NAG Toolbox

nag_ode_bvp_ps_lin_cgl_deriv (d02ud)

1 Purpose

nag_ode_bvp_ps_lin_cgl_deriv (d02ud) differentiates a function discretized on Chebyshev Gauss–Lobatto points. The grid points on which the function values are to be provided are normally returned by a previous call to nag_ode_bvp_ps_lin_cgl_grid (d02uc).

2 Syntax

```
[fd, ifail] = nag_ode_bvp_ps_lin_cgl_deriv(n, f)
[fd, ifail] = d02ud(n, f)
```

3 Description

nag_ode_bvp_ps_lin_cgl_deriv (d02ud) differentiates a function discretized on Chebyshev Gauss–Lobatto points on $[-1, 1]$. The polynomial interpolation on Chebyshev points is equivalent to trigonometric interpolation on equally spaced points. Hence the differentiation on the Chebyshev points can be implemented by the Fast Fourier transform (FFT).

Given the function values $f(x_i)$ on Chebyshev Gauss–Lobatto points $x_i = -\cos((i-1)\pi/n)$, for $i = 1, 2, \ldots, n+1$, f is differentiated with respect to x by means of forward and backward FFTs on the function values $f(x_i)$. nag_ode_bvp_ps_lin_cgl_deriv (d02ud) returns the computed derivative values $f'(x_i)$, for $i = 1, 2, \ldots, n+1$. The derivatives are computed with respect to the standard Chebyshev Gauss–Lobatto points on $[-1, 1]$; for derivatives of a function on $[a, b]$ the returned values have to be scaled by a factor $2/(b-a)$.

4 References

Canuto C, Hussaini M Y, Quarteroni A and Zang T A (2006) *Spectral Methods: Fundamentals in Single Domains* Springer

Greengard L (1991) Spectral integration and two-point boundary value problems *SIAM J. Numer. Anal.* **28(4)** 1071–80

Trefethen L N (2000) *Spectral Methods in MATLAB* SIAM

5 Parameters

5.1 Compulsory Input Parameters

1: **n** – INTEGER

n, where the number of grid points is $n+1$.

Constraint: $\mathbf{n} > 0$ and $\mathbf{n}$ is even.

2: $\mathbf{f}(\mathbf{n}+\mathbf{1})$ – REAL (KIND=nag_wp) array

The function values $f(x_i)$, for $i = 1, 2, \ldots, n+1$

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: $\mathbf{fd}(\mathbf{n}+\mathbf{1})$ – REAL (KIND=nag_wp) array

The approximations to the derivatives of the function evaluated at the Chebyshev Gauss–Lobatto points. For functions defined on $[a,b]$, the returned derivative values (corresponding to the domain $[-1,1]$) must be multiplied by the factor $2/(b-a)$ to obtain the correct values on $[a,b]$.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

Constraint: $\mathbf{n} > 0$.

Constraint: **n** is even.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The accuracy is close to ***machine precision*** for small numbers of grid points, typically less than 100. For larger numbers of grid points, the error in differentiation grows with the number of grid points. See Greengard (1991) for more details.

8 Further Comments

The number of operations is of the order $n\log(n)$ and the memory requirements are $O(n)$; thus the computation remains efficient and practical for very fine discretizations (very large values of n).

9 Example

The function $2x + \exp(-x)$, defined on $[0, 1.5]$, is supplied and then differentiated on a grid.

9.1 Program Text

```
     function d02ud_example

fprintf('d02ud example results\n\n');

n = nag_int(16);
a = 0;
b = 1.5;

% Set up solution grid
[x, ifail] = d02uc(n, a, b);

% Evaluate function on Chebyshev grid
f = 2*x+exp(-x);
```

```
% Calculate derivative of function
[fd, ifail] = d02ud(n, f);

scale = 2/(b-a);
fd = scale*fd;

% Print function and its derivatives
fprintf('\nOriginal function F and numerical derivative Fx\n');
fprintf('      x          F          Fx\n');
fprintf('%10.4f %10.4f %10.4f\n', [x f fd]');
```

9.2 Program Results

```
     d02ud example results

Original function F and numerical derivative Fx
      x          F          Fx
    0.0000     1.0000     1.0000
    0.0144     1.0145     1.0143
    0.0571     1.0587     1.0555
    0.1264     1.1341     1.1187
    0.2197     1.2421     1.1972
    0.3333     1.3832     1.2835
    0.4630     1.5554     1.3706
    0.6037     1.7542     1.4532
    0.7500     1.9724     1.5276
    0.8963     2.2007     1.5919
    1.0370     2.4285     1.6455
    1.1667     2.6448     1.6886
    1.2803     2.8386     1.7221
    1.3736     3.0004     1.7468
    1.4429     3.1221     1.7638
    1.4856     3.1975     1.7736
    1.5000     3.2231     1.7769
```