NAG Toolbox

nag_roots_contfn_cntin (c05aw)

1 Purpose

nag_roots_contfn_cntin (c05aw) attempts to locate a zero of a continuous function using a continuation method based on a secant iteration.

2 Syntax

```
[x, user, ifail] = nag_roots_contfn_cntin(x, eps, eta, f, nfmax, 'user', user)
[x, user, ifail] = c05aw(x, eps, eta, f, nfmax, 'user', user)
```

3 Description

nag_roots_contfn_cntin (c05aw) attempts to obtain an approximation to a simple zero α of the function $f(x)$ given an initial approximation x to α. The zero is found by a call to nag_roots_contfn_cntin_rcomm (c05ax) whose specification should be consulted for details of the method used.

The approximation x to the zero α is determined so that at least one of the following criteria is satisfied:

(i) $|x - \alpha| \sim$ **eps**,

(ii) $|f(x)| <$ **eta**.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **x** – REAL (KIND=nag_wp)

An initial approximation to the zero.

2: **eps** – REAL (KIND=nag_wp)

An absolute tolerance to control the accuracy to which the zero is determined. In general, the smaller the value of **eps** the more accurate **x** will be as an approximation to α. Indeed, for very small positive values of **eps**, it is likely that the final approximation will satisfy $|\mathbf{x} - \alpha| <$ **eps**. You are advised to call the function with more than one value for **eps** to check the accuracy obtained.

Constraint: **eps** > 0.0.

3: **eta** – REAL (KIND=nag_wp)

A value such that if $|f(x)| <$ **eta**, x is accepted as the zero. **eta** may be specified as 0.0 (see Section 7).

4: **f** – REAL (KIND=nag_wp) FUNCTION, supplied by the user.

f must evaluate the function f whose zero is to be determined.

```
[result, user] = f(x, user)
```

Input Parameters

1: **x** – REAL (KIND=nag_wp)

The point at which the function must be evaluated.

2: **user** – INTEGER array

f is called from nag_roots_contfn_cntin (c05aw) with the object supplied to nag_roots_contfn_cntin (c05aw).

Output Parameters

1: **result**

The value of f evaluated at **x**.

2: **user** – INTEGER array

5: **nfmax** – INTEGER

The maximum permitted number of calls to **f** from nag_roots_contfn_cntin (c05aw). If **f** is inexpensive to evaluate, **nfmax** should be given a large value (say > 1000).

Constraint: **nfmax** > 0.

5.2 Optional Input Parameters

1: **user** – INTEGER array

user is not used by nag_roots_contfn_cntin (c05aw), but is passed to **f**. Note that for large objects it may be more efficient to use a global variable which is accessible from the m-files than to use **user**.

5.3 Output Parameters

1: **x** – REAL (KIND=nag_wp)

The final approximation to the zero, unless **ifail** $= 1$, 2 or 5, in which case it contains no useful information.

2: **user** – INTEGER array

3: **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: **eps** > 0.0.

Constraint: **nfmax** > 0.

ifail $= 2$

Internal scale factor invalid for this problem. Consider using nag_roots_contfn_cntin_rcomm (c05ax) instead and setting **scal**.

ifail $= 3$

Either **f** has no zero near **x** or too much accuracy has been requested. Check the coding of **f** or increase **eps**.

ifail $= 4$

More than **nfmax** calls have been made to **f**.

nfmax *may be too small for the problem (because* ***x*** *is too far away from the zero), or* ***f*** *has no zero near* ***x****, or too much accuracy has been requested in calculating the zero. Increase* ***nfmax****, check the coding of* ***f*** *or increase* ***eps****.*

ifail $= 5$

A serious error occurred in an internal call to an auxiliary function.

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 levels of accuracy depend on the values of **eps** and **eta**. If full machine accuracy is required, they may be set very small, resulting in an exit with **ifail** $= 3$ or 4, although this may involve many more iterations than a lesser accuracy. You are recommended to set **eta** $= 0.0$ and to use **eps** to control the accuracy, unless you have considerable knowledge of the size of $f(x)$ for values of x near the zero.

8 Further Comments

The time taken by nag_roots_contfn_cntin (c05aw) depends primarily on the time spent evaluating the function f (see Section 5) and on how close the initial value of **x** is to the zero.

If a more flexible way of specifying the function f is required or if you wish to have closer control of the calculation, then the reverse communication function nag_roots_contfn_cntin_rcomm (c05ax) is recommended instead of nag_roots_contfn_cntin (c05aw).

9 Example

This example calculates the zero of $f(x) = e^{-x} - x$ from a starting value **x** $= 1.0$. Two calculations are made with **eps** $= 1.0\text{e}{-3}$ and $1.0\text{e}{-4}$ for comparison purposes, with **eta** $= 0.0$ in both cases.

9.1 Program Text

```
function c05aw_example

fprintf('c05aw example results\n\n');

x = 1;
eta = 0;
nfmax = nag_int(200);
fprintf('\n');
% Repeat with tolerance eps set to varying powers of 10
for k=3:4
  [xOut, user, ifail] = c05aw(x, 10^-k, eta, @f, nfmax);
  switch ifail
```

```
    case {0}
      fprintf('With eps = %10.2e, root = %14.5f\n', 10^-k, xOut);
    case {3, 4}
      fprintf('With eps = %10.2e, final value = %14.5f\n', 10^-k, xOut);
    otherwise
      break;
  end
end

function [result, user] = f(x, user)
  result = x - exp(-x);
```

9.2 Program Results

```
     c05aw example results

With eps =   1.00e-03, root =        0.56715
With eps =   1.00e-04, root =        0.56715
```