云南大学数学与统计学院实验教学中心

实验报告

|  |  |  |
| --- | --- | --- |
| **课程名称**：数学建模实验 | **学期：**2016~2017学年下学期 | |
| **指导教师**：李朝迁 | | |
| **学生：**刘鹏 20151910042 信计 | **学生：**王泽坤 20151910011 应数 | **学生：**段奕臣 20151910002 应数 |
| **实验名称**：**无约束优化** & MATLAB实现 | | **成绩**： |
| **实验编号**：NO.6 | **实验日期**：2017年6月26日 | **实验学时**：2 |
| **学院：**数学与统计学院 | **专业：**信息与计算科学 | **年级**：2015级 |

# 一、实验目的

1. 学习优化的基本步骤；

2. 初步学会MATLAB的优化操作。

# 二、实验内容

1. 完成课后布置的习题；

2. 调试书上的经典代码，以掌握MATLAB命令。

# 三、实验平台

Windows10 Enterprise 1703中文版操作系统；

MATLAB R2017a 中文版。

# 四、实验记录与实验结果分析

1题

利用help或者document学习fminbnd，fminunc，fminsearch命令。

**Solution**:

（1）

**fminbnd**

Find minimum of single-variable function on fixed interval

＊fminbnd函数能找出单变量函数在给定区间上的最小值。

fminbnd is a one-dimensional minimizer that finds a minimum for a problem specified by

＊fminbnd是一个一维的分析函数，能够找出如下形式的函数的最小值



*x*, *x*1, and *x*2 are finite scalars, and *f*(*x*) is a function that returns a scalar.

＊*x*, *x*1 *,x*2都是有限量，而*f* (*x*)是一个返回标量的函数。

**Syntax**

x = fminbnd (fun, x1, x2)

x = fminbnd (fun, x1, x2, options)

x = fminbnd (problem)

[x, fval] = fminbnd (**\_\_\_**)

[x, fval, exitflag] = fminbnd (**\_\_\_**)

[x, fval, exitflag, output] = fminbnd (**\_\_\_**)

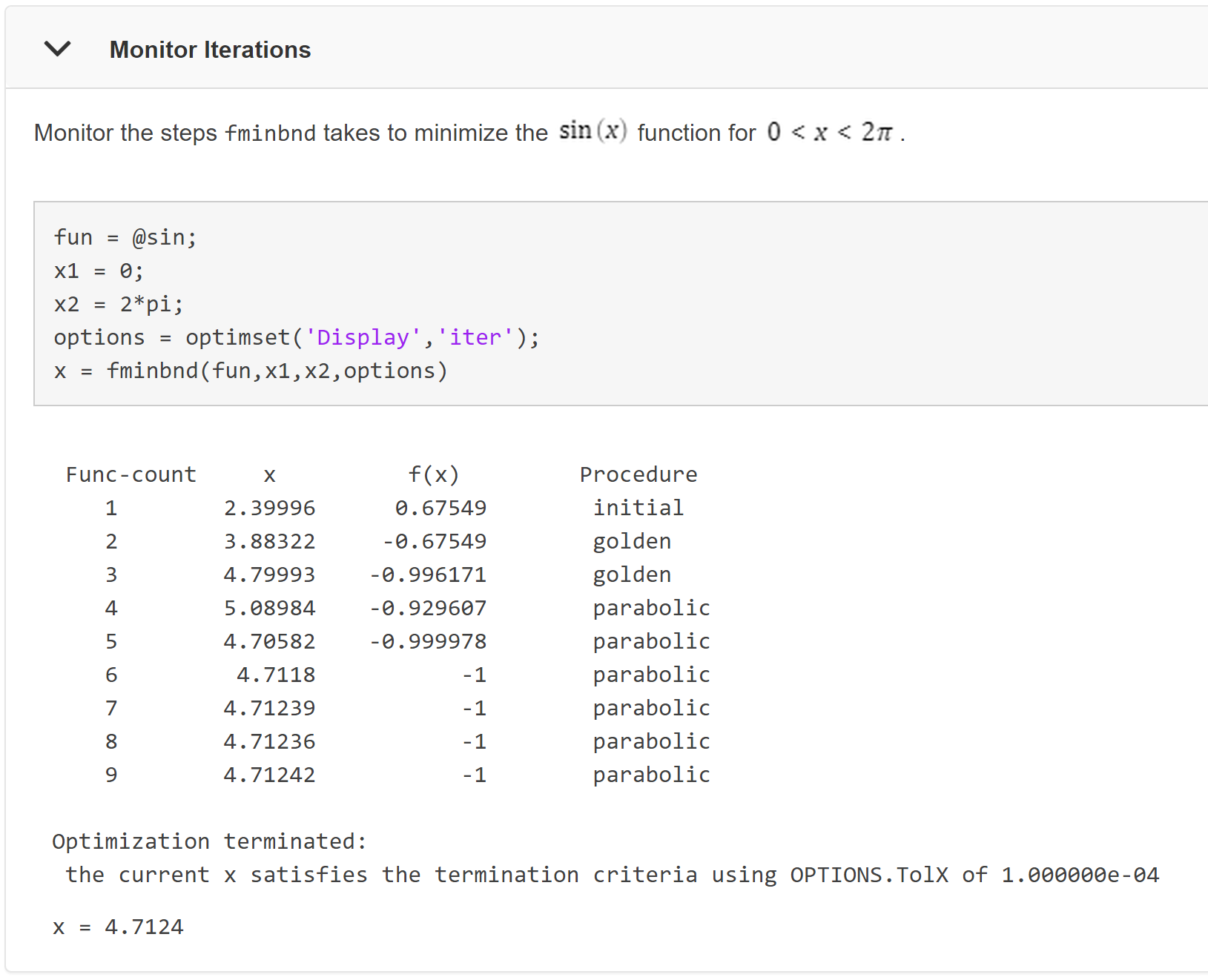
**Description**

x = fminbnd (fun, x1, x2) returns a value x that is a local minimizer of the scalar valued function that is described in fun in the interval x1 < x < x2.

＊x = fminbnd (fun, x1, x2)语句返回一个标量数值*x*，它是函数在x1，x2之间的最小值点。

x = fminbnd (fun, x1, x2, options) minimizes with the optimization options specified in options. Use optimset to set these options.

＊带参数的fminbnd，可以通过optimset设置参数。



x = fminbnd (problem) finds the minimum for problem, where problem is a structure.

＊problem是一个结构，此语句返回problem的最小值。

[x, fval] = fminbnd (**\_\_\_**), for any input arguments, returns the value of the objective function computed in fun at the solution x.

＊通过给定的参数，此语句返回最小值点*x*与其对应的最小值*fval*。

[x, fval, exitflag] = fminbnd (**\_\_\_**) additionally returns a value exitflag that describes the exit condition.

＊在上一个函数的基础上，增加了退出条件。

[x, fval, exitflag, output] = fminbnd (**\_\_\_**) additionally returns a structure output that contains information about the optimization.

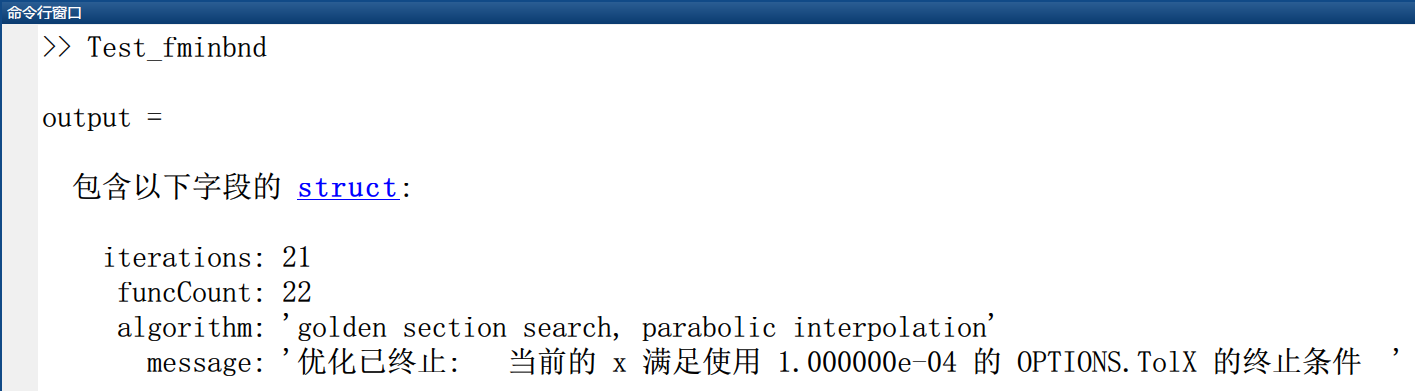
＊返回值中增加了一个output结构，它存数了参数的信息。

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | % example 1  x **=** fminbnd**(@**square**,**1**,**2**);**  % example 2  **[**a**,**b**]** **=** fminbnd**(@**square**,**0.5**,**2**);**  % example 3  **[**a**,**b**,**flag**]** **=** fminbnd**(@**square**,**0.3**,**2**);**  % example 4  **[**a**,**b**,**flag**,**output**]** **=** fminbnd**(@**square**,**0.3**,**2**);**  output  **function** y **=** square**(**x**)**  y **=** x**^**2**;**  **end** |

程序代码 1

运行结果：



运行结果 1

代码分析：

通过调试fminbnd的几种可能用法，查看如何使用这个函数。

（2）

**fminunc**

Find minimum of unconstrained multivariable function

＊这个函数会返回无约束的多变量函数的最小值。

Nonlinear programming solver.

＊非线性规划的解决器

Finds the minimum of a problem specified by

＊这个命令会确定如下形式的函数的最小值



where *f*(*x*) is a function that returns a scalar.

＊然后会返回一个标量。

*x* is a vector or a matrix; see Matrix Arguments.

＊x可以是一个向量，也可以是一个矩阵；参见矩阵参数。

**Syntax**

x = fminunc (fun, x0)

x = fminunc (fun, x0, options)

x = fminunc(problem)

[x, fval] = fminunc (**\_\_\_**)

[x, fval, exitflag, output] = fminunc (**\_\_\_**)

[x, fval, exitflag, output, grad, hessian] = fminunc (**\_\_\_**)

**Description**

x = fminunc (fun, x0) starts at the point x0 and attempts to find a local minimum x of the function described in fun. The point x0 can be a scalar, vector, or matrix.

＊以x0为初始值，尝试性地去找附近的最小值点x，x0可以是标量向量或者矩阵。

**Note:** Passing Extra Parameters explains how to pass extra parameters to the objective function and nonlinear constraint functions, if necessary.

＊注意：额外参数传递解释了如何把额外的参数传递到对象函数以及非线性约束函数。

fminunc is for nonlinear problems without constraints. If your problem has constraints, generally use fmincon. See Optimization Decision Table.

＊fminunc函数是为了解决非线性无约束问题的，如果你想要解决有约束的问题，那就需要fmincon函数了。

x = fminunc (fun, x0, options) minimizes fun with the optimization options specified in options. Use optimoptions to set these options.

＊带参数。

x = fminunc(problem) finds the minimum for problem, where problem is a structure described in Input Arguments. Create the problem structure by exporting a problem from Optimization app, as described in Exporting Your Work.

＊problem是一个结构体，详见input arguments。可以通过最优化工具箱来创建一个

[x, fval] = fminunc (**\_\_\_**), for any syntax, returns the value of the objective function fun at the solution x.

＊返回的不仅仅是最优解x，而且还有x处的目标函数值。

[x, fval, exitflag, output] = fminunc (**\_\_\_**) additionally returns a value exitflag that describes the exit condition of fminunc, and a structure output with information about the optimization process.

＊在上面的基础上，又增加了退出条件 – exitflag、以及关于优化进行的信息的一个结构体output。

[x, fval, exitflag, output, grad, hessian] = fminunc (**\_\_\_**) additionally returns:

＊在上面的基础上，又增加了以下两个输出：

* grad — Gradient of fun at the solution x.

＊在解x处的梯度grad。

* hessian — Hessian of fun at the solution x. See fminunc Hessian.

＊在x处的黑森矩阵。详见fminunc hessian。

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **function** **[**f**,**g**]** **=** rosenbrockwithgrad**(**x**)**  % Calculate objective f  f **=** 100**\*(**x**(**2**)** **-** x**(**1**)^**2**)^**2 **+** **(**1**-**x**(**1**))^**2**;**  **if** nargout **>** 1 % gradient required  g **=** **[-**400**\*(**x**(**2**)-**x**(**1**)^**2**)\***x**(**1**)-**2**\*(**1**-**x**(**1**));**  200**\*(**x**(**2**)-**x**(**1**)^**2**)];**  **end** |

程序代码 2

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | % filename: Test\_fminunc  options **=** optimoptions**(**'fminunc'**,**'Algorithm'**,**...  'trust-region'**,**'SpecifyObjectiveGradient'**,**true**);**  problem**.**options **=** options**;**  problem**.**x0 **=** **[-**1**,**2**];**  problem**.**objective **=** **@**rosenbrockwithgrad**;**  problem**.**solver **=** 'fminunc'**;**  **[**x**,** fval**]** **=** fminunc**(**problem**)** |

程序代码 3

运行结果：



运行结果 2

代码分析：

（3）

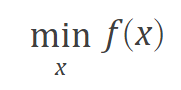
**fminsearch**

Find minimum of unconstrained multivariable function using derivative-free method

＊使用单纯形方法找无约束多变量函数的最小值。

Nonlinear programming solver that searches for the minimum of a problem specified by

＊这个函数时非线性规划问题的解决器，可以搜索问题的最小值，问题由以下式子界定：



*f*(*x*) is a function that returns a scalar, and *x* is a vector or a matrix.

＊其中f(x)是一个返回标量的函数，而x可以是向量，也可以是矩阵。

**Syntax**

x = fminsearch (fun, x0)

x = fminsearch (fun, x0, options)

x = fminsearch(problem)

[x, fval] = fminsearch (**\_\_\_**)

[x, fval, exitflag] = fminsearch (**\_\_\_**)

[x, fval, exitflag, output] = fminsearch (**\_\_\_**)

**Description**

x = fminsearch (fun, x0) starts at the point x0 and attempts to find a local minimum x of the function described in fun.

＊x = fminsearch(fun, x0)从初值x0开始，尝试找函数fun的极值。

x = fminsearch (fun, x0, options) minimizes with the optimization options specified in the structure options. Use optimset to set these options.

＊x = fminsearch(fun, x0, options)，指定了优化选项。使用optimset设置这些选项。

x = fminsearch(problem) finds the minimum for problem, where problem is a structure.

＊x = fminsearch(problem)，其中problem是一个结构，这个结构中把需要的信息都以成员的形式写出了。

[x, fval] = fminsearch (**\_\_\_**), for any previous input syntax, returns in fval the value of the objective function fun at the solution x.

＊[x, fval] = fminsearch(\_\_\_)，对于任何上述的输入语法，都会x处的函数值。

[x, fval, exitflag] = fminsearch (**\_\_\_**) additionally returns a value exitflag that describes the exit condition.

＊[x, fval, exitflag] = fminsearch(\_\_\_)另外返回一个值exitflag，该值描述退出条件。

[x, fval, exitflag, output] = fminse­arch (**\_\_\_**) additionally returns a structure output with information about the optimization process.

＊[x, fval, exitflag, output] = fminsearch(\_\_\_)，额外地返回一个包含优化过程信息的结构输出。

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | % filename: Test\_fminsearch  fun **=** **@(**x**)**100**\*(**x**(**2**)** **-** x**(**1**)^**2**)^**2 **+** **(**1 **-** x**(**1**))^**2**;**  options **=** optimset**(**'PlotFcns'**,@**optimplotfval**);**  x0 **=** **[-**1.2**,**1**];**  x **=** fminsearch**(**fun**,**x0**,**options**)** |

程序代码 4

运行结果：



代码分析：

options的所有写入内容都是建立在优化工具箱的基础之上的。

2题

利用help或者document学习lsqnonlin，lsqcurvefit命令。

**Solution**:

（1）

**lsqnonlin**

Solve nonlinear least-squares (nonlinear data-fitting) problems

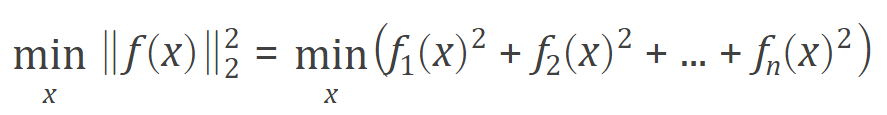
＊解决非线性最小二乘（非线性数据拟合）问题。

Nonlinear least-squares solver

＊非线性最小二乘解决器。

Solves nonlinear least-squares curve fitting problems of the form

＊解决如下形式的非线性最小二乘曲线拟合问题：



with optional lower and upper bounds lb and ub on the components of x.

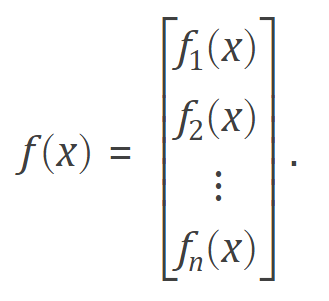
＊可以选择参数lb和ub，它们分别是x的上下界。

x, lb, and ub can be vectors or matrices; see Matrix Arguments.

＊x，lb，ub可以是向量，也可以是矩阵，详见矩阵参数。

Rather than compute the value (the sum of squares), lsqnonlin requires the user-defined function to compute the vector-valued function

＊除了进行最小二乘计算之外，这个函数还可以计算用户自定义的函数以及向量值函数。



**Syntax**

x = lsqnonlin (fun, x0)

x = lsqnonlin (fun, x0, lb, ub)

x = lsqnonlin (fun, x0, lb, ub, options)

x = lsqnonlin (problem)

[x, resnorm] = lsqnonlin (**\_\_\_**)

[x, resnorm, residual, exitflag, output] = lsqnonlin (**\_\_\_**)

[x, resnorm, residual, exitflag, output, lambda, jacobian] = lsqnonlin (**\_\_\_**)

**Description**

x = lsqnonlin (fun, x0) starts at the point x0 and finds a minimum of the sum of squares of the functions described in fun. The function fun should return a vector of values and not the sum of squares of the values. (The algorithm implicitly computes the sum of squares of the components of fun(x).)

＊从x0处开始，找到由fun描述的最小二乘的值。函数fun返回的是一个向量，而不是平方的和。

Note: Passing Extra Parameters explains how to pass extra parameters to the vector function fun(x), if necessary.

＊注意：额外参数传递规则解释了在必要情况下，如何将额外的参数传递给向量值函数fun(x)。

x = lsqnonlin (fun, x0, lb, ub) defines a set of lower and upper bounds on the design variables in x, so that the solution is always in the range lb ≤ x ≤ ub. You can fix the solution component x(i) by specifying lb(i) = ub(i).

＊x的上下界被lb与ub约束，所以输出结果就在区间[ lb, ub ]之内。可以通过指定lb(i)= ub(i)来确定x(i)的取值。

Note: If the specified input bounds for a problem are inconsistent, the output x is x0 and the outputs resnorm and residual are [].

＊注意：如果上下界前后矛盾，那么输出就是初值x0，而其他的也都是空的。

Components of x0 that violate the bounds lb ≤ x ≤ ub are reset to the interior of the box defined by the bounds. Components that respect the bounds are not changed.

＊超过限制的x0，将会被抹掉，而合理的就不会变。

x = lsqnonlin (fun, x0, lb, ub, options) minimizes with the optimization options specified in options. Use optimoptions to set these options. Pass empty matrices for lb and ub if no bounds exist.

＊带了参数。

x = lsqnonlin(problem) finds the minimum for problem, where problem is a structure described in Input Arguments. Create the problem structure by exporting a problem from Optimization app, as described in Exporting Your Work.

＊结构体参数。

[x, resnorm] = lsqnonlin (**\_\_\_**), for any input arguments, returns the value of the squared 2-norm of the residual at x : sum (fun(x).^2).

＊额外输出x平方的和。

[x, resnorm, residual, exitflag, output] = lsqnonlin (**\_\_\_**) additionally returns the value of the residual fun(x) at the solution x, a value exitflag that describes the exit condition, and a structure output that contains information about the optimization process.

＊额外返回x处的残差值，描述退出条件的值exitflag，以及包含有关优化过程的结构体output。

[x, resnorm, residual, exitflag, output, lambda, jacobian] = lsqnonlin (**\_\_\_**) additionally returns a structure lambda whose fields contain the Lagrange multipliers at the solution x, and the Jacobian of fun at the solution x.

＊额外另外返回一个结构lambda，它包含解x的拉格朗日乘数，返回函数在解处的雅可比矩阵。

（2）

**lsqcurvefit**

Solve nonlinear curve-fitting (data-fitting) problems in least-squares sense

＊利用最小二乘，解决非线性曲线拟合问题。

Nonlinear least-squares solver

＊非线性最小二乘解决器。

Find coefficients x that solve the problem

＊找到能解决这个问题的最小系数x



given input data xdata, and the observed output ydata, where xdata and ydata are matrices or vectors, and F(x, xdata) is a matrix-valued or vector-valued function of the same size as ydata.

Optionally, the components of x can have lower and upper bounds lb, and ub. The arguments x, lb, and ub can be vectors or matrices; see Matrix Arguments.

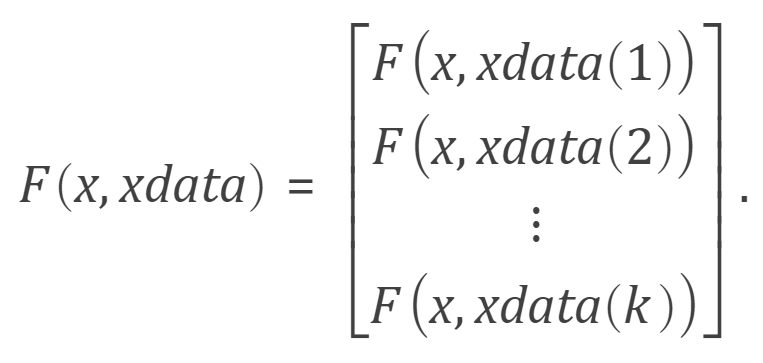
＊xdata和ydata可以是矩阵或者向量，F(x, xdata)是矩阵值或者向量值的函数，与ydata同型。x也可以有上下界lb和ub，x，lb，ub可以是矩阵或者向量。

The lsqcurvefit function uses the same algorithm as lsqnonlin. lsqcurvefit simply provides a convenient interface for data-fitting problems.

＊lsqnonlin与lsqcurvefit的算法是一致的，后者为数据拟合问题提供了好的用户接口。

Rather than compute the sum of squares, lsqcurvefit requires the user-defined function to compute the vector-valued function

＊除了计算平方和之外，lsqcurvefit要求用户定义的函数是向量值函数。

****

**Syntax**

x = lsqcurvefit (fun, x0, xdata, ydata)

x = lsqcurvefit (fun, x0, xdata, ydata, lb, ub)

x = lsqcurvefit (fun, x0, xdata, ydata, lb, ub, options)

x = lsqcurvefit (problem)

[x, resnorm] = lsqcurvefit (**\_\_\_**)

[x, resnorm, residual, exitflag, output] = lsqcurvefit (\_\_\_)

[x, resnorm, residual, exitflag, output, lambda, jacobian] = lsqcurvefit (**\_\_\_**)

**Description**

x = lsqcurvefit (fun, x0, xdata, ydata) starts at x0 and finds coefficients x to best fit the nonlinear function fun (x, xdata) to the data ydata (in the least-squares sense). ydata must be the same size as the vector (or matrix) F returned by fun.

＊从x0开始，找到能最好地拟合非线性函数fun(x, xdata)系数x，ydata必须与fun的返回值同型。

Note: Passing Extra Parameters explains how to pass extra parameters to the vector function fun(x), if necessary.

＊＊注意：额外参数传递规则解释了在必要情况下，如何将额外的参数传递给向量值函数fun(x)。

x = lsqcurvefit (fun, x0, xdata, ydata, lb, ub) defines a set of lower and upper bounds on the design variables in x, so that the solution is always in the range lb ≤ x ≤ ub. You can fix the solution component x(i) by specifying lb(i) = ub(i).

＊加入了参数x的上下限。

Note: If the specified input bounds for a problem are inconsistent, the output x is x0 and the outputs resnorm and residual are [].

＊＊注意：如果上下界前后矛盾，那么输出就是初值x0，而其他的也都是空的。

Components of x0 that violate the bounds lb ≤ x ≤ ub are reset to the interior of the box defined by the bounds. Components that respect the bounds are not changed.

＊超过限制的x0，将会被抹掉，而合理的就不会变。

x = lsqcurvefit (fun, x0, xdata, ydata, lb, ub, options) minimizes with the optimization options specified in options. Use optimoptions to set these options. Pass empty matrices for lb and ub if no bounds exist.

＊加入了options选项。如果lb与ub都不存在，那么需要补上两个空矩阵。

x = lsqcurvefit(problem) finds the minimum for problem, where problem is a structure described in Input Arguments. Create the problem structure by exporting a problem from Optimization app, as described in Exporting Your Work.

＊结构体。

[x, resnorm] = lsqcurvefit(**\_\_\_**), for any input arguments, returns the value of the squared 2-norm of the residual at x: sum((fun(x, xdata)-ydata).^2).

＊多了残差输出。

[x, resnorm, residual, exitflag, output] = lsqcurvefit (**\_\_\_**) additionally returns the value of the residual fun(x, xdata)-ydata at the solution x, a value exitflag that describes the exit condition, and a structure output that contains information about the optimization process.

＊额外返回x处的残差值，描述退出条件的值exitflag，以及包含有关优化过程的结构体output。

[x, resnorm, residual, exitflag, output, lambda, jacobian] = lsqcurvefit(**\_\_\_**) additionally returns a structure lambda whose fields contain the Lagrange multipliers at the solution x, and the Jacobian of fun at the solution x.

＊额外另外返回一个结构lambda，它包含解x的拉格朗日乘数，返回函数在解处的雅可比矩阵。

3题

对Page 151例1、2进行验证。

**Solution**:

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | % filename: Data\_fit  clc  cd **..;**cd **..;**cd **./**#Data**/**06  i **=** xlsread**(**'experiment\_6.xlsx'**,**'A2:A34'**);**  y **=** xlsread**(**'experiment\_6.xlsx'**,**'B2:B34'**);**  fun **=** **@(**x**,**t**)** x**(**1**)** **+** x**(**2**)** **\*** exp**(-**x**(**4**)** **\*** t**)** **+** x**(**3**)** **\*** exp**(-**x**(**5**)** **\*** t**);**  x0 **=** **[**0.5**,**1.5**,-**1**,**0.01**,**0.02**];**  opt **=** optimset**(**'Display'**,**'iter'**);**  x **=** lsqcurvefit**(**fun**,**x0**,**i**,**y**,[],[],**opt**)**  y **=** y **+** rand**()/**2**;**  x **=** lsqcurvefit**(**fun**,**x0**,**i**,**y**,[],[],**opt**)** |

程序代码 5

运行结果：

|  |
| --- |
| Norm of First-order  Iteration Func-count f(x) step optimality  0 6 7.03048 346  1 12 7.03048 10 346  2 18 0.972321 2.5 58.9  3 24 0.972321 5 58.9  4 30 0.0639713 1.25 1.83  5 36 0.0639713 2.5 1.83  6 42 0.0639713 0.625 1.83  7 48 0.0634801 0.15625 0.279  8 54 0.0629471 0.3125 1.47  9 60 0.0629471 0.625 1.47  10 66 0.0627967 0.15625 0.546  11 72 0.0627825 0.3125 0.618  12 78 0.0627632 0.3125 0.572  13 84 0.0627445 0.3125 0.489  14 90 0.0627253 0.3125 0.373  15 96 0.0627253 0.625 0.373  16 102 0.0627003 0.15625 0.66  17 108 0.0627003 0.3125 0.66  18 114 0.0626835 0.078125 0.44  19 120 0.0626667 0.15625 0.482  20 126 0.0626667 0.3125 0.482  21 132 0.0626543 0.078125 0.539  22 138 0.06264 0.15625 0.361  23 144 0.06264 0.3125 0.361  24 150 0.0626255 0.078125 0.723  25 156 0.0626083 0.15625 0.27  26 162 0.0626052 0.3125 1.6  27 168 0.0625375 0.078125 0.242  28 174 0.0624883 0.15625 1.7  29 180 0.0624204 0.3125 0.389  30 186 0.0624189 0.625 0.941  31 192 0.0624046 0.15625 0.251  32 198 0.0623886 0.3125 1.85  33 204 0.0623391 0.3125 0.336  34 210 0.0623253 0.625 0.465  35 216 0.0623253 1.25 0.465  36 222 0.0623201 0.3125 0.323  37 228 0.0623201 0.625 0.323  38 234 0.0623142 0.15625 0.934  39 240 0.0623062 0.3125 0.277  40 246 0.0622759 0.625 0.829  41 252 0.0622759 1.25 0.829  42 258 0.0622721 0.3125 0.571  43 264 0.0622708 0.625 0.691  44 270 0.0622692 0.625 0.698  45 276 0.0622675 0.625 0.659  46 282 0.062266 0.625 0.609  47 288 0.0622645 0.625 0.554  48 294 0.0622645 1.25 0.554  49 300 0.0622627 0.3125 0.32  50 306 0.0622627 0.625 0.32  51 312 0.0622601 0.15625 0.944  52 318 0.0622569 0.3125 0.205  53 324 0.0588124 0.625 24.7  54 330 0.0570094 1.25 0.813  55 336 0.0570094 2.5 0.813  56 342 0.0570088 0.625 0.757  57 348 0.0570083 0.625 0.721  58 354 0.0570078 0.625 0.689  59 360 0.0570074 0.625 0.658  60 366 0.057007 0.625 0.628  61 372 0.0570066 0.625 0.598  62 378 0.0570062 0.625 0.568  63 384 0.0570058 0.625 0.537  64 390 0.0570055 0.625 0.518  65 396 0.0570051 0.625 0.523  66 402 0.0570048 0.625 0.529  Local minimum possible.  lsqcurvefit stopped because the final change in the sum of squares relative to  its initial value is less than the default value of the function tolerance.  <stopping criteria details>  x =  0.2217 -13.3675 14.1518 0.0270 0.0278  Norm of First-order  Iteration Func-count f(x) step optimality  0 6 5.42566 304  1 12 5.42566 10 304  2 18 0.384073 2.5 32.6  3 24 0.384073 5 32.6  4 30 0.384073 1.25 32.6  5 36 0.0959295 0.3125 0.6  6 42 0.0959295 0.625 0.6  7 48 0.09525 0.15625 0.476  8 54 0.09525 0.3125 0.476  9 60 0.0946324 0.078125 1.96  10 66 0.09369 0.15625 0.456  11 72 0.0550464 0.3125 6.44  12 78 0.0550464 0.625 6.44  13 84 0.0515626 0.15625 2.05  14 90 0.0515626 0.3125 2.05  15 96 0.0512174 0.078125 0.204  16 102 0.0512174 0.15625 0.204  17 108 0.0512123 0.0390625 0.0827  18 114 0.0512123 0.078125 0.0827  19 120 0.0511951 0.0195312 0.446  20 126 0.0511761 0.0390625 0.098  21 132 0.0511289 0.078125 0.57  22 138 0.0511289 0.15625 0.57  23 144 0.0511109 0.0390625 0.14  24 150 0.0511075 0.078125 0.115  25 156 0.0511075 0.15625 0.115  26 162 0.0511019 0.0390625 0.246  27 168 0.0511019 0.078125 0.246  28 174 0.0510983 0.0195312 0.131  29 180 0.0510948 0.0390625 0.168  30 186 0.0510948 0.078125 0.168  31 192 0.0510924 0.0195312 0.157  32 198 0.0510897 0.0390625 0.115  33 204 0.0510897 0.078125 0.115  34 210 0.0510869 0.0195312 0.214  35 216 0.0510836 0.0390625 0.0727  36 222 0.0510804 0.078125 0.628  37 228 0.0510582 0.0195312 0.0654  38 234 0.0510247 0.0390625 0.866  39 240 0.050988 0.078125 0.152  40 246 0.050988 0.15625 0.152  41 252 0.0509863 0.0390625 0.0974  42 258 0.0509863 0.078125 0.0974  43 264 0.0509842 0.0195312 0.212  44 270 0.0509817 0.0390625 0.063  45 276 0.0509764 0.078125 0.65  46 282 0.0509582 0.078125 0.14  47 288 0.0509582 0.15625 0.14  48 294 0.0509568 0.0390625 0.105  49 300 0.0509568 0.078125 0.105  50 306 0.0509554 0.0195312 0.182  51 312 0.0509538 0.0390625 0.0741  52 318 0.0509534 0.078125 0.396  53 324 0.0509467 0.0195312 0.0606  54 330 0.0509376 0.0390625 0.542  55 336 0.0509268 0.078125 0.128  56 342 0.0509268 0.15625 0.128  57 348 0.0509257 0.0390625 0.115  58 354 0.0509257 0.078125 0.115  59 360 0.0509248 0.0195312 0.157  Local minimum possible.  lsqcurvefit stopped because the final change in the sum of squares relative to  its initial value is less than the default value of the function tolerance.  <stopping criteria details>  x =  0.1812 -1.8910 2.7765 0.0156 0.0211  >> |

运行结果 3

代码分析：

扰动比较大。可以发现，迭代收敛变快了。

4题

Page 168 的1题：

取不同的初值计算下列平方和形式的非线性规划，尽可能求出所有局部极小点，进而找出全局极小点，并对不同算法（搜索方向、搜索步长、数值梯度与分析梯度等）的结果进行分析、比较。

1. 
2. 
3. 
4. 



**Solution:**

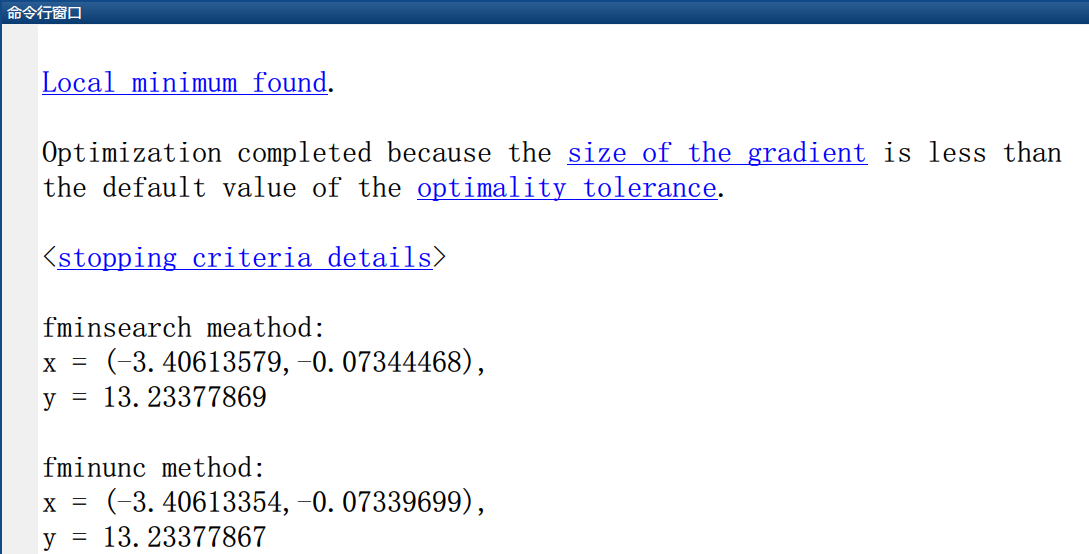
（1）

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | clc**;**clear all  fun1 **=** **@(**x**)**min**((**x**(**1**)^**2 **+** x**(**2**)** **-** 11**)^**2 **+** **(**x**(**1**)** **+** x**(**2**)^**2 **+** 7**)^**2**);**  x0 **=** **[**0**,**0**];**  **[**x\_search**,**fval\_search**]** **=** fminsearch**(**fun1**,**x0**);**  **[**x\_unc**,**fval\_unc**]** **=** fminunc**(**fun1**,**x0**);**  fprintf**(**'fminsearch meathod: \nx = (%3.8f,%3.8f), \ny = %3.8f\n\n'**,**...  x\_search**(**1**),**...  x\_search**(**2**),**...  fval\_search**);**  fprintf**(**'fminunc method: \nx = (%3.8f,%3.8f), \ny = %3.8f\n'**,**...  x\_unc**(**1**),**...  x\_unc**(**2**),**...  fval\_unc**);** |

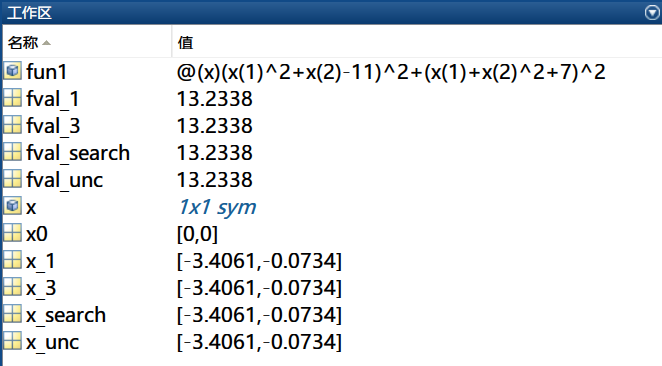
程序代码 6

运行结果：



运行结果 4

代码分析：

****

Figure

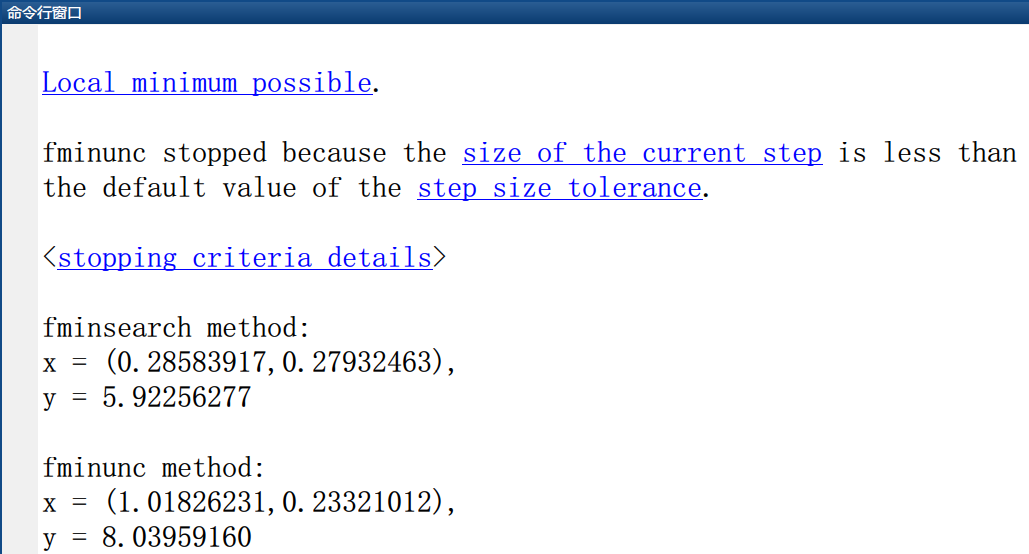
（2）

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | clc  fun2 **=** **@(**x**)**min**((**x**(**1**)^**2 **+** 12 **\*** x**(**2**)** **-** 1**)^**2 **+** **(**49 **\*** x**(**1**)^**2 ...  **+** 49 **\*** x**(**2**)^**2.**+** 84 **\*** x**(**1**)** **+** 2324 **\*** x**(**2**)** **-** 681**)^**2**);**  x0 **=** **[**1**,**0**];**  **[**x\_search**,**fval\_search**]** **=** fminsearch**(**fun2**,**x0**);**  **[**x\_unc**,**fval\_unc**]** **=** fminunc**(**fun2**,**x0**);**  fprintf**(**'fminsearch method: \nx = (%3.8f,%3.8f), \ny = %3.8f\n\n'**,**...  x\_search**(**1**),**...  x\_search**(**2**),**...  fval\_search**);**  fprintf**(**'fminunc method: \nx = (%3.8f,%3.8f), \ny = %3.8f\n'**,**...  x\_unc**(**1**),**x\_unc**(**2**),**fval\_unc**);** |

程序代码 7

运行结果：



运行结果 5

代码分析：

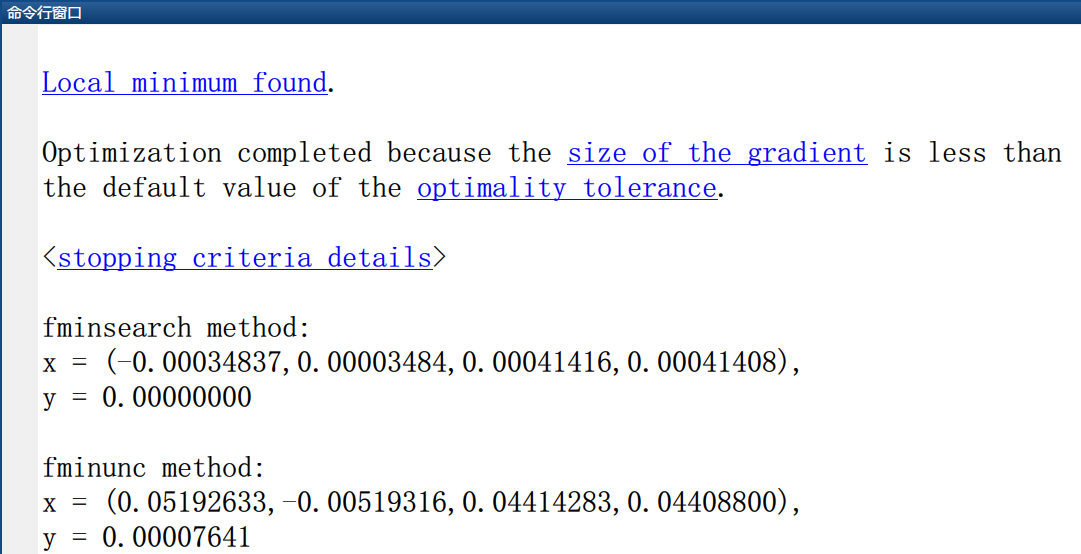
（3）

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | clc  fun2 **=** **@(**x**)**...  min**((**x**(**1**)** **+** 10 **\*** x**(**2**))^**2 **+** 5 **\*** **(**x**(**3**)** **-** x**(**4**))^**2 ...  **+** **(**x**(**2**)** **-** 2 **\*** x**(**3**))^**4 **+** 10 **\*** **(**x**(**1**)** **-** x**(**4**))^**4**);**  x0 **=** **[**10**,**10**,**10**,**10**];**  **[**x\_search**,**fval\_search**]** **=** fminsearch**(**fun2**,**x0**);**  **[**x\_unc**,**fval\_unc**]** **=** fminunc**(**fun2**,**x0**);**  fprintf**(**'fminsearch method:\nx = (%3.8f,%3.8f,%3.8f,%3.8f), \ny = %3.8f\n\n'**,**...  x\_search**(**1**),**...  x\_search**(**2**),**...  x\_search**(**3**),**...  x\_search**(**4**),**...  fval\_search**);**  fprintf**(**'fminunc method:\nx = (%3.8f,%3.8f,%3.8f,%3.8f), \ny = %3.8f\n'**,**...  x\_unc**(**1**),**...  x\_unc**(**2**),**...  x\_unc**(**3**),**...  x\_unc**(**4**),**...  fval\_unc**);** |

程序代码 8

运行结果：



运行结果 6

代码分析：

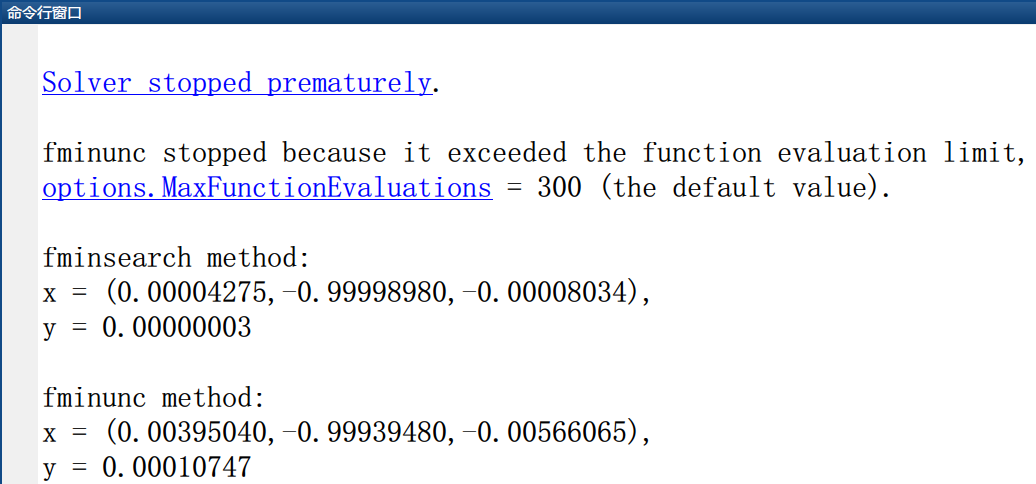
（4）

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | clc  fun4 **=** **@(**x**)**min**(**100 **\*** **((**x**(**3**)** **-** 10 **\*** Theta**(**x**(**1**),**x**(**2**)))^**2 ...  **+** **((**x**(**1**)^**2 **+** x**(**2**)^**2**)^(**1**/**2**)** **-** 1**)^**2**)** **+** x**(**3**)^**2**);**  x0 **=** **[**10**,**10**,**10**];**  **[**x\_search**,**fval\_search**]** **=** fminsearch**(**fun4**,**x0**);**  **[**x\_unc**,**fval\_unc**]** **=** fminunc**(**fun4**,**x0**);**  fprintf**(**'fminsearch method: \nx = (%3.8f,%3.8f,%3.8f), \ny = %3.8f\n\n'**,**...  x\_search**(**1**),**...  x\_search**(**2**),**...  x\_search**(**3**),**...  fval\_search**);**  fprintf**(**'fminunc method: \nx = (%3.8f,%3.8f,%3.8f), \ny = %3.8f\n'**,**...  x\_unc**(**1**),**...  x\_unc**(**2**),**...  x\_unc**(**3**),**...  fval\_unc**);**  %- Definition of function Theta -%  **function** y **=** Theta**(**a**,**b**)**  **if** a **>** 0  y **=** 1 **/** **(**2 **\*** pi**)** **\*** atan**(**a **./** b**);**  **else**  y **=** 1 **/** **(**2 **\*** pi**)** **\*** atan**(**a **./** b**)** **+** 1 **/** 2**;**  **end**  **end** |

程序代码 9

运行结果：



运行结果 7

代码分析：

5题

有一组数据 ，其中，由表7.9给出。现要用这组数据拟合函数



中的参数，初值可选为，用GN和LM两种方法求解。对作一扰动，即，为内的随机数，观察并分析迭代收敛是否会变慢。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | 0.844 | 12 | 0.718 | 23 | 0.478 |
| 2 | 0.908 | 13 | 0.685 | 24 | 0.467 |
| 3 | 0.932 | 14 | 0.658 | 25 | 0.457 |
| 4 | 0.936 | 15 | 0.628 | 26 | 0.48 |
| 5 | 0.925 | 16 | 0.603 | 27 | 0.438 |
| 6 | 0.908 | 17 | 0.58 | 28 | 0.431 |
| 7 | 0.881 | 18 | 0.558 | 29 | 0.424 |
| 8 | 0.85 | 19 | 0.538 | 30 | 0.42 |
| 9 | 0.818 | 20 | 0.522 | 31 | 0.414 |
| 10 | 0.784 | 21 | 0.506 | 32 | 0.411 |
| 11 | 0.751 | 22 | 0.49 | 33 | 0.406 |

**表 7.9**

**Solution**:

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | % filename: Data\_fit  cd **..;**cd **..;**cd **./**#Data**/**06  i **=** xlsread**(**'experiment\_6.xlsx'**,**'A2:A34'**);**  y **=** xlsread**(**'experiment\_6.xlsx'**,**'B2:B34'**);**  fun **=** **@(**x**,**t**)** x**(**1**)** **+** x**(**2**)** **\*** exp**(-**x**(**4**)** **\*** t**)** **+** x**(**3**)** **\*** exp**(-**x**(**5**)** **\*** t**);**  x0 **=** **[**0.5**,**1.5**,-**1**,**0.01**,**0.02**];**  **[**x**,**xx**,**xxx**,**xxxx**,**xxxxx**]** **=** lsqcurvefit**(**fun**,**x0**,**i**,**y**)**  y **=** y **+** rand**()/**2**;**  **[**x**,**xx**,**xxx**,**xxxx**,**xxxxx**]** **=** lsqcurvefit**(**fun**,**x0**,**i**,**y**)** |

运行结果：

代码分析：

6题

经济学中著名的Cobb-Douglas生产函数的一般形式为



其中分别表示产值、资金、劳动力，式中要由经济统计数据确定。现有《中国统计年鉴（2003）》给出的统计数据如表7.10所示，请分别用线性和非线性最小二乘拟合求出式中的，并解释的含义。

|  |  |  |  |
| --- | --- | --- | --- |
| **年份** | **总产值/万亿元** | **资金/万亿元** | **劳动力/亿人** |
| 1984 | 0.7171 | 0.0910 | 4.8179 |
| 1985 | 0.8964 | 0.2543 | 4.9873 |
| 1986 | 1.0202 | 0.3121 | 5.1282 |
| 1987 | 1.1962 | 0.3792 | 5.2783 |
| 1988 | 1.4928 | 0.4754 | 5.4334 |
| 1989 | 1.6909 | 0.441 | 5.5329 |
| 1990 | 1.8548 | 0.4517 | 6.4749 |
| 1991 | 2.1618 | 0.5595 | 6.5491 |
| 1992 | 2.6638 | 0.808 | 6.6152 |
| 1993 | 3.4634 | 1.3072 | 6.6808 |
| 1994 | 4.6759 | 1.7042 | 6.7455 |
| 1995 | 5.8478 | 2.0019 | 6.8065 |
| 1996 | 6.7885 | 2.2914 | 6.895 |
| 1997 | 7.4463 | 2.4941 | 6.982 |
| 1998 | 7.8345 | 2.8406 | 7.0637 |
| 1999 | 8.2068 | 2.9854 | 7.1394 |
| 2000 | 9.9468 | 3.2918 | 7.2085 |
| 2001 | 9.7315 | 3.7314 | 7.3025 |
| 2002 | 10.4791 | 4.35 | 7.374 |

表 7.10

**Solution:**

程序代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | % filename: Cobb\_Douglas  %% initialization  cd **..;**cd **..;**cd **./**#Data**/**06  Total\_product **=** xlsread**(**'experiment\_6.xlsx'**,**'Sheet2'**,**'B2:B20'**);**  Fund **=** xlsread**(**'experiment\_6.xlsx'**,**'Sheet2'**,**'C2:C20'**);**  Labour **=** xlsread**(**'experiment\_6.xlsx'**,**'Sheet2'**,**'D2:D20'**);**  fun **=** **@(**x**,**xdata**)** x**(**1**)** **.\*** xdata**(:,**1**).^**x**(**2**)** **.\*** xdata**(:,**2**).^**x**(**3**);**  x0 **=** **[**0.6**,**0.091**,**4.8179**];**  %% 非线性解法  xdata **=** **[**Labour Fund **];**  ydata **=** Total\_product**;**  x\_nonlin **=** lsqcurvefit**(**fun**,**x0**,**xdata**,**ydata**)**  %% 线性解法  lb **=** zeros**(**3**,**1**);**  C **=** **[**ones**(**19**,**1**),**log**(**Fund**),**log**(**Labour**)];**  ub **=** ones**(**3**,**1**);**  x **=** lsqlin**(**C**,**log**(**Total\_product**),[],[],[],[],**lb**,**ub**);**  x\_lin **=** **[**exp**(**x**(**1**)),**x**(**2**),**x**(**3**)]** |

程序代码 10

运行结果：



运行结果 8

代码分析：

要想用线性方法来完成这个题目，首先需要将方程进行转化，然后求解。如下所示：



这也就是，，，这样就有等式：。对于非线性拟合，直接根据数据与函数设计函数进行就可以了，比较简单。

# 四、实验过程

这次试验的难度有点大。

# 五、实验总结

# 六、参考文献

[1] 大学数学实验/姜启源，谢金星，邢文训，张立平，北京：清华大学出版社，2010.12

[2] MATLAB教程/张志涌，杨祖樱，北京：北京航空航天大学出版社，2015.1

# 七、教师评语