UCTP
TEST PROBLEMS FOR
UNCONSTRAINED
OPTIMIZATION

Hans Bruun Nielsen
1. Introduction

This report documents the MATLAB and Fortran77 implementation of a series of problems for testing algorithms for unconstrained minimization, i.e. for finding

$$x^* = \arg\min_{x \in \mathcal{D}} \{F(x)\},$$

(1.1)

where $F : \mathbb{R}^n \mapsto \mathbb{R}$ and $\mathcal{D}$ is some region around $x^*$, i.e. $x^*$ is a local minimizer. In all the problems the function is continuously differentiable, and the subprogram also return information that can be used to compute the gradient

$$g(x) = F'(x) = \begin{bmatrix} \frac{\partial F}{\partial x_1}(x) \\ \vdots \\ \frac{\partial F}{\partial x_n}(x) \end{bmatrix}. $$

(1.2)

Most of the problems are least squares problems. Given a function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ (with $m \geq n$). The corresponding function $F$ is

$$F(x) = \frac{1}{2} \|f(x)\|^2 = f(x)^\top f(x),$$

(1.3)

with the gradient

$$F'(x) = J(x)^\top f(x),$$

(1.4a)

where $J$ is the Jacobian matrix defined by

$$(J_f(x))_{ij} = \frac{\partial f_i}{\partial x_j}(x).$$

(1.4b)

The work on this package started in connection with [5] and was further developed in connection with [6]. Work on new subroutines for the package described in [4] lead to a Fortran version. There are other testing packages around, e.g. [1], but the present package is unique for providing (as far as possible) identical test problems in MATLAB and Fortran77. This is useful for testing minimization software developed in MATLAB and translated to a production version in Fortran.

Contents

1. Introduction 4
4. Test Problems 9
References 19

The package consists of two functions. The first is used to get the problem, and the other defines the problem for a least squares solver or a general minimizer.

2.1. User’s Guide to uctpget

A typical call is

\[ [\text{par, } x, \tau_0, \text{delta0}] = \text{uctpget(pno, m, n)} \]

Input Parameters

\- pno: Problem number, cf. Section 4. Integer in the range \( 1 \leq \text{pno} \leq 22 \).
\- m, n: Number of functions and variables, respectively. Dummy variables for some of the problems.

Output Parameters

\- par: struct representing the problem. In all cases the field par.p contains the problem number, par.pt gives the type of the problem (see uctpval), and par.xm holds \( x^* \). Other fields depend on the problem.
\- x: Starting vector.
\- tau0: Used for initial damping parameter in Levenberg-Marquardt type algorithms.
\- delta0: Used for initial radius in trust-region type algorithms.

2.2. User’s Guide to uctpval

A typical call is

\[ [F, \{, dF\}] = \text{uctpval}(x, \text{par}) \]

Input Parameters

\- x: Argument for the test function.
\- par: struct representing the problem, see uctpget.

Output Parameters

\- F: If \( \text{par.p} \leq 21 \) and \( \text{par.pt} = 0 \): Vector \( f(x) \).
Otherwise: Scalar \( F(x) \).
\- dF: Optional. If present, then
  - If \( \text{par.p} \leq 21 \) and \( \text{par.pt} = 0 \): Jacobian \( J(x) \).
  - Otherwise: Gradient \( F'(x) \).

2.3. Example

The MATLAB commands

\[ [\text{par } x \text{ tau0 } \text{delta0}] = \text{uctpget}(4,2,2); \ x \]

sets parameters for the Rosenbrock function treated as a least squares problem, and returns \( x = [-1.2 \ 1] \). Next,

\[ [f, d] = \text{uctpval}(x0, \text{par}) \]

returns

\[ f = \begin{bmatrix} -4.40 \\ 2.20 \end{bmatrix}, \ d = \begin{bmatrix} 24 & 10 \\ -1 & 0 \end{bmatrix} \]

Finally, the commands

\[ \text{par.pt} = 1; \ [f, d] = \text{uctpval}(x0, \text{par}) \]

give

\[ f = \begin{bmatrix} 12.10 \\ -44.00 \end{bmatrix}, \ d = \begin{bmatrix} -107.80 \\ -44.00 \end{bmatrix} \]

showing that \( \text{par.pt} \neq 0 \) changes the type of the test function.
3. Fortran User’s Guide

The package consists of three subroutines. The first is used to get the problem, and the other two define the problem for a general minimizer and a least squares solver, respectively.

The calling program must include the following declarations

\begin{verbatim}
INTEGER MM, PNO
DOUBLE PRECISION PD(2000)
COMMON /PROB/ PNO, MM, PD
\end{verbatim}

PROB transfers information about the chosen problem.

The subroutines call the following BLAS (see [2]) subroutines and functions

\begin{verbatim}
DCOPY DDOT DNM2 DGEMV DSYRK
\end{verbatim}

If BLAS is not available on your computer, you can compile uctp.f with the file minaux.f, that includes these subprograms together with a number of other BLAS routines. They were obtained from

\url{http://www.netlib.org/blas/blas.tgz}

At lines 24, 437 and 490 in the file uctp.f you can find instructions about how to modify the file if BLAS is available on your computer.

3.1. User’s Guide to UCTPG

A typical call is

\begin{verbatim}
CALL UCTPG(PNO,M,N,X,TAU0,DX0,XMIN,FXMIN)
\end{verbatim}

The parameters are

\begin{verbatim}
PNO  INTEGER. Problem number, cf. Section 4.
     Must satisfy 1 \leq PNO \leq 22.
M, N INTEGER. Number of functions and variables, respectively.
     Dummy variables for some of the problems. Must be positive
     and must satisfy M*N \leq 2000.
X    REAL*8 ARRAY with N elements. Starting vector.
TAU0 REAL*8. Can be used for initial damping parameter in
     Levenberg-Marquardt type algorithms.
\end{verbatim}

\begin{verbatim}
DX0  REAL*8. Can be used for initial radius in trust-region type
     algorithms.
XMIN REAL*8 ARRAY with N elements. Solution X*.
FXMIN REAL*8. Value at the solution, F(X*).
\end{verbatim}

3.2. User’s Guide to UCTPV

A typical call is

\begin{verbatim}
CALL UCTPV(N,X,DF,F)
\end{verbatim}

The parameters are

\begin{verbatim}
N    INTEGER. Number of variables.
X    REAL*8 ARRAY with N elements. Argument for the test
     function.
DF   REAL*8 ARRAY with N elements. Returns the gradient F'(X).
F    REAL*8. Returns the function value F(X).
\end{verbatim}

3.3. User’s Guide to UCTPV2

A typical call is

\begin{verbatim}
CALL UCTPV2(N,M,X,DF,F)
\end{verbatim}

The parameters are

\begin{verbatim}
N    INTEGER. Number of variables.
M    INTEGER. Number of functions.
X    REAL*8 ARRAY with N elements. Argument for the test
     function.
DF   REAL*8 ARRAY with M rows and N columns. Returns the Jacobian
     J(X).
F    REAL*8 ARRAY with M elements. Returns the vector f(X).
\end{verbatim}
3.4. Example
The following program (where we have omitted printing instructions) treats the Rosenbrock function.

```
INTEGER M,N,PNO
DOUBLE PRECISION X(2),XMIN(2),FXMIN,PD(2000),
& TAU,DX, F1,G(2), F2(2),J(2,2)
COMMON /PROPB/ PNO,M, PD

CALL UCTPG(4,2,2, X,TAU,DX, XMIN,FXMIN)
CALL UCTPV(2,X,G,F1)
CALL UCTPV2(2,2,X,J,F2)
STOP
END
```

We get the following results, cf. Section 2.3. \( \mathbf{x} = [-1.2 \; 1] \)

- \( F_1 = 12.10 \quad G = -107.80 \)  
- \( F_2 = -4.40 \quad J = 24 \; 10 \)  
- \( 2.20 \quad -1 \; 0 \)

4. Test Problems
The first 21 test problems are basically least squares problems, cf. (1.3). The first 17 problems were taken from [3], they originate from Argonne National Laboratory, USA. Most of these problems have appeared in the optimization literature, and we give the names which are usually connected with these functions.

Problem 18 was generated to simulate a data fitting problem, describing e.g. the concentration of medicine in the blood as function of time since taking the drug. The ordinates were found as \( y_i = 4(e^{-4t_i} - e^{-5t_i}) + r_i \), where the \( \{r_i\} \) were determined so that the given \( \mathbf{x}^* \) was the true least squares solution and \( F(\mathbf{x}^*) = 5 \cdot 10^{-3} \). Rounding the \( y_i \)-values to 6 decimals gives a slight perturbation in \( \mathbf{x}^* \) and \( F(\mathbf{x}^*) \). In problem 19 the separability [6] of the model is exploited and the number of parameters is reduced from \( n = 4 \) to \( n = 2 \).

Problems 20 and 21 are respectively a scaled and a “separated” version of problem 10. Finally, problem 22 is a “pure” minimization problem.

In the list below we use the following format

1) Dimension  
2) Function definition  
3) Starting point \( \mathbf{x}_0, \tau_0 \) and \( \Delta_0 \)  
4) Solution

The values \( \tau_0 \) and \( \Delta_0 \) are meant for use in connection with Levenberg-Marquardt and trust-region type algorithms, respectively.

In the presentation \( \mathbf{e} \) (\( \mathbf{E} \)) denotes the vector (matrix) of all ones, and \( \mathbf{I} \) is the identity matrix.

1. Linear function, full rank
   a) \( n \) variable, \( m \geq n \)
   b) \( \mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{e}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{I} - \frac{2}{m} \mathbf{E} \\ -\frac{2}{m} \mathbf{E} \end{bmatrix} \)
   c) \( \mathbf{x}_0 = \mathbf{e}, \quad \tau_0 = 10^{-8}, \quad \Delta_0 = 10 \)
   d) \( \mathbf{x}^* = -\mathbf{e}, \quad F(\mathbf{x}^*) = \frac{3}{2m+1} \)

2. Linear function, rank 1
   a) \( n \) variable, \( m \geq n \)
   b) \( \mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{e}, \quad \mathbf{A} = \begin{bmatrix} 1 \\ \vdots \\ m \end{bmatrix} \begin{bmatrix} 1 & \cdots & n \end{bmatrix} \)
   c) \( \mathbf{x}_0 = \mathbf{e}, \quad \tau_0 = 10^{-8}, \quad \Delta_0 = 10 \)
   d) Any \( \mathbf{x}^* \) such that \[ \begin{bmatrix} 1 & \cdots & n \end{bmatrix} \mathbf{x}^* = \frac{3}{2m+1} \]
   \[ F(\mathbf{x}^*) = \frac{m(m-1)}{4(2m+1)} \]
3. Linear function, rank 1 with zero columns and rows
   a) \( n \) variable, \( n \geq 3, \ m \geq n \)
   b) \( \mathbf{f(x)} = \mathbf{A}x - \mathbf{e}, \quad \mathbf{A} = \begin{bmatrix} 0 & 0^T & 0 \\ 0 & A & 0 \\ 0 & 0^T & 0 \end{bmatrix} \)
   c) \( \mathbf{x}_0 = \mathbf{e}, \ \tau_0 = 10^{-8}, \ \Delta_0 = 10 \)
   d) Any \( \mathbf{x}^* \) such that \( [0 \quad 2 \quad \cdots \quad n-2 \quad 0] \mathbf{x}^* = \frac{3}{2m-3} \)
   \( F(\mathbf{x}^*) = \frac{m^2 + 3m - 6}{4(2m - 3)} \)

4. Rosenbrock function
   a) \( n = m = 2 \)
   b) \( \mathbf{f(x)} = \begin{bmatrix} 10(x_2 - x_1^2) \\ 1 - x_1 \end{bmatrix} \)
   c) \( \mathbf{x}_0 = [-1.2, \ 1]^T, \ \tau_0 = 1, \ \Delta_0 = 1 \)
   d) \( \mathbf{x}^* = \mathbf{e}, \quad F(\mathbf{x}^*) = 0 \)

5. Helical valley function
   a) \( n = m = 3 \)
   b) \( \mathbf{f(x)} = \begin{bmatrix} 10 \left( x_3 - \frac{5}{\pi} \arctan \frac{x_2}{x_1} - \theta \right) \\ 10 \left( \sqrt{x_1^2 + x_2^2} - 1 \right) \end{bmatrix} \)
   \( \theta = \begin{cases} 0 & x_1 > 0 \\ 5 & x_1 < 0 \end{cases} \)
   c) \( \mathbf{x}_0 = [-1 \ 0 \ 0]^T, \ \tau_0 = 1, \ \Delta_0 = 1 \)
   d) \( \mathbf{x}^* = [1 \ 0 \ 0]^T, \quad F(\mathbf{x}^*) = 0 \)

6. Powell singular function
   a) \( n = m = 4 \)
   b) \( \mathbf{f(x)} = \begin{bmatrix} x_1 + 10x_2 \\ \sqrt{5(x_3 - x_4)} \\ (x_2 - 2x_3)^2 \\ \sqrt{10(x_1 - x_4)^2} \end{bmatrix} \)
   c) \( \mathbf{x}_0 = [3 \ -1 \ 0 \ 1]^T, \ \tau_0 = 10^{-8}, \ \Delta_0 = 1 \)
   d) \( \mathbf{x}^* = 0, \quad F(\mathbf{x}^*) = 0 \)

7. Freudenstein and Roth function
   a) \( n = m = 2 \)
   b) \( \mathbf{f(x)} = \begin{bmatrix} x_1 - x_2 \ast (2 - x_2 \ast (5 - x_2)) - 13 \\ x_1 - x_2 \ast (14 - x_2 \ast (1 + x_2)) - 29 \end{bmatrix} \)
   c) \( \mathbf{x}_0 = [0.5, \ -2]^T, \ \tau_0 = 1, \ \Delta_0 = 1 \)
   d) \( \mathbf{x}^* \simeq [11.4128, \ -0.896805]^T, \quad F(\mathbf{x}^*) \simeq 24.921 \)

8. Bard function
   a) \( n = 3, \ m = 15 \)
   b) \( f_i(x) = y_i \ast \left( x_1 + \frac{u_i}{x_2 v_i + x_3 w_i} \right) \)
   c) \( \mathbf{x}_0 = \mathbf{e}, \ \tau_0 = 10^{-8}, \ \Delta_0 = 1 \)
   d) \( \mathbf{x}^* \simeq [0.082411 \ 1.133036 \ 2.343695]^T, \quad F(\mathbf{x}^*) \simeq 4.10744 \times 10^{-3} \)

Data: \( u_i = i, \ v_i = 16 - i, \ w_i = \min\{u_i, v_i\} \) and

<table>
<thead>
<tr>
<th>( i )</th>
<th>( y_i )</th>
<th>( i )</th>
<th>( y_i )</th>
<th>( i )</th>
<th>( y_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.14</td>
<td>6</td>
<td>0.32</td>
<td>11</td>
<td>0.73</td>
</tr>
<tr>
<td>2</td>
<td>0.18</td>
<td>7</td>
<td>0.35</td>
<td>12</td>
<td>0.96</td>
</tr>
<tr>
<td>3</td>
<td>0.22</td>
<td>8</td>
<td>0.39</td>
<td>13</td>
<td>1.34</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>9</td>
<td>0.37</td>
<td>14</td>
<td>2.10</td>
</tr>
<tr>
<td>5</td>
<td>0.29</td>
<td>10</td>
<td>0.58</td>
<td>15</td>
<td>4.39</td>
</tr>
</tbody>
</table>
9. Kowalik and Osborne function
   a) $n = 4, \ m = 11$
      
   b) $f_i(x) = y_i - \frac{x_1 u_i (u_i + x_2)}{u_i (u_i + x_3) + x_4}$
      
   c) $x_0 = [0.25, 0.39, 0.415, 0.39]^\top, \ \tau_0 = 1, \ \Delta_0 = 0.1$
      
   d) $x^* \simeq [0.192807, 0.191282, 0.123057, 0.136062]^\top, \ F(x^*) \simeq 1.53753 \times 10^{-4}$

   Data:
   \[
   \begin{array}{|c|c|c|c|c|c|}
   \hline
   i & y_i & u_i & i & y_i & u_i \\
   \hline
   1 & 0.1957 & 4.0000 & 7 & 0.0435 & 0.1250 \\
   2 & 0.1947 & 2.0000 & 8 & 0.0342 & 1.0000 \\
   3 & 0.1735 & 1.0000 & 9 & 0.0323 & 0.0833 \\
   4 & 0.1600 & 0.5000 & 10 & 0.0235 & 0.0714 \\
   5 & 0.0844 & 0.2500 & 11 & 0.0246 & 0.0625 \\
   6 & 0.0627 & 0.1670 & & & \\
   \hline
   \end{array}
   \]

10. Meyer function
   a) $n = 3, \ m = 16$
      
   b) $f_i(x) = x_1 \exp \left( \frac{x_2}{t_i + x_3} \right) - y_i$
      
   c) $x_0 = [0.02, 4000, 250]^\top, \ \tau_0 = 1, \ \Delta_0 = 100$
      
   d) $x^* \simeq [0.00560964, 6181.35, 345.224]^\top, \ F(x^*) \simeq 43.9729$

   Data: $t_i = 45+5i$, $i$
   \[
   \begin{array}{|c|c|c|c|}
   \hline
   i & y_i & i & y_i \\
   \hline
   1 & 34780 & 7 & 11540 \\
   2 & 28610 & 8 & 9744 \\
   3 & 23650 & 9 & 8261 \\
   4 & 19630 & 10 & 7030 \\
   5 & 16370 & 11 & 6005 \\
   6 & 13720 & 12 & 5147 \\
   \hline
   \end{array}
   \]

11. Watson function
   a) $2 \leq n \leq 31, \ m = 31$
      
   b) $f_i(x) = \sum_{j=2}^{n} (j-1) t_j^{i-2} x_j - \left( \sum_{j=1}^{i-1} t_j^{i-1} x_j \right)^2 \ i=1, \ldots, 29$
      
   \[
   \begin{array}{|c|c|c|c|}
   \hline
   i & x_1 & x_2 - x_1^2 & 1 \\
   \hline
   30 & 1.234 & 0.5 & 31 \\
   \hline
   \end{array}
   \]
   where $t_i = i/29$.
      
   c) $x_0 = 0, \ \tau_0 = 10^{-8}, \ \Delta_0 = 1$
      
   d) $F(x^*) \simeq 1.43835 \times 10^{-3}$, $6.998801 \times 10^{-7}$, $2.361196 \times 10^{-10}$

12. Box 3-dimensional function
   a) $n = 3, \ m \geq 3$ variable
      
   b) $f_i(x) = e^{-x_1 t_i} - e^{-x_2 t_i} - x_3 (e^{-t_i} - e^{-10 t_i})$
      
   where $t_i = i/10$
      
   c) $x_0 = [0, 10, 20]^\top, \ \tau_0 = 10^{-8}, \ \Delta_0 = 1, \ \Delta_0 = 1$
      
   d) $x^* = [1, 10, 1]^\top, \ x^* = [10, 1, -1]^\top$ or $x^* = [\alpha, \alpha, \alpha]^\top$ for any $\alpha \in \mathbb{R}, \ F(x^*) = 0$

13. Jennrich and Sampson function
   a) $n = 2, \ m \geq 2$ variable
      
   b) $f_i(x) = 2 + 2i - (e^{x_1 i} + e^{x_2 i})$
      
   c) $x_0 = [0.3, 0.4]^\top, \ \tau_0 = 1, \ \Delta_0 = 0.05$
      
   d) $m = 5 : \ x_1^* = x_2^* \simeq 0.378468, F(x^*) \simeq 4.8879031$
      
   $m = 10 : \ x_1^* = x_2^* \simeq 0.257825, F(x^*) \simeq 62.1811$
      
   $m = 20 : \ x_1^* = x_2^* \simeq 0.165191, F(x^*) \simeq 724.740$
14. Brown and Dennis function
   a) $n = 4$, $m \geq 4$ variable
   b) $f_i(x) = (x_1 + x_2t_i - e^{t_i})^2 + (x_3 + x_4 \sin t_i - \cos t_i)^2$
   c) $x_0 = [25 \ 5 \ -5 \ -1]^T$, $\tau_0 = 10^{-3}$, $\Delta_0 = 0.5$
   d) $m = 5$:
      $x^* \approx [0.77781 \ 1.87187 \ 1.15301 \ -0.67095]^T$, $F(x^*) \approx 9.08309 \times 10^{-5}$
      $m = 10$:
      $x^* \approx [-0.18950 \ 3.45424 \ 1.32570 \ -1.336679]^T$, $F(x^*) \approx 7.21613 \times 10^{-1}$
      $m = 20$:
      $x^* \approx [-11.5944 \ 13.2036 \ -0.4034 \ 0.2368]^T$, $F(x^*) \approx 4.29112 \times 10^4$

15. Chebyquad function
   a) $n$ variable, $m \geq n$
   b) $f_i(x) = \frac{1}{n} \sum_{j=1}^{n} T_i(x_j) - y_i$
      where $T_i$ is the $i$th Chebyshev polynomial shifted to the interval $[0,1]$ and
      $y_i = \int_{0}^{1} T_i(x) dx = \left\{ \begin{array}{ll}
      0 & \text{for } i \text{ odd} \\
      -1/(i^2 - 1) & \text{for } i \text{ even}
   \end{array} \right.$
   c) $x_0 = \frac{1}{n+1} [1 \ \cdots \ n]^T$, $\tau_0 = 1$, $\Delta_0 = \frac{1}{n+1}$
   d) $n$ | $m = n$ | $m = 2n$
   
   |   | $F(x^*) \approx 0$ | $F(x^*) \approx 5.34479 \times 10^{-2}$ |
   | 5 | $F(x^*) \approx 1.75844 \times 10^{-3}$ | $F(x^*) \approx 2.94780 \times 10^{-2}$ |
   | 8 | $F(x^*) \approx 0$ | $F(x^*) \approx 3.55274 \times 10^{-2}$ |
   | 9 | $F(x^*) \approx 3.25198 \times 10^{-3}$ | $F(x^*) \approx 3.02614 \times 10^{-2}$ |
   | 10 | $F(x^*) \approx 3.25198 \times 10^{-3}$ | $F(x^*) \approx 3.02614 \times 10^{-2}$ |

16. Brown almost linear function
   a) $n$ variable, $m = n$
   b) $f_i(x) = \begin{align*}
      x_i + x_1 + \cdots + x_n - (n + 1) & \quad i = 1, \ldots, n - 1 \\
      x_1 \cdots x_n - 1 & \quad i = n
   \end{align*}$
   c) $x_0 = \delta e$, $\tau_0 = 1$, $\Delta_0 = 1$
   d) $x^* = [\alpha \ \cdots \ \alpha \ \alpha^{1-n}]^T$, where $\alpha$ is a real root (in particular $\alpha = 1$) in $na^n - (n+1)\alpha^{n-1} + 1 = 0$. $F(x^*) = 0$.
      Local minimum at $x^* = [0 \ \cdots \ 0 \ n+1]^T$. $F(x^*) = 0.5$

17. Osborne 1 function
   a) $n = 5$, $m = 33$
   b) $f_i(x) = y_i - (x_1 + x_2e^{-x_4 t_i} + x_3e^{-x_5 t_i})$
   c) $x_0 = [0.5 \ 1.5 \ -1 \ 0.01 \ 0.02]^T$, $\tau_0 = 10^{-8}$, $\Delta_0 = 0.1$
   d) $x^* \approx [0.37541 \ 1.93585 \ -1.46469 \ 0.01287 \ 0.02212]^T$
      $F(x^*) \approx 2.73245 \times 10^{-5}$
   Data: $t_i = 10(i-1)$,
18. Exponential fit, 4 parameters
   a) \( n = 4, \ m = 45 \)
   b) \( f_i(x) = y_i - (x_3 e^{x_1 t_i} + x_4 e^{x_2 t_i}) \)
   where \( t_i \) and \( y_i \) are given in the previous problem and the
   coefficients \( c_j(x) \) are found as the least squares solution to
   the linear problem \( c_1 e^{-x_1 t_i} + c_2 e^{-x_2 t_i} \approx y_i \), \( i = 1, \ldots, m \)
   c) \( x_0 = [-1, -2, 1, -1]^T \), \( \tau_0 = 10^{-3}, \ \Delta_0 = 1 \)
   d) \( x^* \approx [-4, -5, 4, -4]^T \), \( F(x^*) \approx 5 \cdot 10^{-3} \)

Data: \( t_i = 0.02i \),

<table>
<thead>
<tr>
<th></th>
<th>y_i</th>
<th></th>
<th>y_i</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.090542</td>
<td>16</td>
<td>0.288903</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>0.124569</td>
<td>17</td>
<td>0.300820</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>0.179367</td>
<td>18</td>
<td>0.303077</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>0.196564</td>
<td>19</td>
<td>0.283087</td>
<td>34</td>
</tr>
<tr>
<td>5</td>
<td>0.269707</td>
<td>20</td>
<td>0.262078</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>0.286027</td>
<td>21</td>
<td>0.281593</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>0.289892</td>
<td>22</td>
<td>0.267531</td>
<td>37</td>
</tr>
<tr>
<td>8</td>
<td>0.317475</td>
<td>23</td>
<td>0.218926</td>
<td>38</td>
</tr>
<tr>
<td>9</td>
<td>0.308191</td>
<td>24</td>
<td>0.225572</td>
<td>39</td>
</tr>
<tr>
<td>10</td>
<td>0.336995</td>
<td>25</td>
<td>0.205994</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>0.348371</td>
<td>26</td>
<td>0.197371</td>
<td>41</td>
</tr>
<tr>
<td>12</td>
<td>0.321337</td>
<td>27</td>
<td>0.182440</td>
<td>42</td>
</tr>
<tr>
<td>13</td>
<td>0.299423</td>
<td>28</td>
<td>0.183892</td>
<td>43</td>
</tr>
<tr>
<td>14</td>
<td>0.338972</td>
<td>29</td>
<td>0.152825</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>0.304763</td>
<td>30</td>
<td>0.174028</td>
<td>45</td>
</tr>
</tbody>
</table>

19. Exponential fit, 2 parameters
   a) \( n = 2, \ m = 45 \)
   b) \( f_i(x) = y_i - (c_1(x)e^{x_1 t_i} + c_2(x)e^{x_2 t_i}) \)
   where \( t_i \) and \( y_i \) are given in the previous problem and the
   coefficients \( c_j(x) \) are found as the least squares solution to
   the linear problem \( c_1 e^{-x_1 t_i} + c_2 e^{-x_2 t_i} \approx y_i \), \( i = 1, \ldots, m \)
   c) \( x_0 = [-1, -2]^T \), \( \tau_0 = 10^{-3}, \ \Delta_0 = 1 \)
   d) \( x^* \approx [-4, -5]^T \), \( F(x^*) \approx 5 \cdot 10^{-3} \)

20. Scaled Meyer function
   a) \( n = 3, \ m = 16 \)
   b) \( f_i(x) = x_1 \exp\left(\frac{10 x_2}{t_i + x_3} - 13\right) - 10^{-3} y_i \)
   where \( t_i = 0.45 + 0.05i \) and the \( \{y_i\} \) are given in problem 10.
   c) \( x_0 = [8.85, 4.0, 2.5]^T \), \( \tau_0 = 1, \ \Delta_0 = 1 \)
   d) \( x^* \approx [2.481778, 6.18135, 3.45224]^T \), \( F(x^*) \approx 43.9729 \cdot 10^{-6} \)

21. Separated Meyer function
   a) \( n = 2, \ m = 16 \)
   b) \( f_i(x) = c(x) \exp\left(\frac{x_1}{t_i + x_2}\right) - y_i \)
   where \( t_i = 45 + 5i \) and the \( \{y_i\} \) are given in problem 10. The
   coefficient \( c(x) \) is found as the least squares solution to the
   linear problem \( c \exp\left(\frac{x_1}{t_i + x_2}\right) \approx y_i \), \( i = 1, \ldots, m \)
   c) \( x_0 = [4000, 250]^T \), \( \tau_0 = 1, \ \Delta_0 = 100 \)
   d) \( x^* \approx [6181.35, 345.224]^T \), \( F(x^*) \approx 43.9729 \)

22. Exp and squares
   a) \( n \) variable, \( m = 1 \)
   b) \( F(x) = \exp\left(-\sum_{j=1}^{n} x_j^2 + \frac{1}{2} \sum_{j=1}^{n} j^2 x_j^2\right) \)
   c) \( x_0 = 0, \ \tau_0 = 10^{-3}, \ \Delta_0 = 1 \)
   d) \( x_j^* = e^{-y}/j^2 \) where \( y \) is the solution to
   \( (1 + \cdots + \frac{1}{m}) e^{-y} = y \)
References


