NAG Toolbox for MATLAB

d02ja

1 Purpose

d02ja solves a regular linear two-point boundary-value problem for a single *n*th-order ordinary differential equation by Chebyshev-series using collocation and least-squares.

2 Syntax

```
[c, ifail] = d02ja(n, cf, bc, x0, x1, k1, kp)
```

3 Description

d02ja calculates the solution of a regular two-point boundary-value problem for a single *n*th-order linear ordinary differential equation as a Chebyshev-series in the range (x_0, x_1). The differential equation

$$f_{n+1}(x)y^{(n)}(x) + f_n(x)y^{(n-1)}(x) + \cdots + f_1(x)y(x) = f_0(x)$$

is defined by the user-supplied real function **cf**, and the boundary conditions at the points x_0 and x_1 are defined by the user-supplied (sub)program **bc**.

You specify the degree of Chebyshev-series required, **k1** $-$ 1, and the number of collocation points, **kp**. The function sets up a system of linear equations for the Chebyshev coefficients, one equation for each collocation point and one for each boundary condition. The boundary conditions are solved exactly, and the remaining equations are then solved by a least-squares method. The result produced is a set of coefficients for a Chebyshev-series solution of the differential equation on a range normalized to the range $(-1, 1)$.

e02ak can be used to evaluate the solution at any point on the range (x_0, x_1) – see Section 9 for an example. e02ah followed by e02ak can be used to evaluate its derivatives.

4 References

Picken S M 1970 Algorithms for the solution of differential equations in Chebyshev-series by the selected points method *Report Math. 94* National Physical Laboratory

5 Parameters

5.1 Compulsory Input Parameters

1: **n – int32 scalar**

The order *n* of the differential equation.

Constraint: **n** ≥ 1.

2: **cf – string containing name of m-file**

cf defines the differential equation (see Section 3). It must return the value of a function $f_j(x)$ at a given point *x*, where, for $1 \leq j \leq n+1$, $f_j(x)$ is the coefficient of $y^{(j-1)}(x)$ in the equation, and $f_0(x)$ is the right-hand side.

Its specification is:

```
[result] = cf(j, x)
```

Input Parameters

1: **j – int32 scalar**

The index of the function f_j to be evaluated.

2: **x – double scalar**

The point at which f_j is to be evaluated.

Output Parameters

1: **result – double scalar**

The result of the function.

3: **bc – string containing name of m-file**

bc defines the boundary conditions, each of which has the form $y^{(k-1)}(x_1) = s_k$ or $y^{(k-1)}(x_0) = s_k$. The boundary conditions may be specified in any order.

Its specification is:

```
[j, rhs] = bc(ii)
```

Input Parameters

1: **ii – int32 scalar**

The index of the boundary condition to be defined.

Output Parameters

1: **j – int32 scalar**

Must be set to $-k$ if the boundary condition is $y^{(k-1)}(x_0) = s_k$, and to $+k$ if it is $y^{(k-1)}(x_1) = s_k$.

j must not be set to the same value k for two different values of **ii**.

2: **rhs – double scalar**

Must be set to the value s_k.

4: **x0 – double scalar**

5: **x1 – double scalar**

The left- and right-hand boundaries, x_0 and x_1, respectively.

Constraint: **x1** $>$ **x0**.

6: **k1 – int32 scalar**

the number of coefficients to be returned in the Chebyshev-series representation of the solution (hence the degree of the polynomial approximation is **k1** $- 1$).

Constraint: **k1** $\geq$ **n** $+ 1$.

7: **kp – int32 scalar**

The number of collocation points to be used.

Constraint: **kp** $\geq$ **k1** $-$ **n**.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

w, lw, iw

5.4 Output Parameters

1: **c(k1) – double array**

The computed Chebyshev coefficients; that is, the computed solution is:

$$\sum_{i=1}^{\mathbf{k1}}{}' \mathbf{c}(i) T_{i-1}(x)$$

where $T_i(x)$ is the ith Chebyshev polynomial of the first kind, and $\sum'$ denotes that the first coefficient, $\mathbf{c}(1)$, is halved.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{n} < 1$,
or $\mathbf{x0} \geq \mathbf{x1}$,
or $\mathbf{k1} < \mathbf{n} + 1$,
or $\mathbf{kp} < \mathbf{k1} - \mathbf{n}$.

ifail $= 2$

On entry, $\mathbf{lw} < 2 \times (\mathbf{kp} + \mathbf{n}) \times (\mathbf{k1} + 1) + 7 \times \mathbf{k1}$ (insufficient workspace).

ifail $= 3$

Either the boundary conditions are not linearly independent (that is, in the user-supplied (sub)program **bc** the variable j is set to the same value k for two different values of i), or the rank of the matrix of equations for the coefficients is less than the number of unknowns. Increasing **kp** may overcome this problem.

ifail $= 4$

The least-squares function f04am has failed to correct the first approximate solution (see f04am).

7 Accuracy

The Chebyshev coefficients are determined by a stable numerical method. The accuracy of the approximate solution may be checked by varying the degree of the polynomial and the number of collocation points (see Section 8).

8 Further Comments

The time taken by d02ja depends on the complexity of the differential equation, the degree of the polynomial solution, and the number of matching points.

The collocation points in the range (x_0, x_1) are chosen to be the extrema of the appropriate shifted Chebyshev polynomial. If $\mathbf{kp} = \mathbf{k1} - \mathbf{n}$, then the least-squares solution reduces to the solution of a system of linear equations, and true collocation results. The accuracy of the solution may be checked by repeating the calculation with different values of **k1** and with **kp** fixed but $\mathbf{kp} \gg \mathbf{k1} - \mathbf{n}$. If the Chebyshev coefficients decrease rapidly (and consistently for various **k1** and **kp**), the size of the last two or three gives an indication of the error. If the Chebyshev coefficients do not decay rapidly, it is likely that the solution cannot be well-represented by Chebyshev-series. Note that the Chebyshev coefficients are calculated for the range $(-1, 1)$.

Systems of regular linear differential equations can be solved using d02jb. It is necessary before using d02jb to write the differential equations as a first-order system. Linear systems of high-order equations in their original form, singular problems, and, indirectly, nonlinear problems can be solved using d02tg.

9 Example

d02ja_bc.m

```
function [jj, rhs] = bc(ii)
  rhs = 0;
  if (ii == 1)
    jj = int32(1);
  else
    jj = int32(-1);
  end
```

d02ja_cf.m

```
function result = cf(j, x)
  if (j == 2)
    result = 0;
  else
    result = 1;
  end
```

```
n = int32(2);
x0 = -1;
x1 = 1;
k1 = int32(4);
kp = int32(10);
[c, ifail] = d02ja(n, 'd02ja_cf', 'd02ja_bc', x0, x1, k1, kp)
```

```
c =
   -0.6108
   -0.0000
    0.3054
    0.0000
ifail =
           0
```