显示为数学公式的形式。在 MATLAB R2006b 及其以后的版本中,用户可利用实时编辑器窗口进行符号计算,结果展示更为直观,如图 5.1-1 所示。

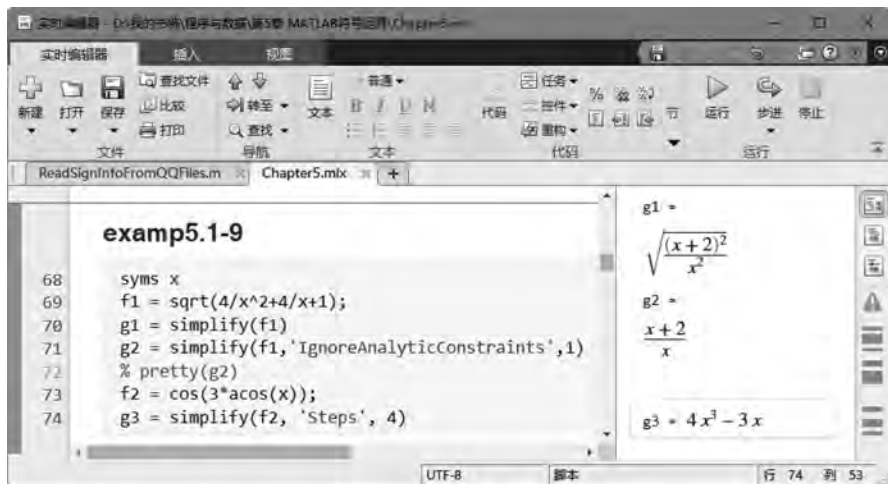


图 5.1-1 MATLAB 实时编辑器窗口

【例 5.1-10】 已知 $f(x) = e^x$, $g(x) = \sin x$, 求复合函数 $y = f(g(x))$, 并计算 $f(g(\pi))$ 。

```

>> syms f(x) g(x)           % 定义符号函数
>> f(x) = exp(x);           % 定义符号函数表达式
>> g(x) = sin(x);           % 定义符号函数表达式

>> y1(x) = f(g(x))          % 求复合函数(方法 1)
y1(x) =
    exp(sin(x))

>> y2(x) = compose(f,g)     % 求复合函数(方法 2)
y2(x) =
    exp(sin(x))

>> y = y1(pi)               % 计算复合函数在 x = pi 处的函数值
y =
    1

```

【例 5.1-11】 求符号函数 $f(x) = e^{\sin x}$ 的反函数。

```

>> syms f(x)               % 定义符号函数
>> f(x) = exp(sin(x));     % 定义符号函数表达式
>> g(x) = finverse(f(x))   % 求符号函数的反函数
g(x) =
    asin(log(x))

```

5.1.5 符号运算中的转换操作

1. 符号数与数值型数(或字符)的转换

当利用符号计算得到计算结果时,有时需要将其转化为数值型结果,以便在后续数值计算中加以利用。符号数与数值型数(或字符)的转换函数如表 5.1-5 所示。

表 5.1-5 符号数与数值型数(或字符)的转换函数

函 数 名	说 明	函 数 名	说 明
sym	创建符号对象	int8, int16, int32, int64	把符号矩阵转为有符号整型矩阵
double	把符号矩阵转为双精度矩阵	uint8, uint16, uint32, uint64	把符号矩阵转为无符号整型矩阵
eval	执行 MATLAB 运算	poly2sym	根据系数向量得到符号多项式
single	把符号矩阵转为单精度矩阵	sym2poly	根据符号多项式得到系数向量
vpa	按指定的有效数字位数来显示符号数值对象	char	把符号对象转为字符串

【例 5.1-12】 计算符号函数 $f(x)=\ln(5.2)e^x$ 在 $x=3$ 处的函数值。

```
>> syms f(x) % 定义符号函数
>> f(x) = log(sym(5.2)) * exp(x); % 指定符号函数表达式
>> y = f(3) % 计算符号函数在 x = 3 处的函数值
y =
exp(3) * log(26/5)
>> y1 = double(y) % 把符号数转为双精度数
y1 =
33.1142
>> y2 = vpa(y,10) % 以 10 位有效数字形式显示符号数
y2 =
33.1141937
>> x = 3; % 指定 x 的值
>> y3 = eval(f) % 执行 MATLAB 运算,得到函数值
y3 =
33.1142
```

2. 符号表达式中的变量替换

MATLAB 中提供了 subs 函数,用来对符号表达式中的变量(或符号项)进行替换。

【例 5.1-13】 把符号表达式 $f=a\sin(x)+b$ 中的 a 和 b 分别换为 2 和 5,并把 $\sin(x)$ 换为 $\ln(y)$ 。

```
>> syms a b x y % 定义符号变量
>> f = a * sin(x) + b; % 定义符号表达式
>> f1 = subs(f, sin(x), log(y)) % 符号项替换
f1 =
b + a * log(y)
% 变量替换方式一
```

```
>> f2 = subs(f1,[a,b],[2,5])           % 同时替换变量 a 和 b 的值
f2 =
    2 * log(y) + 5
% 变量替换方式二
>> f3 = subs(f1,{a,b},{2,5})           % 同时替换变量 a 和 b 的值
f3 =
    2 * log(y) + 5
```

【例 5.1-14】 把符号表达式 $f=a\sin(x)+b$ 中的 a 和 b 分别换为 2 和 5, 并求 f 在 $x=1, 2, 3$ 三点处的值。

```
>> syms a b x                           % 定义符号变量
>> f = a * sin(x) + b;                  % 定义符号表达式
>> y = subs(f, {a,b,x}, {2, 5, 1:3})    % 同时替换多个符号变量的值
y =
    [ 2 * sin(1) + 5, 2 * sin(2) + 5, 2 * sin(3) + 5]
>> y = double(y)                        % 将计算结果转为双精度值
y =
    6.6829    6.8186    5.2822
```

3. 将符号表达式转为函数

当通过符号计算得到一个表达式时, 希望把它转化成关于其中某个变量的函数, 这里的函数可以是符号函数, 也可以是匿名函数或 M 文件函数。MATLAB 中的 `symfun` 函数用来将符号表达式转为符号函数, `matlabFunction` 函数用来将符号表达式转为匿名函数或 M 文件函数。

【例 5.1-15】 把符号表达式 $a\sin(x)+b$ 转为关于 x 的符号函数 $f(x)$, 并求其在 $x=1, 2, 3$ 三点处的函数值。

```
>> syms a b x
>> f(x) = symfun(a * sin(x) + b, x);    % 把符号表达式转为符号函数
>> y = f(1:3)
y =
    [ b + a * sin(1), b + a * sin(2), b + a * sin(3)]
```

【例 5.1-16】 创建符号表达式 $f=a(x+b)^c+d$, 并将 a, b, c, d 分别替换为 2, -1, 1/2, 3, 并将替换后表达式转为 x 的函数 $f(x)$, 然后求函数在 $x=10$ 处的函数值。

```
>> syms a b c d x                       % 定义符号变量
>> f = a * (x + b)^c + d;                % 定义符号表达式
>> g = subs(f, {a,b,c,d}, {2, -1, sym(1/2), 3}); % 同时替换多个变量
>> FunFromSym1 = matlabFunction(g)        % 将符号表达式转为匿名函数
FunFromSym1 =
    @(x)sqrt(x-1.0). * 2.0 + 3.0
>> y = FunFromSym1(10)                   % 调用匿名函数计算函数值
y =
    9
% 将符号表达式转为 M 文件函数 FunFromSym2.m
>> matlabFunction(g, 'file', [pwd, '\FunFromSym2.m'], ...
    'vars', {'x'}, 'outputs', {'y'});
>> y = FunFromSym2(10)                   % 调用 M 文件函数计算函数值
y =
    9
```

本例中由符号表达式转化成的 M 文件函数 FunFromSym2.m 的代码如下：

```
function y = FunFromSym2(x)
% FUNFROMSYM2
%     Y = FUNFROMSYM2(X)

%     This function was generated by the Symbolic Math Toolbox version 8.5.
%     12 - Apr - 2021 09:58:54

y = sqrt(x-1.0) .* 2.0 + 3.0;
```

5.1.6 符号函数绘图

根据符号表达式或符号函数进行绘图的 MATLAB 函数如表 5.1-6 所示。

表 5.1-6 根据符号表达式或符号函数进行绘图的 MATLAB 函数

函 数 名	说 明	函 数 名	说 明
fplot	绘制平面曲线	ezpolar	绘制极坐标函数图
fplot3	绘制三维曲线	ezplot	绘制平面曲线
fmesh	绘制三维网格图	ezplot3	绘制三维曲线
fsurf	绘制三维面图	ezmesh	绘制三维网格图
fcontour	绘制等高线图	ezsurf	绘制三维面图
fimplicit	绘制平面隐函数图	ezcontour	绘制等高线图
fimplicit3	绘制三维隐函数图	ezcontourf	绘制填充式等高线图
		ezmeshc	绘制带有等高线的三维网格图
		ezsurfz	绘制带有等高线的三维面图

注：以 f 开头的函数是 MATLAB 较新版本里提供的绘图函数，以 ez 开头的函数是 MATLAB 较老版本中提供的绘图函数。

【例 5.1-17】 绘制函数 $f(x) = \frac{1}{\ln|x|}$ 在 $x \in [-6, 6]$ 上的图形。

```
>> syms f(x) % 定义符号函数
>> f(x) = 1/log(abs(x)); % 指定符号函数表达式
>> fplot(f, [-6, 6]); % 绘制函数图形
>> xlabel('x'); % 添加 x 轴标签
>> ylabel(' $ $ f(x) = \frac{1}{\ln|x|} $ $ ', 'Interpreter', 'Latex'); % 添加 y 轴标签
```

以上命令做出的函数图形如图 5.1-2 所示。

【例 5.1-18】 绘制函数 $f(x, y) = \begin{cases} \frac{xy}{x^2 + y^2}, & x^2 + y^2 \neq 0 \\ 0, & x^2 + y^2 = 0 \end{cases}$ 在 $-1 \leq x, y \leq 1$ 上的图形。

```
>> syms f(x, y) % 定义符号函数
>> f(x, y) = x * y / (x^2 + y^2); % 指定符号函数表达式
>> fsurf(f, [-1, 1, -1, 1]) % 绘制函数图形
>> xlabel('x'); ylabel('y'); zlabel('z'); % 添加坐标轴标签
>> hold on % 开启图形保持
>> plot3(0, 0, 0, 'r.', 'MarkerSize', 20) % 绘制原点
>> title(' $ $ f(x, y) = \frac{xy}{x^2 + y^2} $ $ ', 'Interpreter', 'latex') % 添加标题
>> view(127, 34) % 设置视点位置
```

以上命令做出的函数图形如图 5.1-3 所示。

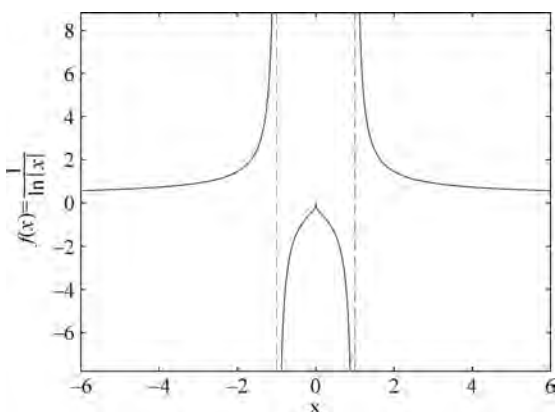


图 5.1-2 一元符号函数图形

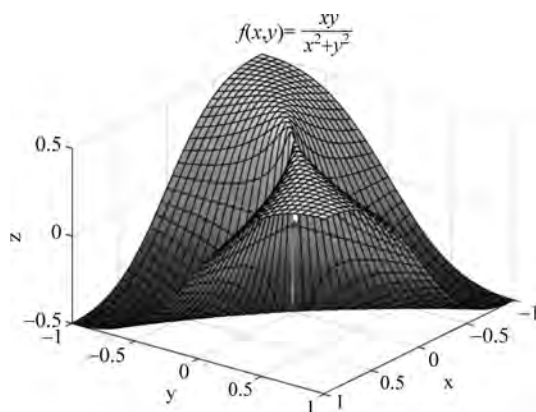


图 5.1-3 二元符号函数图形

5.2 符号微积分

MATLAB 符号计算功能强大,可以解决高等数学中大多数微积分问题,而且求解命令简单,符合人们求解问题的思路。

5.2.1 极限、导数和级数的符号计算

MATLAB 中完成极限、导数和级数符号计算的函数如表 5.2-1 所示。

表 5.2-1 求极限、导数和级数的 MATLAB 函数

函 数 名	使用示例	说 明
limit	limit(f,x,a)	求极限 $\lim_{x \rightarrow a} f(x)$
	limit(f,x,a,'left')	求左极限 $\lim_{x \rightarrow a^-} f(x)$
	limit(f,x,a,'right')	求右极限 $\lim_{x \rightarrow a^+} f(x)$
diff	diff(f,x,n)	求 n 阶导数 $f^{(n)}(x)$
jacobian	jacobian(f,v)	求多元向量函数 $f(v)$ 的 Jacobian 矩阵
taylor	taylor(f, x, x ₀ , 'Order', n)	把 $f(x)$ 在 $x=x_0$ 处作 $n-1$ 阶泰勒展开
symsum	symsum(f,k,a,b)	求 $\sum_{k=a}^b f_k$

【例 5.2-1】求下列极限。

$$(1) \lim_{n \rightarrow \infty} \frac{(-1)^n}{(n+1)^2}; (2) \lim_{x \rightarrow 0^-} \frac{\sin ax}{ax}; (3) \lim_{x \rightarrow \infty} \left(1 - \frac{2}{x}\right)^{kx}; (4) \lim_{y \rightarrow c} \frac{a}{1+x^2+y^2}.$$

```
>> syms n a b c k x y           % 定义符号变量
>> xn = (-1)^n/(n+1)^2;         % 数列通项
>> L1 = limit(xn,n,inf)          % 求数列极限
```

```

L1 =
    0

>> f1 = sin(a * x)/(a * x);           % 定义符号表达式
>> L2 = limit(f1,x,0,'left')          % 求函数极限
L2 =
    1

>> f2 = (1 - 2/x)^(k * x);           % 定义符号表达式
>> L3 = limit(f2,x,inf)              % 求函数极限
L3 =
    exp(- 2 * k)

>> f3 = a/(1 + x^2 + y^2);           % 定义符号表达式
>> L4 = limit(limit(f3,x,b),y,c)      % 求函数极限
L4 =
    a/(b^2 + c^2 + 1)

```

【例 5.2-2】 求显函数和隐函数的导数。

(1) 已知 $f(x) = \sin^2 x$, 求 $f'(1)$ 和 $f''(x)$; (2) 已知 $\cos(x + \sin y) = \sin y$, 求 $\frac{dy}{dx}$ 。

```

>> syms x y                          % 定义符号变量
>> f(x) = sin(x)^2;                  % 定义一元符号函数
>> df = diff(f,x)                    % 求一阶导函数
df(x) =
    2 * cos(x) * sin(x)

>> df_1 = df(1)                      % 求一阶导数值
df_1 =
    2 * cos(1) * sin(1)

>> ddf = diff(f,x,2)                 % 求二阶导函数
ddf(x) =
    2 * cos(x)^2 - 2 * sin(x)^2

>> F(x,y) = cos(x + sin(y)) - sin(y); % 定义二元符号函数
>> dy = -diff(F,x)/diff(F,y)         % 隐函数求导
dy(x, y) =
    -sin(x + sin(y))/(cos(y) + sin(x + sin(y)) * cos(y))

```

【例 5.2-3】 求 $f(x_1, x_2) = \begin{bmatrix} x_1 + x_2 \\ x_2 \ln(x_1) \end{bmatrix}$ 的 Jacobian 矩阵: $\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$ 。

```

>> syms x1 x2                        % 定义符号变量
>> f = [x1 + x2; x2 * log(x1)];      % 定义符号表达式向量
>> v = [x1; x2];                     % 定义自变量向量
>> jac = jacobian(f,v)               % 求 Jacobian 矩阵
jac =
    [ 1, 1]
    [ x2/x1, log(x1)]

```

【例 5.2-4】 求函数 $f(x)=e^x$ 在 $x=0$ 处展开的 5 阶 Maclaurin 公式(函数在 $x=0$ 处的泰勒公式称为函数的 Maclaurin 公式)。

```
>> syms x % 定义符号变量
>> f = exp(x); % 定义符号表达式
>> g = taylor(f, x, 0, 'Order', 6) % 泰勒展开
g =
x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1
```

【例 5.2-5】 求下列无穷级数。

$$(1) \sum_{k=3}^{+\infty} \frac{k-2}{2^k}; (2) \sum_{k=1}^{+\infty} \left[\frac{1}{(2k+1)^2}, \frac{(-1)^k}{3^k} \right]。$$

```
>> syms k % 定义符号变量
>> f1 = (k-2)/2^k; % 级数的一般项
>> s1 = symsum(f1,k,3,inf) % 级数求和
s1 =
1/2

>> f2 = [1/(2*k+1)^2, (-1)^k/3^k]; % 定义向量形式的一般项
>> s2 = symsum(f2,k,1,inf) % 级数求和
s2 =
[ pi^2/8 - 1, -1/4]
```

5.2.2 符号积分计算

MATLAB 中提供了 int 函数,用来求符号函数的积分,其调用格式如下:

```
intf = int(f,x) % 求以 x 为自变量的函数 f 的不定积分
intf = int(f,x,a,b) % 求以 x 为自变量的函数 f 从 a 到 b 的定积分
```

【例 5.2-6】 求下列积分。

$$(1) \int x \ln(ax) dx; (2) \int_{-1}^1 \sqrt{1-x^2} dx; (3) \int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}} dx; (4) \int_1^2 \int_x^{2x} \int_{xy}^{2xy} \frac{(x+y)}{z} dz dy dx。$$

```
>> syms x y z a % 定义符号变量
>> F = int(x*log(a*x),x) % 求不定积分
F =
(x^2*(log(a*x) - 1/2))/2

>> f1 = sqrt(1-x^2); % 定义符号表达式
>> s1 = int(f1,x,-1,1) % 求定积分
s1 =
pi/2

>> f2 = exp(-x^2/2); % 定义符号表达式
```

```

>> s2 = int(f2,x,-inf,inf)           % 求反常积分
s2 =
    2^(1/2) * pi^(1/2)

>> f3 = (x+y)/z;                     % 定义符号表达式
>> s3 = int(int(int(f3,z,x*y,2*x*y),y,x,2*x),x,1,2) % 求多重积分
s3 =
    (35 * log(2))/6

>> s4 = double(s3)                   % 符号数转为双精度数
s4 =
    4.0434

```

5.3 符号方程求解

5.3.1 符号代数方程求解

MATLAB 中提供了 solve 函数,用来求解代数方程的符号解。solve 函数不仅可以求解单个线性/非线性方程,而且还可以求解线性/非线性方程组。它的调用格式如下:

```
[y1, ..., yN] = solve(eqns,vars,Name,Value)
```

【例 5.3-1】 求解下列方程。

(1) $x^3 - 2x^2 + 4x = 8$; (2) $\sin x + \cos 2x = 1$; (3) $x + xe^x - 10 = 0$ 。

```

>> syms x
>> Result1 = solve(x^3 - 2 * x^2 + 4 * x == 8, x) % 求解多项式方程
Result1 =
    2
   -2i
    2i

>> Result2 = solve(sin(x) + cos(2 * x) == 1, x) % 求解三角函数方程
Result2 =
    0
   pi/6
 (5 * pi)/6

% 求解三角函数方程,并返回参数及其条件
>> [Result3,params,conditions] = solve(sin(x) + cos(2 * x) == 1,...
x, 'ReturnConditions',true)

Result3 = % 方程的所有解,其中带有参数 k
    pi * k
   pi/6 + 2 * pi * k
 (5 * pi)/6 + 2 * pi * k

params = % 参数 k
k

```

```

conditions =                                % 参数 k 满足的条件(k 均为整数)
in(k, 'integer')
in(k, 'integer')
in(k, 'integer')

>> Result4 = solve(x + x * exp(x) == 10, x)    % 求解超越方程
警告: Unable to solve symbolically. Returning a numeric solution using vpasolve.

Result4 =                                    % 没有找到符号解, 返回最接近的数值解
1.6335061701558463841931651789789

```

【说明】 对于本例,在求解三角函数方程时,方程的解通常具有周期性,默认情况下, solve 函数只返回一个周期上的解,可将 ReturnConditions 参数值设为 true,使其返回所有解。在求解第三个方程时, solve 函数返回了一个警告信息,这是因为 solve 函数没有找到符号解,此时返回最接近的数值解。

【例 5.3-2】 求解方程组 $\begin{cases} x^{-3} + y^{-3} = 28 \\ x^{-1} + y^{-1} = 4 \end{cases}$ 。

```

>> syms x y                                % 定义符号变量
>> [X,Y] = solve([1/x^3 + 1/y^3 == 28, 1/x + 1/y == 4], [x,y]) % 求解方程组
X =
1
1/3

Y =
1/3
1

```

5.3.2 符号常微分方程求解

MATLAB 中求解符号常微分方程的函数是 dsolve,其调用格式如下:

```
[y1, ..., yN] = dsolve(eqns,conds,Name,Value)
```

【例 5.3-3】 求解常微分方程 $\frac{d^2 y}{dx^2} = x + y$ 。

```

>> syms y(x)                                % 定义符号函数 y(x)
>> Y = dsolve(diff(y,2) == x + y)           % 求解常微分方程
Y =
C2 * exp(x) - x + C1 * exp(-x)

```

上述结果中的 C1、C2 都是任意常数。

【例 5.3-4】 求解初值问题: $\frac{dy}{dt} = 1 + y^2, y(0) = 1$ 。

```

>> syms y(t)                                % 定义符号函数 y(t)
>> Y = dsolve(diff(y) == 1 + y^2, y(0) == 1) % 求解初值问题

```

```

Y =
    tan(t + pi/4)

% 考虑分析上的约束,求解更具一般意义的解
>> Y = dsolve(diff(y) == 1 + y^2, y(0) == 1, ...
    'IgnoreAnalyticConstraints', false)
Y =
    piecewise(in(C1, 'integer'), tan(t + pi/4 + pi*C1))

```

【说明】 本例演示了 `dsolve` 函数的两种调用方法,其中第二种调用用到了 `IgnoreAnalyticConstraints` 参数,其字面意思是“忽略分析上的约束”,这是出于一些求解结果在一般性上的考虑。譬如下面表达式: $\ln e^x$,通常我们认为其等于 x ,这是基于默认 x 为实数情况下得出的结论。如果 $x = 2\pi i$,则 $\ln e^{2\pi i} = \ln 1 = 0$ 而不是等于 $2\pi i$ 。上述 `IgnoreAnalyticConstraints` 有两个参数值可供选择: `true` 和 `false`。默认情况下为 `true`,此时不对所求结果进行一般意义上的推广,所求解在最一般意义条件下可能不成立,但仍满足原始微分方程以及定解条件。而如果选择 `false`,`dsolve` 返回的解(前提是能够求得解析解)在最一般意义下也会成立,但是会增加求不出统一的解析表达式的概率。

【例 5.3-5】 求解两点边值问题: $xy'' - 3y' = x^2$, $y(1) = 0$, $y(5) = 0$, 并绘制解曲线。

```

>> syms y(x)                                % 定义符号函数
% 求解两点边值问题
>> Y = dsolve(x * diff(y,2) - 3 * diff(y) == x^2, [y(1) == 0, y(5) == 0])
Y =
    (31 * x^4)/468 - x^3/3 + 125/468

>> h = fplot(Y, [-1,6]);                    % 绘制解曲线
>> set(h, 'color', 'k', 'LineWidth', 2, 'LineStyle', '--'); % 设置解曲线的属性
>> hold on;                                  % 开启图形保持
>> plot([1 5], [0,0], 'p', 'color', 'r', 'markersize', 12); % 画微分方程的两个边值点
>> text(1,1, 'y(1) = 0');                    % 图上标注边值条件
>> text(4,1, 'y(5) = 0');
>> title('');                                % 设置标题为空
>> hold off;                                  % 关闭图形保存

```

上述代码得到的图形如图 5.3-1 所示。

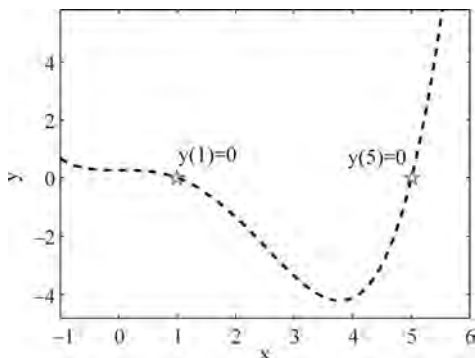


图 5.3-1 两点边值问题的解曲线

【例 5.3-6】 求解两点边值问题： $xy'' - 3y' = x^2, y(1) = 0, y'(5) = 1$ 。

```
>> syms y(x) % 定义符号函数
>> Dy = diff(y); % 求符号函数的一阶导函数
>> eq = x * diff(y,2) - 3 * Dy == x^2; % 定义符号微分方程
>> Y = dsolve(eq, [y(1) == 0, Dy(5) == 1]) % 求解常微分方程
Y =
(13 * x^4)/250 - x^3/3 + 211/750
```

【例 5.3-7】 求解常微分方程组 $\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -x \end{cases}$ 。

```
>> syms x(t) y(t) % 定义符号函数
>> [X, Y] = dsolve(diff(x) == y, diff(y) == -x) % 求解常微分方程组
X =
C1 * cos(t) + C2 * sin(t)

Y =
C2 * cos(t) - C1 * sin(t)
```

上述结果中的 C_1, C_2 都是任意常数。

5.4 建模案例选讲——药物中毒的施救方案

5.4.1 问题描述

一名儿童一次性误服 11 片治疗哮喘病的、剂量为 100mg/片的氨茶碱片,到达医院就诊时,距服药时刻已过去两个小时,此时孩子已出现呕吐、头晕等不良症状。按照药品使用说明书,氨茶碱的成人用量是每次 100~200mg,儿童用量是 3~5mg/kg,如果过量服用,可使血药浓度(单位容积血液中的药量)过高,当血药浓度达到 100 μ g/ml 时,会出现严重中毒,达到 200 μ g/ml 则可致命。接诊医生需要根据专业知识迅速判断孩子的中毒状况,选择合理的施救方案。

5.4.2 问题分析

人体服用药物后的吸收代谢过程如图 5.4-1 所示,药物首先进入胃肠道,再由胃肠道的外壁进入血液循环系统,然后随血液循环被人体脏器代谢而排出体外。血液系统对药物的吸收率(药物从胃肠道向血液系统转移的转移率)通常与胃肠道中的药量成正比,血液系统对药物的排除率(药物从血液系统向体外转移的转移率)通常与血液中的药量成正比。

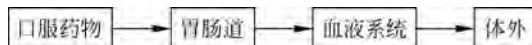


图 5.4-1 口服药物的吸收代谢过程

接诊医生需要根据药物的吸收代谢规律及时计算出孩子入院时的血药浓度,以此选择接下来的施救方案。如果血药浓度达到危险的水平,可选的临床施救方案包括:口服活性炭和体外血液透析,前者通过口服活性炭来吸附药物,可使血液中药物的排除率增至原来(人体自身)的2倍,后者通过在体外对血液进行过滤,可使血液中药物的排除率增至原来的6倍。

5.4.3 模型假设与符号约定

为了判断孩子的中毒状况,需要建立数学模型,求出胃肠道和血液系统中药量随时间变化的规律。这里把 t 时刻胃肠道中的药量记为 $x(t)$,血液系统中的药量记为 $y(t)$,以孩子误服药物的时刻为 $t=0$ 时刻。在此基础上给出如下模型假设。

(1) 由于药物从胃肠道向血液系统转移的转移率与胃肠道中的药量 $x(t)$ 成正比,这里记比例系数为 $\lambda(\lambda>0)$,并假设总剂量为1100mg的药物在 $t=0$ 时刻全部进入胃肠道,即 $x(0)=1100$ 。

(2) 由于血液系统中药物的排除率与血液系统中的药量 $y(t)$ 成正比,这里记比例系数为 $\mu(\mu>0)$,并假设 $t=0$ 时刻血液系统中药量为0,即 $y(0)=0$ 。

(3) 假设任意 t 时刻,血液系统中的药物分布是均匀的。

(4) 在药物总量确定的情况下,血药浓度与血液总量有关。人体血液总量约占体重的7%~8%(L/kg),即每千克体重约有70~80ml血液,一个体重为50~60kg的成年人的血液总量为4000ml左右。假设误服药物儿童的血液总量为成人的一半,约为2000ml。

(5) 比例系数 λ 和 μ 可由氨茶碱吸收和排除的半衰期确定,从药品说明书可知,氨茶碱吸收的半衰期约为5h,即 $x(5)=x(0)/2$,排除的半衰期约为6h,若只考虑血液系统对药物的排除,则有 $y(t+6)=y(t)/2$ 。

5.4.4 模型建立

由假设(1)可知, $x(0)=1100$,随着药物从胃肠道向血液系统的转移,胃肠道中药量 $x(t)$ 会逐渐下降,其下降速度 $\frac{dx}{dt}$ 与 $x(t)$ 成正比(比例系数 $\lambda>0$),从而建立微分方程模型:

$$\frac{dx}{dt} = -\lambda x, x(0) = 1100 \quad (5.4-1)$$

由假设(2)可知, $y(0)=0$,从 $t=0$ 开始,药物从胃肠道向血液系统转移(血液系统吸收药物),同时血液系统启动对药物的排除,血液系统中的药量 $y(t)$ 先增后减。 $y(t)$ 因吸收作用而增长的速度为 λx ,因排除作用而减少的速度与 $y(t)$ 成正比(比例系数 $\mu>0$),所以 $y(t)$ 满足微分方程:

$$\frac{dy}{dt} = \lambda x - \mu y, y(0) = 0 \quad (5.4-2)$$

5.4.5 模型求解

求解式(5.4-1)和式(5.4-2)的MATLAB代码及结果如下:

```

>> syms x(t) y(t) lambda mu % 定义符号函数及参数
>> equ = [diff(x) == -lambda * x, diff(y) == lambda * x - mu * y]; % 定义微分方程
>> cond = [x(0) == 1100, y(0) == 0]; % 定义初值条件
>> [x(t), y(t)] = dsolve(equ, cond) % 求解微分方程
x(t) =
1100e-λt
y(t) =

$$\frac{1100\lambda e^{-\mu t}}{\lambda - \mu} - \frac{1100\lambda e^{-\lambda t}}{\lambda - \mu}$$


```

由以上代码可知：

$$x(t) = 1100e^{-\lambda t} \quad (5.4-3)$$

$$y(t) = \frac{1100\lambda}{\lambda - \mu} (e^{-\mu t} - e^{-\lambda t}) \quad (5.4-4)$$

接下来由氨茶碱吸收和排除的半衰期确定比例系数 λ 和 μ 。由于氨茶碱吸收的半衰期约为5h,可知 $x(5) = x(0)/2$,即 $1100e^{-5\lambda} = 1100/2$,解得 $\lambda = (\ln 2)/5 \approx 0.1386$ 。

为了根据氨茶碱排除的半衰期确定 μ ,不考虑血液系统对药物的吸收,只考虑血液系统对药物的排除,此时 $\frac{dy}{dt} = -\mu y$,假设 τ 时刻血液系统中的药量 $y(\tau) = a$,解得 $y(t) = ae^{-\mu(t-\tau)}$, $t \geq \tau$ 。由于氨茶碱排除的半衰期约为6h,可知 $y(\tau+6) = y(\tau)/2$,即 $ae^{-6\mu} = a/2$,解得 $\mu = (\ln 2)/6 \approx 0.1155$ 。

求解 λ 和 μ 的MATLAB代码如下：

```

>> lambda = solve(x(5) == x(0)/2) % 求解血液对药物的吸收率系数
lambda =

$$\frac{\log(2)}{5}$$

>> syms y2(t) mu tau a % 定义符号函数及参数
>> y2(t) = dsolve(diff(y2) == -mu * y2, y2(tau) == a) % 求解微分方程
y2(t) =
ae-μteμτ
>> mu = solve(y2(tau+6) == y2(tau)/2, mu) % 求解血液对药物的排除率
mu =

$$\frac{\log(2)}{6}$$


```

将 λ 和 μ 的值代入式(5.4-3)和式(5.4-4)可得：

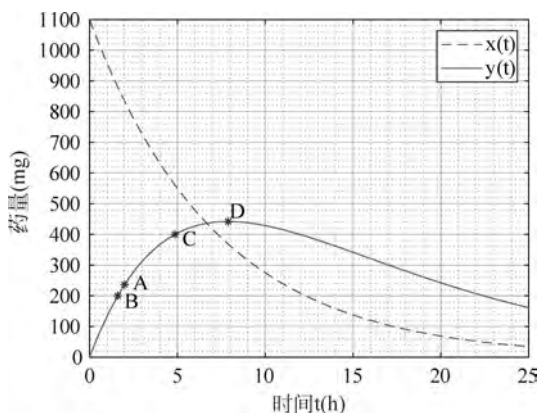
$$x(t) = 1100e^{-(\ln 2)t/5} \quad (5.4-5)$$

$$y(t) = 6600(e^{-(\ln 2)t/6} - e^{-(\ln 2)t/5}) \quad (5.4-6)$$

5.4.6 结果与分析

下面根据模型求解结果绘制 $x(t)$ 和 $y(t)$ 随时间变化的曲线(如图5.4-2所示),并计算孩子就诊时($t=2$)的血药含量,从而判断孩子的中毒状况。

根据模型假设(4)和问题描述,孩子的血液总量为2000ml,当血药浓度达到 $100\mu\text{g/ml}$,即血液中药物含量达到200mg时,会出现严重中毒,当血药浓度达到 $200\mu\text{g/ml}$,即血液中药物

图 5.4-2 胃肠道中的药量 $x(t)$ 和血液系统中的药量 $y(t)$

含量达到 400mg 时可致命。将 $t=2$ 代入式(5.4-6)可得 $y(2)=236.5588\text{mg}$, 由此可知孩子就诊时(图 5.4-2 中 A 点)已经出现严重中毒。结合图 5.4-2 可知, 若不及时进行施救, 血液中药量将在一段时间后超过 400mg, 这将会危及孩子性命。

进一步地, 还可根据模型式(5.4-6)计算血液中药物含量分别达到 200mg 和 400mg 的具体时刻, 以及在不进行施救的情况下血液中药量达到峰值的时刻和相应的峰值。记血液中药量分别达到 200mg、400mg 和峰值的具体时刻分别为 t_B , t_C 和 t_D , 则有

$$y(t_B)=200, y(t_C)=400, y'(t_D)=0 \quad (5.4-7)$$

求解式(5.4-7)并绘图的 MATLAB 代码如下:

```
>> y2h = double(subs(y,2))           % t = 2h 时血液中药量
y2h =
    236.5588
>> tB = double(solve(y(t) == 200))   % 严重中毒时刻
tB =
    1.6090
>> tC = double(solve(y(t) == 400))   % 致命中毒时刻
tC =
    4.8648
>> tD = double(solve(diff(y) == 0))   % 不加施救情形下, 血药量达到峰值时刻
tD =
    7.8910
>> ymax = double(subs(y,tD))         % 最大血药量
ymax =
    442.0653

% 结果可视化
>> figure                             % 创建 figure 窗口
>> fplot(x,[0,25], 'b-- ')           % 胃肠道中药量变化曲线
>> hold on                             % 开启图形保持
>> fplot(y,[0,25], 'r')               % 血液中药量变化曲线
>> grid on                             % 添加主网格线
>> set(gca, 'XMinorGrid', 'on', 'YMinorGrid', 'on') % 添加次网格线
>> xlabel('时间 t(h)'); ylabel('药量(mg)'); % 添加坐标轴标签
>> plot([2,tB,tC,tD],[y2h,200,400,ymax], 'b* ') % 绘制各时刻标记点
>> text([2,tB,tC,tD]+0.5,[y2h,200,400,ymax],... % 添加标记点的文本标注
        {'A','B','C','D'})
>> legend('x(t)', 'y(t)')             % 添加图例
```

由以上求解结果可知血液中药量 $y(t)$ 在 $t_B = 1.6090\text{h}$ 达到 200mg , 在 $t_C = 4.8648\text{h}$ (到达医院后 2.8648h) 达到 400mg , 在 $t_D = 7.8910\text{h}$ (到达医院后 5.8910h) 达到峰值 442.0653mg 。以上各状态分别对应图 5.4-2 中 B、C、D 点。

5.4.7 施救方案

模型计算结果表明,若不及时进行施救,孩子会有生命危险,因此需要选择合适的施救方案。下面针对口服活性炭和体外血液透析两种施救方案,重新计算血液中药量的变化规律。

1. 口服活性炭

假设孩子到达医院时($t=2$)即开始施救,施救中($t \geq 2$)血液里的药量记为 $z(t)$ 。由于口服活性炭可使血液中药物的排除率增至原来(人体自身)的 2 倍,则血液中药物的排除率与血液中药量的比例系数将变为原来的 2 倍,即 $\mu = (\ln 2)/3$,新的模型为

$$\begin{cases} \frac{dx}{dt} = -\lambda x, & x(0) = 1100 \\ \frac{dz}{dt} = \lambda x - \mu z, & t \geq 2, z(2) = 236.5588 \\ \lambda = \frac{\ln 2}{5}, & \mu = \frac{\ln 2}{3} \end{cases} \quad (5.4-8)$$

求解模型式(5.4-8)的 MATLAB 代码如下:

```
% ----- 施救方案:口服活性炭 -----
>> syms x(t) z(t) % 定义符号函数
>> lambda = log(2)/5; % 定义 lambda
>> mu = 2 * log(2)/6; % 定义 mu
>> eq2 = [diff(x) == -lambda * x, diff(z) == lambda * x - mu * z]; % 定义微分方程
>> cond2 = [x(0) == 1100, z(2) == 236.5588]; % 定义初值条件
>> [x(t), z(t)] = dsolve(eq2, cond2); % 求解微分方程
>> z = vpa(z, 5) % 以 5 位有效数字形式显示
z(t) =
    1650.0 * exp(-0.13863 * t) - 1609.5 * exp(-0.23105 * t)

% 结果可视化
>> figure % 创建 figure 窗口
>> fplot(x, [0, 25], 'b--') % 胃肠道中药量变化曲线
>> hold on % 开启图形保持
>> fplot(y, [0, 25], 'r-') % 血液中药量变化曲线(不施救)
>> fplot(z, [2, 25], 'k') % 血液中药量变化曲线(施救)
>> grid on % 添加主网格线
>> set(gca, 'XMinorGrid', 'on', 'YMinorGrid', 'on') % 添加次网格线
>> xlabel('时间 t(h)'); ylabel('药量(mg)'); % 添加坐标轴标签
>> legend('x(t)', 'y(t)', 'z(t)') % 添加图例

>> t4 = double(solve(diff(z) == 0)) % 施救情形下,血药量达到峰值时刻
t4 =
    5.2582
>> zmax = double(subs(z, t4)) % 最大血药量
zmax =
    318.3973
```

由以上代码可知：

$$z(t) = 1650e^{-0.13863t} - 1609.5e^{-0.23105t}, \quad t \geq 2 \quad (5.4-9)$$

$x(t), y(t), z(t)$ 随时间变化的曲线如图 5.4-3 所示。由 $z'(t) = 0$ 解得 $t = 5.2582$ ，相应的 $z(t) = 318.3973$ 。很显然，经口服活性炭施救后，血液中药量 $z(t)$ 在 $t = 5.2582\text{h}$ 时达到峰值 318.3973mg ，远低于 $y(t)$ 的峰值 (442.0653mg) 和致命水平 (400mg)。

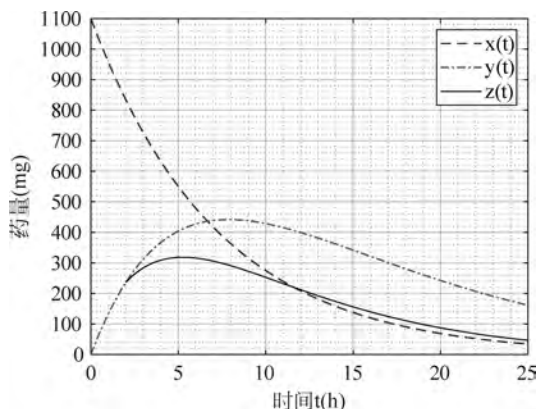


图 5.4-3 经口服活性炭施救后血液系统中的药量 $z(t)$

2. 体外血液透析

由于体外血液透析可使血液中药物的排除率增至原来(人体自身)的 6 倍，则血液中药物的排除率与血液中药量的比例系数将变为原来的 6 倍，即 $\mu = \ln 2$ ，新的模型为：

$$\begin{cases} \frac{dx}{dt} = -\lambda x, & x(0) = 1100 \\ \frac{dz}{dt} = \lambda x - \mu z, & t \geq 2, z(2) = 236.5588 \\ \lambda = \frac{\ln 2}{5}, & \mu = \ln 2 \end{cases} \quad (5.4-10)$$

求解模型式(5.4-10)的 MATLAB 代码如下：

```
% ----- 施救方案:体外血液透析 -----
>> syms x(t) z(t) % 定义符号函数
>> lambda = log(2)/5; % 定义 lambda
>> mu = 6 * log(2)/6; % 定义 mu
>> eq2 = [diff(x) == -lambda * x, diff(z) == lambda * x - mu * z]; % 定义微分方程
>> cond2 = [x(0) == 1100, z(2) == 236.5588]; % 定义初值条件
>> [x(t), z(t)] = dsolve(eq2, cond2); % 求解微分方程
>> z = vpa(z, 5) % 以 5 位有效数字形式显示
z(t) =
    275.0 * exp(-0.13863 * t) + 112.59 * exp(-0.69315 * t)

% 结果可视化
>> figure % 创建 figure 窗口
>> fplot(x, [0, 25], 'b--') % 胃肠道中药量变化曲线
>> hold on % 开启图形保持
>> fplot(y, [0, 25], 'r-.') % 血液中药量变化曲线(不施救)
```

```

>> fplot(z,[2,25], 'k')           % 血液中药量变化曲线(施救)
>> grid on                        % 添加主网格线
>> set(gca, 'XMinorGrid', 'on', 'YMinorGrid', 'on') % 添加次网格线
>> xlabel('时间 t(h)'); ylabel('药量(mg)'); % 添加坐标轴标签
>> legend('x(t)', 'y(t)', 'z(t)') % 添加图例

```

由以上代码可知：

$$z(t) = 275e^{-0.13863t} - 112.59e^{-0.69315t}, \quad t \geq 2 \quad (5.4-11)$$

$x(t)$, $y(t)$, $z(t)$ 随时间变化的曲线如图 5.4-4 所示。很显然, 经体外血液透析施救后, 血液中药量 $z(t)$ 关于时间 t 单调递减, 说明体外血液透析的救治方案是十分有效的。

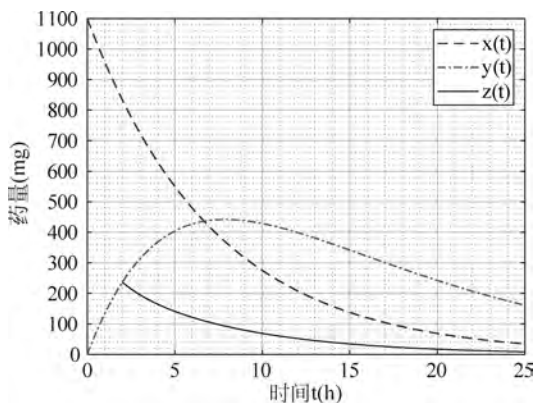


图 5.4-4 经体外血液透析施救后血液系统中的药量 $z(t)$

由以上计算可知两种施救方案均有效地将血液中药量降至致命水平以下, 虽然体外血液透析可使血液中药量下降更快, 但其治疗成本较高, 并且存在安全风险, 综合考虑, 应该选择口服活性炭方案。