NUMERICAL ANALYSIS PROGRAMS IN MAPLE

This README file gives instructions for the Maple programs on the disk. The programs are designed to run on a minimally configured computer. Minimal hard disk space plus the Maple package are all that is needed. All the programs are given as ASCII files with the .TXT extension and as Maple worksheets with the .MWS extension.

The .TXT files can be altered using any word processor that creates a standard ASCII file. These are commonly called "Text Only" files. The .MWS files can be altered only within Maple.

To run the programs with the .TXT extension, open the file in Maple as a "maple text" file. When presented with the "text format choice", select the option "OK". The file will be loaded as a Maple worksheet. Once a worksheet is loaded into Maple, scroll to the top of the file and hit ENTER on the text portion of the line

```
> Restart;
```

This will reinitialize Maple and move the cursor to the next block. Keep hitting ENTER on the text portion of the program. Hitting ENTER on the first program line will compile the program and send the cursor to the bottom of the file. The last line of the file will be the command causing execution of the program.

Some of the programs require the input of large amounts of data or generate extensive output. To enable the programs to be run quickly and efficiently, the input data can be placed in data files and the data files read by the program. When the output is likely to be extensive, the programs have been constructed so that it is convenient to place the output directly into an output file. The programs will prompt you for the form of the input or output that you would like to use. For example, when running the program for Neville's method, NEVLLE31.MWS, using the defined data file NEVLLE31.DTA for the sample problem, you will first see a screen that states:

```
Choice of input method:
1.  Input entry by entry from the keyboard
2.  Input data from a text file
3.  Generate data using a function F
Choose 1, 2, or 3 please
```

If you choose `1` you will need to enter all the data for the program from the keyboard, and any mistake in a data entry will require the program to be rerun. Choosing `2` will lead to the input of the data file NEVLLE31.DTA. Choosing 3 will cause the program to prompt you for the input of the function F.

The Maple programs are contained in four subdirectories: MWS-5, TXT-5, MWS-8, and TXT-8. The MWS directories contain Maple worksheets for Windows (DOS) based versions of Maple. They illustrate the input and execution of the programs applied to the same problems. The TXT subdirectories contain ASCII files with the .TXT extension and can be run on any system on which Maple is installed. These files will be converted into Maple worksheets as described above.

The directories MWS-5 and TXT-5 are for releases of Maple earlier than release 6. The main difference is the syntax of the `evaln` command, which is used to implement functions of multiple variables. The older releases used syntax of the form

```
evaln( a . (1..3));
```

to obtain

$$a_1, a_2, a_3$$

The syntax in the newer versions utilizing MWS-8 and TXT-8 obtains this with the command

```
evaln( a || (1..3));
```

The only programs that are effected by this change are the .TXT and .MWS files for

RKO4SY57, NWTSY101, BROYM102, STPDC104, and CONTM104

PROGRAMS FOR CHAPTER 2

BISECTION METHOD "BISECT21.MWS" PAGE 36

This program uses the Bisection Method to approximate a root of the equation $f(x) = 0$ lying in the interval $[a, b]$. The sample problem uses

$$f(x) = x^3 + 4x^2 - 10.$$

INPUT: $a = 1, \quad b = 2, \quad TOL = 5 \times 10^{-4}, \quad N_0 = 20$

SECANT METHOD "SECANT22.MWS" PAGE 41

This program uses the Secant Method to approximate a root of the equation $f(x) = 0$. The sample problem uses

$$f(x) = \cos x - x.$$

INPUT: $p_0 = \frac{1}{2}, \quad p_1 = \frac{\pi}{4}, \quad TOL = 5 \times 10^{-4}, \quad N_0 = 15$

METHOD OF FALSE POSITION "FALPOS23.MWS" PAGE 42

This program uses the Method of False Position to approximate a root of the equation $f(x) = 0$. The sample problem uses

$$f(x) = \cos x - x.$$

INPUT: $p_0 = \frac{1}{2}, \quad p_1 = \frac{\pi}{4}, \quad TOL = 5 \times 10^{-4}, \quad N_0 = 15$

NEWTON'S METHOD "NEWTON24.MWS" PAGE 45

This program uses Newton's Method to approximate a root of the equation $f(x) = 0$. The sample problem uses

$$f(x) = \cos x - x \quad \text{with} \quad f'(x) = -\sin x - 1.$$

INPUT: $p_0 = \frac{\pi}{4}, \quad TOL = 5 \times 10^{-4}, \quad N_0 = 15$

MÜLLER'S METHOD "MULLER25.MWS" PAGE 58

This program uses Müller's Method to approximate a root of an arbitrary polynomial of the form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0.$$

The sample problem uses

$$f(x) = 16x^4 - 40x^3 + 5x^2 + 20x + 6.$$

INPUT: $n = 4, \quad a_0 = 6, \quad a_1 = 20, \quad a_2 = 5, \quad a_3 = -40, \quad a_4 = 16,$
$TOL = 0.00001, \quad N_0 = 30, \quad x_0 = \frac{1}{2}, \quad x_1 = -\frac{1}{2}, \quad x_2 = 0$

PROGRAMS FOR CHAPTER 3

NEVILLE ITERATED INTERPOLATION "NEVLLE31.MWS" PAGE 73

This program uses Neville's Iterated Interpolation Method to evaluate the n^{th} degree interpolating polynomial $P(x)$ on the $n+1$ distinct numbers $x_0, ..., x_n$ at the number x for a given function f. The sample problem considers the Bessel function of the first kind of order zero at $x = 1.5$.

INPUT: NEVLLE31.DTA, $n = 4$, $x = 1.5$

NEWTON INTERPOLATORY DIVIDED-DIFFERENCE FORMULA "DIVDIF32.MWS" PAGE 79

This program uses Newton's Interpolatory Divided-Difference Formula to evaluate the divided-difference coefficients of the $n^{\text{ th}}$ degree interpolatory polynomial $P(x)$ on the $n+1$ distinct numbers $x_0, ..., x_n$ for a given function f. The sample problem considers the Bessel function of the first kind of order zero.

INPUT: DIVDIF32.DTA, $n = 4$

HERMITE INTERPOLATION "HERMIT33.MWS" PAGE 88

This program uses Hermite's Interpolation Method to obtain the coefficients of the Hermite interpolating polynomial $H(x)$ on the $n+1$ distinct numbers $x_0, ..., x_n$ for a given function f. The sample problem considers the Bessel function of the first kind of order zero.

INPUT: HERMIT33.DTA, $n = 2$

NATURAL CUBIC SPLINE INTERPOLATION "NCUBSP34.MWS" PAGE 94

This program uses the Natural Cubic Spline Method to construct the free cubic spline interpolation S for a function f. The sample problem considers $f(x) = e^{2x}$ on the interval $[0, 1]$.

INPUT: (Select input option 2.) NCUBSP34.DTA, $n = 4$

CLAMPED CUBIC SPLINE INTERPOLATION "CCUBSP35.MWS" PAGE 94

This program uses the Clamped Cubic Spline Method to construct the clamped cubic spline interpolation S for the function f. The sample problem considers $f(x) = e^{2x}$ on the interval $[0, 1]$.

INPUT: (Select input option 2.) CCUBSP35.DTA, $n = 4$, $FPO = 2$, $FP1 = 2e^2$

BÉZIER CURVE "BEZIER36.MWS" PAGE 107

This program uses the Bézier Curve method to construct parametric curves to approximate given data. The sample program considers

$$\begin{aligned}
(x_0, y_0) &= (0, 0)\\
(x_0^+, y_0^+) &= (1/4, 1/4)\\
(x_1, y_1) &= (1, 1)\\
(x_1^-, y_1^-) &= (1/2, 1/2)\\
(x_1^+, y_1^+) &= (-1/2, -1/2)\\
(x_2, y_2) &= (2, 2)\\
(x_2^-, y_2^-) &= (-1, -1)
\end{aligned}$$

INPUT: BEZIER36.DTA, $n = 2$

PROGRAMS FOR CHAPTER 4

COMPOSITE SIMPSON'S RULE "CSIMPR41.MWS" PAGE 123

This program uses Composite Simpson's Rule to approximate

$$\int_a^b f(x)dx.$$

The sample problem uses

$$f(x) = \sin x, \quad \text{on} \quad [0, \pi].$$

INPUT: $a = 0, \quad b = \pi, \quad n = 10$

ROMBERG INTEGRATION "ROMBRG42.MWS" PAGE 135

This program uses the Romberg Method to approximate

$$\int_a^b f(x)dx.$$

The sample problem uses

$$f(x) = \sin x, \quad \text{on} \quad [0, \pi].$$

INPUT: $a = 0, \quad b = \pi, \quad n = 6$

ADAPTIVE QUADRATURE "ADAPQR43.MWS" PAGE 147

This program uses the Adaptive Quadrature Method to approximate

$$\int_a^b f(x)dx$$

within a given tolerance $TOL > 0$. The sample problem uses

$$f(x) = \frac{100}{x^2} \sin \frac{10}{x}, \quad \text{on} \quad [1, 3].$$

INPUT: $a = 1, \quad b = 3, \quad TOL = 0.0001, \quad N = 20$

COMPOSITE SIMPSON'S RULE "DINTGL44.MWS" PAGE 156

FOR DOUBLE INTEGRALS

This program uses the Composite Simpson's Rule for Double Integrals to approximate

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\ dydx.$$

The sample problem uses

$$f(x, y) = e^{\frac{y}{x}}$$

with $c(x) = x^3$, $d(x) = x^2$, $a = 0.1$ and $b = 0.5$.

INPUT: $a = 0.1$, $b = 0.5$, $m = 10$, $n = 10$

GAUSSIAN QUADRATURE "DGQINT45.MWS" PAGE 156

FOR DOUBLE INTEGRALS

This program uses Gaussian Quadrature to approximate

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\ dydx.$$

The sample problem uses $f(x, y) = e^{y/x}$ with $c(x) = x^3$, $d(x) = x^2$, $a = 0.1$ and $b = 0.5$.

INPUT: $a = 0.1$, $b = 0.5$, $m = 5$, $n = 5$

This program uses the Gaussian Quadrature to approximate

$$\int_a^b \int_{c(x)}^{d(x)} \int_{\alpha(x,y)}^{\beta(x,y)} f(x,y,z)\; dzdydx.$$

The sample problem uses

$$f(x,y,z) = \sqrt{x^2+y^2}$$

with

$$\alpha(x,y) = \sqrt{x^2+y^2}, \quad \beta(x,y) = 2,$$

$$c(x) = 0.0, \quad d(x) = \sqrt{4-x^2}, \quad a = 0, \quad \text{and} \quad b = 2.$$

INPUT: $a = 0, \quad b = 2, \quad m = 5, \quad n = 5, \quad p = 5$

PROGRAMS FOR CHAPTER 5

EULER METHOD "EULERM51.MWS" PAGE 183

This program uses the Euler Method to approximate the solution of an initial value problem of the form

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \le t \le b.$$

The sample problem uses

$$f(t, y) = y - t^2 + 1, \qquad y(0) = 0.5, \qquad 0 \le t \le 2.$$

INPUT: $a = 0, \quad b = 2, \quad \alpha = 0.5, \quad N = 10$

RUNGE-KUTTA METHOD "RKOR4M52.MWS" PAGE 196
OF ORDER 4

This program uses the Runge-Kutta Method of order 4 to approximate the solution of the initial value problem of the form

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \le t \le b.$$

The sample problem uses

$$f(t, y) = y - t^2 + 1, \qquad y(0) = 0.5, \qquad 0 \le t \le 2.$$

INPUT: $a = 0, \quad b = 2, \quad \alpha = 0.5, \quad N = 10$

ADAMS FOURTH-ORDER "PRCORM53.MWS" PAGE 173
PREDICTOR-CORRECTOR METHOD

This program uses the Adams Fourth-Order Predictor-Corrector Method to approximate the solution of an initial value problem of the form

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \le t \le b.$$

The sample problem uses

$$f(t, y) = y - t^2 + 1, \qquad y(0) = 0.5, \qquad 0 \le t \le 2.$$

INPUT: $a = 0, \quad b = 2, \quad \alpha = 0.5, \quad N = 10$

EXTRAPOLATION METHOD "EXTRAP54.MWS" PAGE 211

This program uses the Extrapolation Method to approximate the solution of an initial value problem of the form

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \le t \le b$$

to within a given tolerance.

The sample problem uses

$$f(t, y) = y - t^2 + 1, \qquad y(0) = 0.5, \qquad 0 \le t \le 2.$$

INPUT: $a = 0, \quad b = 2, \quad \alpha = 0.5, \quad TOL = 0.00001, \quad HMIN = 0.01,$
$HMAX = 0.25$

RUNGE-KUTTA-FEHLBERG METHOD "RKFVSM55.MWS" PAGE 218

This program uses the Runge-Kutta-Fehlberg Method to approximate the solution of the initial value problem of the form

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \le t \le b$$

to within a given tolerance. The sample problem uses

$$f(t, y) = y - t^2 + 1, \qquad y(0) = 0.5, \qquad 0 \le t \le 2.$$

INPUT: $a = 0, \quad b = 2, \quad \alpha = 0.5, \quad TOL = 0.00001, \quad HMIN = 0.01,$
$HMAX = 0.25$

ADAMS VARIABLE-STEPSIZE "VPRCOR56.MWS" PAGE 221
PREDICTOR-CORRECTOR METHOD

This program uses the Adams Variable Stepsize Predictor-Corrector Method to approximate the solution of an initial value problem of the form

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \leq t \leq b$$

to within a given tolerance.

The sample problem uses

$$f(t, y) = y - t^2 + 1, \qquad y(0) = 0.5, \qquad 0 \leq t \leq 2.$$

INPUT: $a = 0, \quad b = 2, \quad \alpha = 0.5, \quad TOL = 0.00001, \quad HMIN = 0.01,$
$HMAX = 0.25$

RUNGE-KUTTA METHOD FOR SYSTEMS "RKO4SY57.MWS" PAGE 225
OF DIFFERENTIAL EQUATIONS

This program uses the Runge-Kutta for Systems of Differential Equations Method to approximate a the solution of the mth-order system of first-order initial value problems. The sample problem considers the second order system

$$f_1(u_1, u_2) = -4u_1 + 3u_2 + 6, \quad u_1(0) = 0$$

$$f_2(u_1, u_2) = -2.4u_1 + 1.6u_2 + 3.6, \quad u_2(0) = 0.$$

INPUT: $a = 0, \quad b = 0.5, \quad \alpha_1 = 0, \quad \alpha_2 = 0, \quad N = 5$

This program uses the Trapezoidal Method with Newton Iteration to approximate the solution to the initial value problem

$$y' = f(t, y), \qquad y(a) = \alpha, \qquad a \le t \le b.$$

The sample problem uses

$$f(t, y) = 5e^{5t}(y - t)^2 + 1, \qquad y(0) = -1, \qquad 0 \le t \le 1.$$

with

$$f_y(t, y) = 10e^{5t}(y - t).$$

INPUT: $a = 0, \quad b = 1, \quad \alpha = -1, \quad N = 5, \quad TOL = 0.000001, \quad M = 10$

PROGRAMS FOR CHAPTER 6

GAUSSIAN ELIMINATION WITH BACKWARD SUBSTITUTION — "GAUSEL61.MWS" — PAGE 248

This program uses the Gaussian Elimination with Backward Substitution Method to solve an $n \times n$ linear system of the form $A\mathbf{x} = \mathbf{b}$. The sample problem solves the linear system

$$\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= -8 \\ 2x_1 - 2x_2 + 3x_3 - 3x_4 &= -20 \\ x_1 + x_2 + x_3 &= -2 \\ x_1 - x_2 + 4x_3 + 3x_4 &= 4. \end{aligned}$$

INPUT: GAUSEL61.DTA, $n = 4$

GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING — "GAUMPP62.MWS" — PAGE 254

This program uses the Gaussian Elimination with Partial Pivoting Method to solve an $n \times n$ linear system. The sample problem solves the linear system

$$\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= -8 \\ 2x_1 - 2x_2 + 3x_3 - 3x_4 &= -20 \\ x_1 + x_2 + x_3 &= -2 \\ x_1 - x_2 + 4x_3 + 3x_4 &= 4. \end{aligned}$$

INPUT: GAUMPP62.DTA, $n = 4$

GAUSSIAN ELIMINATION WITH SCALED PARTIAL PIVOTING "GAUSPP63.MWS" PAGE 255

This program uses the Gaussian Elimination with Scaled Partial Pivoting Method to solve an $n \times n$ linear system. The sample problem solves the linear system

$$\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= -8 \\ 2x_1 - 2x_2 + 3x_3 - 3x_4 &= -20 \\ x_1 + x_2 + x_3 &= -2 \\ x_1 - x_2 + 4x_3 + 3x_4 &= 4. \end{aligned}$$

INPUT: GAUSPP63.DTA, $n = 4$

LU FACTORIZATION "LUFACT64.MWS" PAGE 274

This program uses the Direct Factorization Method to factor the $n \times n$ matrix A into the product $A = LU$ of a lower triangular matrix L and an upper triangular matrix U. The matrix factored in the sample problem is

$$A = \begin{bmatrix} 6 & 2 & 1 & -1 \\ 2 & 4 & 1 & 0 \\ 1 & 1 & 4 & -1 \\ -1 & 0 & -1 & 3 \end{bmatrix}$$

INPUT: LUFACT64.DTA, $n = 4$, $ISW = 1$

CHOLESKI'S METHOD "CHOLFC65.MWS" PAGE 280

This program uses the Choleski Method to factor the positive definite $n \times n$ matrix A into the product LL^t, where L is a lower triangular matrix. The matrix factored in the sample problem is

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}$$

INPUT: CHOLFC65.DTA, $n = 3$

LDL^t FACTORIZATION "LDLFCT66.MWS" PAGE 281

This program uses the LDL^t Factorization Method to factor the positive definite $n \times n$ matrix A into the product LDL^t, where L is a lower triangular matrix and D is a diagonal matrix. The matrix factored in the sample problem is

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}$$

INPUT: LDLFCT66.DTA, $n = 3$

CROUT REDUCTION FOR "CRTRLS67.MWS" PAGE 235
TRIDIAGONAL LINEAR SYSTEMS

This program uses the Crout Reduction for Tridiagonal Linear Systems Method to solve a tridiagonal $n \times n$ linear system. The sample system is

$$\begin{aligned} 2x_1 - x_2 \qquad\qquad &= 1 \\ -x_1 + 2x_2 - x_3 \qquad &= 0 \\ - x_2 + 2x_3 - x_4 &= 0 \\ - x_3 + 2x_4 &= 1. \end{aligned}$$

INPUT: CRTRLS67.DTA, $n = 4$

PROGRAMS FOR CHAPTER 7

JACOBI ITERATIVE METHOD "JACITR71.MWS" PAGE 309

This program uses the Jacobi Iterative Method to approximate the solution to the $n \times n$ linear system $A\mathbf{x} = \mathbf{b}$, given an initial approximation $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})^t$.

The sample problem approximates the solution to the linear system

$$\begin{aligned}
10x_1 - x_2 + 2x_3 \quad &= 6\\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25\\
2x_1 - x_2 + 10x_3 - x_4 &= -11\\
3x_2 - x_3 + 8x_4 &= 15.
\end{aligned}$$

starting with the initial vector $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$.

INPUT: JACITR71.DTA, $n = 4$, $TOL = 0.001$, $N = 30$

GAUSS–SEIDEL ITERATIVE METHOD "GSEITR72.MWS" PAGE 311

This program uses the Gauss-Seidel Iterative Method to approximate the solution to the $n \times n$ linear system $A\mathbf{x} = \mathbf{b}$, given an initial approximation $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})^t$. The sample problem approximates the solution to the linear system

$$\begin{aligned}
10x_1 - x_2 + 2x_3 \quad &= 6\\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25\\
2x_1 - x_2 + 10x_3 - x_4 &= -11\\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}$$

starting with the initial vector $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$.

INPUT: GSEITR72.DTA, $n = 4$, $TOL = 0.001$, $N = 30$

SUCCESSIVE–OVER–RELAXATION (SOR) METHOD "SORITR73.MWS" PAGE 313

This program uses the Successive-Over-Relaxation Method to approximate the solution to the $n \times n$ linear system $A\mathbf{x} = \mathbf{b}$, given a parameter ω and an initial approximation

$$\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right)^t.$$

The sample problem approximates the solution to the linear system

$$\begin{aligned} 4x_1 + 3x_2 &= 24 \\ 3x_1 + 4x_2 - x_3 &= 30 \\ - x_2 + 4x_3 &= -24 \end{aligned}$$

starting with the initial vector $\mathbf{x}^{(0)} = (1, 1, 1)^t$.

INPUT: SORITR73.DTA, $n = 3$, $TOL = 0.001$, $N = 30$, $\omega = 1.25$

ITERATIVE REFINEMENT "ITREFN74.MWS" PAGE 320

This program uses the Iterative Refinement Method to approximate a solution to the linear system $A\mathbf{x} = \mathbf{b}$ when A is suspected to be ill-conditioned. The sample problem considers the linear system

$$\begin{aligned} 3.333x_1 + 15920x_2 - 10.333x_3 &= 15913 \\ 2.222x_1 + 16.710x_2 + 9.6120x_3 &= 28.544 \\ 1.5611x_1 + 5.1791x_2 + 1.6852x_3 &= 8.4254. \end{aligned}$$

INPUT: ITREFN74.DTA, $n = 3$, $N = 25$, $D = 5$, $TOL = 0.00001$

This program uses the Preconditioned Conjugate Gradient Method to approximate the solution to the $n \times n$ linear system $A\mathbf{x} = \mathbf{b}$, given a preconditioning matrix C^{-1} and an initial approximation

$$\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right)^t.$$

The sample problem approximates the solution to the linear system

$$\begin{aligned} 4x_1 + 3x_2 &= 24 \\ 3x_1 + 4x_2 - x_3 &= 30 \\ - x_2 + 4x_3 &= -24 \end{aligned}$$

starting with the initial vector $\mathbf{x}^{(0)} = (0, 0, 0)^t$.

INPUT: PCCGRD75.DTA, $n = 3$, $TOL = 0.001$, $N = 3$

PROGRAMS FOR CHAPTER 8

PADÉ APPROXIMATION "PADEMD81.MWSS" PAGE 365

This program uses Padé Approximation to compute the rational approximation

$$r(x) = \frac{p_0 + p_1 x + \cdots + p_n x^n}{q_0 + q_1 x + \cdots + q_m x^m}$$

to a function $f(x)$ given its Maclaurin series $a_0 + a_1 x + a_2 x^2 + \cdots$. The sample problem uses $f(x) = e^{-x}$, where $a_0 = 1, \quad a_1 = -1, \quad a_2 = \frac{1}{2}, \quad a_3 = -\frac{1}{6}, \quad a_4 = \frac{1}{24}, \quad a_5 = -\frac{1}{120}$.

INPUT: PADEMD81.DTA, $m = 2, \quad n = 3$

FAST FOURIER TRANSFORM METHOD "FFTRNS82.MWS" PAGE 378

This program uses the Fast Fourier Transform Method to compute the coefficients in the discrete trigonometric approximation for a given set of data. The sample problem constructs an approximation to the function

$$f(x) = x^4 - 3x^3 + 2x^2 - \tan x(x-2)$$

on the interval [0,2].

INPUT: (Select input option 3.) $m = 4$

PROGRAMS FOR CHAPTER 9

POWER METHOD "POWERM91.MWS" PAGE 390

This program uses the Power Method to approximate the dominant eigenvalue and an associated eigenvector of an $n \times n$ matrix A given a nonzero vector $\mathbf{x}^{(0)}$. The sample problem considers the matrix

$$A = \begin{bmatrix} -4 & 14 & 0 \\ -5 & 13 & 0 \\ -1 & 0 & 2 \end{bmatrix}$$

with $\mathbf{x}^{(0)} = (1, 1, 1)^t$ as the initial approximation to the eigenvector.

INPUT: POWERM91.DTA, $n = 3$, $TOL = 0.0001$, $N = 30$

SYMMETRIC POWER METHOD "SYMPWR92.MWS" PAGE 392

This program uses the Symmetric Power Method to approximate the dominant eigenvalue and an associated eigenvector of a symmetric $n \times n$ matrix A given a nonzero vector $\mathbf{x}^{(0)}$. The sample problem considers the symmetric matrix

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{bmatrix}$$

with $\mathbf{x}^{(0)} = (1, 0, 0)^t$ as the initial approximation to the eigenvector.

INPUT: SYMPWR92.DTA, $n = 3$, $TOL = 0.0001$, $N = 25$

INVERSE POWER METHOD "INVPWR93.MWS" PAGE 396

This program uses the Inverse Power Method to approximate an eigenvalue nearest to a given number q and an associated eigenvector of an $n \times n$ matrix A. The sample problem considers the matrix

$$A = \begin{bmatrix} -4 & 14 & 0 \\ -5 & 13 & 0 \\ -1 & 0 & 2 \end{bmatrix}$$

with $\mathbf{x}^{(0)} = (1, 1, 1)^t$ as the initial approximation to the eigenvector and the number q defined by

$$q = \frac{\mathbf{x}^{(0)t} A \mathbf{x}^{(0)}}{\mathbf{x}^{(0)t} \mathbf{x}^{(0)}}.$$

INPUT: INVPWR93.DTA, $n = 3$, $TOL = 0.0001$, $N = 25$

WIELANDT DEFLATION "WIEDEF94.MWS" PAGE 397

This program uses the Wielandt Deflation Method to approximate the second most dominant eigenvalue and an associated eigenvector of the $n \times n$ matrix A given a nonzero vector $\mathbf{x}^{(0)}$. The sample problem considers the matrix

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{bmatrix}$$

which has the dominant eigenvalue $\lambda = 6$ and associated eigenvector $\mathbf{v} = (1, -1, 1)^t$. The initial approximation $\mathbf{x}^{(0)} = (1, 1)^t$.

INPUT: WIEDEF94.DTA, $n = 3$, $TOL = 0.0001$, $N = 30$

HOUSEHOLDER'S METHOD "HSEHLD95.MWS" PAGE 404

This program uses the Householder Method to obtain a symmetric tridiagonal matrix that is similar to a given symmetric matrix A. The sample problem considers the matrix

$$A = \begin{bmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{bmatrix}$$

INPUT: HSEHLD95.DTA, $n = 4$

QR METHOD "QRSYMT96.MWS" PAGE 410

This program uses the QR Method to obtain the eigenvalues of a symmetric, tridiagonal $n \times n$ matrix of the form

$$A = \begin{bmatrix} a_1^{(1)} & b_2^{(1)} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ b_2^{(1)} & a_2^{(1)} & b_3^{(1)} & \ddots & & & & \vdots \\ 0 & b_3^{(1)} & a_3^{(1)} & b_4^{(1)} & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & \ddots & b_{n-1}^{(1)} & a_{n-1}^{(1)} & b_n^{(1)} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & b_n^{(1)} & a_n^{(1)} \end{bmatrix}$$

The sample problem considers the matrix

$$A = \begin{bmatrix} a_1^{(1)} & b_2^{(1)} & 0 \\ b_2^{(1)} & a_2^{(1)} & b_3^{(1)} \\ 0 & b_3^{(1)} & a_3^{(1)} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

INPUT: QRSYMT96.DTA, $n = 3$, $TOL = 0.00001$, $M = 30$

PROGRAMS FOR CHAPTER 10

NEWTON'S METHOD FOR SYSTEMS "NWTSY101.MWS" PAGE 421

This program uses Newton's Method for Systems to approximate the solution of the nonlinear system of equations $\mathbf{F}(x) = \mathbf{0}$ given an initial approximation $\mathbf{x}^{(0)}$. The sample problem uses

$$\mathbf{F}(x) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))^t$$

where

$$\mathbf{x} = (x_1, x_2, x_3)^t$$

and

$$\begin{aligned}
f_1(x_1, x_2, x_3) &= 3x_1 - \cos(x_2 x_3) - 0.5\\
f_2(x_1, x_2, x_3) &= x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06\\
f_3(x_1, x_2, x_3) &= e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3}.
\end{aligned}$$

INPUT: $n = 3, \quad TOL = 0.00001, \quad N = 25, \quad \mathbf{x}^{(0)} = (0.1, 0.1, -0.1)^t$

BROYDEN'S METHOD "BROYM102.MWS" PAGE 421

This program uses the Broyden Method to approximate the solution of the nonlinear system of equations $\mathbf{F}(x) = \mathbf{0}$ given an initial approximation $\mathbf{x}^{(0)}$. The sample problem uses

$$\mathbf{F}(x) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))^t$$

where

$$\mathbf{x} = (x_1, x_2, x_3)^t$$

and

$$\begin{aligned}
f_1(x_1, x_2, x_3) &= 3x_1 - \cos(x_2 x_3) - 0.5\\
f_2(x_1, x_2, x_3) &= x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06\\
f_3(x_1, x_2, x_3) &= e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3}.
\end{aligned}$$

INPUT: $n = 3, \quad TOL = 0.00001, \quad N = 25, \quad \mathbf{x}^{(0)} = (0.1, 0.1, -0.1)^t$

STEEPEST DESCENT METHOD "STPDC103.MWS" PAGE 421

This program uses the Steepest Descent Method to approximate a solution to the minimum of the function

$$g(\mathbf{x}) = \sum_{i=1}^{n} [f_i(\mathbf{x})]^2$$

given an initial approximation $\mathbf{x}^{(0)}$. This also approximates a zero of

$$\mathbf{F}(x) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))^t.$$

The sample problem uses

$$\mathbf{F}(x) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))^t \quad \text{where} \quad \mathbf{x} = (x_1, x_2, x_3)^t$$

and

$$\begin{aligned}
f_1(x_1, x_2, x_3) &= 3x_1 - \cos(x_2 x_3) - 0.5 \\
f_2(x_1, x_2, x_3) &= x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 \\
f_3(x_1, x_2, x_3) &= e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3}.
\end{aligned}$$

INPUT: $n = 3, \quad TOL = 0.05, \quad N = 10, \quad \mathbf{x}^{(0)} = (0.5, 0.5, 0.5)^t$

CONTINUATION METHOD "CONTM104.MWS" PAGE 446

This program uses the Continuation Method with the Runge-Kutta method of Order Four to approximate the solution of the nonlinear system of equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ given an initial approximation $\mathbf{x}^{(0)}$. The sample problem uses

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}))^t, \quad \text{where} \quad \mathbf{x} = (x_1, x_2, x_3)^t$$

and

$$\begin{aligned}
f_1(x_1, x_2, x_3) &= 3x_1 - \cos(x_2 x_3) - 0.5 \\
f_2(x_1, x_2, x_3) &= x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 \\
f_3(x_1, x_2, x_3) &= e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3}.
\end{aligned}$$

INPUT: $n = 3, \quad N = 1, \quad \mathbf{x}^{(0)} = (0, 0, 0)^t$

PROGRAMS FOR CHAPTER 11

LINEAR SHOOTING METHOD "LINST111.MWS" PAGE 452

This program uses the Linear Shooting Method to approximate the solution of a linear two-point boundary-value problem

$$y'' = p(x)y' + q(x)y + r(x), \quad a \le x \le b, \quad y(a) = \alpha, \quad y(b) = \beta.$$

The sample problem considers the boundary-value problem

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \quad y(1) = 1, \quad y(2) = 2.$$

INPUT: $a = 1, \quad b = 2, \quad \alpha = 1, \quad \beta = 2, \quad N = 10$

LINEAR FINITE-DIFFERENCE METHOD "LINFD112.MWS" PAGE 458

This program uses the Linear Finite-Difference Method to approximate the solution of a linear two-point boundary-value problem

$$y'' = p(x)y' + q(x)y + r(x), \quad a \le x \le b, \quad y(a) = \alpha, \quad y(b) = \beta.$$

The sample problem considers the boundary-value problem

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \quad y(1) = 1, \quad y(2) = 2.$$

INPUT: $a = 1, \quad b = 2, \quad \alpha = 1, \quad \beta = 2, \quad N = 9$

NONLINEAR SHOOTING METHOD "NLINS113.MWS" PAGE 466

This program uses the Nonlinear Shooting Method to approximate the solution of a nonlinear two-point boundary-value problem

$$y'' = f(x, y, y'), \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta.$$

The sample problem considers the boundary-value problem

$$y'' = 4 + 0.25x^3 - 0.125yy', \quad y(1) = 17, \quad y(3) = \frac{43}{3}$$

where

$$f_y(x, y, y') = -\frac{y'}{8} \quad \text{and} \quad f_{y'}(x, y, y') = -\frac{y}{8}.$$

INPUT: $a = 1, \quad b = 3, \quad \alpha = 17, \quad \beta = \frac{43}{3}, \quad N = 20, \quad TOL = 0.0001,$
$M = 25$

NONLINEAR FINITE-DIFFERENCE METHOD "NLFDM114.MWS" PAGE 470

This program uses the Nonlinear Finite-Difference Method to approximate the solution to a nonlinear two-point boundary-value problem

$$y'' = f(x, y, y'), \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta.$$

The sample problem considers the boundary-value problem

$$y'' = 4 + 0.25x^3 - 0.125yy', \quad y(1) = 17, \quad y(3) = \frac{43}{3}$$

where

$$f_y(x, y, y') = -\frac{y'}{8} \quad \text{and} \quad f_{y'}(x, y, y') = -\frac{y}{8}.$$

INPUT: $a = 1, \quad b = 3, \quad \alpha = 17, \quad \beta = \frac{43}{3}, \quad N = 19, \quad TOL = 0.0001,$
$M = 25$

PIECEWISE LINEAR "PLRRG115.MWS" PAGE 479

RAYLEIGH-RITZ METHOD

This program uses the Piecewise Linear Rayleigh-Ritz Method to approximate the solution to a two-point boundary-value problem

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + q(x)y = f(x), \quad 0 \le x \le 1, \quad y(0) = y(1) = 0.$$

The sample problem uses the differential equation

$$-y'' + \pi^2 y = 2\pi^2 \sin \pi x, \quad y(0) = 0, \quad y(1) = 0.$$

INPUT: $n = 9$, PLRRG115.DTA

CUBIC SPLINE "CSRRG116.MWS" PAGE 483

RAYLEIGH-RITZ METHOD

This program uses the Cubic Spline Rayleigh-Ritz Method to approximate the solution of a boundary-value problem

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + q(x)y = f(x), \quad 0 \le x \le 1, \quad y(0) = y(1) = 0.$$

The sample problem uses the differential equation

$$-y'' + \pi^2 y = 2\pi^2 \sin \pi x, \quad y(0) = 0, \quad y(1) = 0.$$

INPUT: $n = 9$, $f'(0) = 2\pi^3$, $f'(1) = -2\pi^3$, $p'(0) = 0$, $p'(1) = 0$, $q'(0) = 0$, $q'(1) = 0$

PROGRAMS FOR CHAPTER 12

POISSON EQUATION "POIFD121.MWS" PAGE 498

FINITE-DIFFERENCE METHOD

This program uses the Poisson Equation Finite-Difference Method to approximate the solution to the Poisson equation

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y)$$

subject to boundary conditions $u(x,y) = g(x,y)$. The sample problem uses

$$f(x,y) = xe^y \quad \text{and} \quad g(x,y) = xe^y.$$

INPUT: $a = 0, \quad b = 2, \quad c = 0, \quad d = 1, \quad n = 6, \quad m = 5,$
$TOL = 10^{-5}, \quad M = 150$

HEAT EQUATION "HEBDM122.MWS" PAGE 507

BACKWARD-DIFFERENCE METHOD

This program uses the Heat Equation Backward-Difference Method to approximate the solution to a parabolic partial-differential equation

$$\frac{\partial u}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < l, \quad 0 < t$$

subject to zero boundary conditions and the initial condition $u(x,0) = f(x)$. The sample problem uses

$$f(x) = \sin \pi x.$$

INPUT: $l = 1, \quad T = 0.5, \quad \alpha = 1, \quad m = 10, \quad N = 50$

CRANK-NICOLSON METHOD "HECNM123.MWS" PAGE 510

This program uses the Crank-Nicolson Method to approximate the solution to a parabolic partial-differential equation

$$\frac{\partial u}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < l, \quad 0 < t$$

subject to zero boundary conditions and the initial condition $u(x,0) = f(x)$. The sample problem uses

$$f(x) = \sin \pi x.$$

INPUT: $l = 1, \quad T = 0.5, \quad \alpha = 1, \quad m = 10, \quad N = 50$

WAVE EQUATION "WVFDM124.MWS" PAGE 518

FINITE-DIFFERENCE METHOD

This program uses the Wave Equation Finite-Difference Method to approximate the solution to a wave equation

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < l, \quad 0 < t$$

subject to zero boundary conditions and initial conditions

$$u(x,0) = f(x) \quad \text{and} \quad \frac{\partial u}{\partial t}(x,0) = g(x).$$

The sample problem uses

$$f(x) = 2\sin(3\pi x) \quad \text{and} \quad g(x) = -12\sin(2\pi x).$$

INPUT: $l = 1, \quad T = 1, \quad \alpha = 2, \quad m = 10, \quad N = 20$

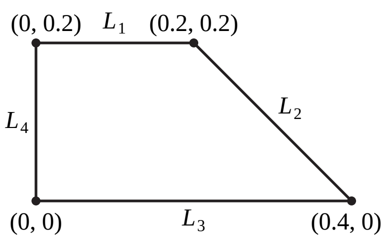
This program uses the Finite-Element Method to approximate the solution to an elliptic partial-differential equation of the form

$$\frac{\partial}{\partial x}\left(p(x,y)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right)+r(x,y)u=f(x,y)$$

subject to Dirichlet, mixed, or Neumann boundary conditions. The sample problem considers Laplace's equation

$$\frac{\partial^2 u}{\partial x^2}(x,y)+\frac{\partial^2 u}{\partial y^2}(x,y)=0$$

on the two-dimensional region shown in the figure below.



The boundary conditions on this region are

$$u(x,y)=4 \quad \text{for} \quad (x,y) \quad \text{on} \quad L_3 \quad \text{and} \quad L_4,$$
$$\frac{\partial u}{\partial n}(x,y)=x \quad \text{for} \quad (x,y) \quad \text{on} \quad L_1,$$

and

$$\frac{\partial u}{\partial n}(x,y)=\frac{x+y}{\sqrt{2}} \quad \text{for} \quad (x,y) \quad \text{on} \quad L_2.$$

INPUT: FINEL125.DTA