

NAG Toolbox

nag_correg_optset (g02zk)

1 Purpose

nag_correg_optset (g02zk) either initializes or resets the optional parameter arrays or sets a single optional parameter for supported problem solving functions in Chapter G02. Currently, only nag_correg_quantile_linreg (g02qg) is supported.

2 Syntax

```
[iopts, opts, ifail] = nag_correg_optset(optstr, iopts, opts, 'liopts', liopts,
    'lopts', lopts)
[iopts, opts, ifail] = g02zk(optstr, iopts, opts, 'liopts', liopts, 'lopts',
    lopts)
```

3 Description

nag_correg_optset (g02zk) has three purposes: to initialize optional parameter arrays, to reset all optional parameters to their default values or to set a single optional parameter to a user-supplied value.

Optional parameters and their values are, in general, presented as a character string, **optstr**, of the form '*option = optval*'; alphabetic characters can be supplied in either upper or lower case. Both *option* and *optval* may consist of one or more tokens separated by white space. The tokens that comprise *optval* will normally be either an integer, real or character value as defined in the description of the specific optional argument. In addition all optional parameters can take an *optval* DEFAULT which resets the optional parameter to its default value.

It is imperative that optional parameter arrays are initialized before any options are set, before the relevant problem solving function is called and before any options are queried using nag_correg_optget (g02zl). To initialize the optional parameter arrays **iopts** and **opts** for a specific problem solving function, the option **Initialize** is used with *value* identifying the problem solving function to be called, via its short name. For example, to initialize optional parameter arrays to be passed to nag_correg_quantile_linreg (g02qg), nag_correg_optset (g02zk) is called as follows:

```
[iopts, opts, ifail] = g02zk('Initialize = g02qg', iopts, opts);
```

Information relating to available option names and their corresponding valid values is given in Section 12 in nag_correg_quantile_linreg (g02qg).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **optstr** – CHARACTER(*)

A string identifying the option to be set.

Initialize = *function name*

Initialize the optional parameter arrays **iopts** and **opts** for use with function *function name*, where *function name* is the short name of the problem solving function you wish to use.

Defaults

Resets all options to their default values.

option = *optval*

See Section 12 in nag_correg_quantile_linreg (g02qg) for details of valid values for *option* and *optval*. The equals sign (=) delimiter must be used to separate the *option* from its *optval* value.

optstr is case insensitive. Each token in the *option* and *optval* component must be separated by at least one space.

2: **iopts(iopts)** – INTEGER array

Optional parameter array.

If **optstr** has the form **Initialize** = *function name*, the contents of **iopts** need not be set.

Otherwise, **iopts must not** have been altered since the last call to nag_correg_optset (g02zk), nag_correg_optget (g02zl) or the selected problem solving function.

3: **opts(lopts)** – REAL (KIND=nag_wp) array

Optional parameter array.

If **optstr** has the form **Initialize** = *function name*, the contents of **opts** need not be set.

Otherwise, **opts must not** have been altered since the last call to nag_correg_optset (g02zk), nag_correg_optget (g02zl) or the selected problem solving function.

5.2 Optional Input Parameters

1: **liopts** – INTEGER

Default: the dimension of the array **iopts**.

The length of the array **iopts**.

Constraint: unless otherwise stated in the documentation for a specific, supported, problem solving function, **liopts** \geq 100.

2: **lopts** – INTEGER

Default: the dimension of the array **opts**.

The length of the array **opts**.

Constraint: unless otherwise stated in the documentation for a specific, supported, problem solving function, **lopts** \geq 100.

5.3 Output Parameters

1: **iopts(iopts)** – INTEGER array

Dependent on the contents of **optstr**, either an initialized, reset or updated version of the optional parameter array.

- 2: **opts(lopts)** – REAL (KIND=nag_wp) array
Dependent on the contents of **optstr**, either an initialized, reset or updated version of the optional parameter 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 = 11

On entry, the *option* supplied in **optstr** was not recognized.

ifail = 12

On entry, the expected delimiter '=' was not found in **optstr**.

ifail = 13

On entry, could not convert the specified *optval* to an integer.

On entry, could not convert the specified *optval* to a real.

ifail = 14

On entry, attempting to initialize the optional parameter arrays but specified function name was not valid.

ifail = 15

On entry, the *optval* supplied for the integer optional parameter is not valid.

ifail = 16

On entry, the *optval* supplied for the real optional parameter is not valid.

ifail = 17

On entry, the *optval* supplied for the character optional parameter is not valid.

ifail = 21

On entry, either the option arrays have not been initialized or they have been corrupted.

ifail = 31

liopts is too small.

ifail = 51

lopts is too small.

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

Not applicable.

8 Further Comments

Not applicable.

9 Example

See the example programs associated with the problem solving function you wish to use for a demonstration of how to use nag_correg_optset (g02zk) to initialize option arrays and set options.

9.1 Program Text

```
function g02zk_example

fprintf('g02zk example results\n\n');

sorder = nag_int(1);
c1 = 'y';
weight = 'u';
dat = [ 420.1577; 541.4117; 901.1575; 639.0802; 750.8756; 945.7989;
        829.3979; 979.1648; 1309.8789; 1492.3987; 502.8390; 616.7168;
        790.9225; 555.8786; 713.4412; 838.7561; 535.0766; 596.4408;
        924.5619; 487.7583; 692.6397; 997.8770; 506.9995; 654.1587;
        933.9193; 433.6813; 587.5962; 896.4746; 454.4782; 584.9989;
        800.7990; 502.4369; 713.5197; 906.0006; 880.5969; 796.8289;
        854.8791; 1167.3716; 523.8000; 670.7792; 377.0584; 851.5430;
        1121.0937; 625.5179; 805.5377; 558.5812; 884.4005; 1257.4989;
        2051.1789; 1466.3330; 730.0989; 2432.3910; 940.9218; 1177.8547;
        1222.5939; 1519.5811; 687.6638; 953.1192; 953.1192; 953.1192;
        939.0418; 1283.4025; 1511.5789; 1342.5821; 511.7980; 689.7988;
        1532.3074; 1056.0808; 387.3195; 387.3195; 410.9987; 499.7510;
        832.7554; 614.9986; 887.4658; 1595.1611; 1807.9520; 541.2006;
        1057.6767; 800.7990; 1245.6964; 1201.0002; 634.4002; 956.2315;
        1148.6010; 1768.8236; 2822.5330; 922.3548; 2293.1920; 627.4726;
        889.9809; 1162.2000; 1197.0794; 530.7972; 1142.1526; 1088.0039;
        484.6612; 1536.0201; 678.8974; 671.8802; 690.4683; 860.6948;
        873.3095; 894.4598; 1148.6470; 926.8762; 839.0414; 829.4974;
        1264.0043; 1937.9771; 698.8317; 920.4199; 1897.5711; 891.6824;
        889.6784; 1221.4818; 544.5991; 1031.4491; 1462.9497; 830.4353;
        975.0415; 1337.9983; 867.6427; 725.7459; 989.0056; 1525.0005;
        672.1960; 923.3977; 472.3215; 590.7601; 831.7983; 1139.4945;
        507.5169; 576.1972; 696.5991; 650.8180; 949.5802; 497.1193;
        570.1674; 724.7306; 408.3399; 638.6713; 1225.7890; 715.3701;
        800.4708; 975.5974; 1613.7565; 608.5019; 958.6634; 835.9426;
        1024.8177; 1006.4353; 726.0000; 494.4174; 776.5958; 415.4407;
        581.3599; 643.3571; 2551.6615; 1795.3226; 1165.7734; 815.6212;
        1264.2066; 1095.4056; 447.4479; 1178.9742; 975.8023; 1017.8522;
        423.8798; 558.7767; 943.2487; 1348.3002; 2340.6174; 587.1792;
        1540.9741; 1115.8481; 1044.6843; 1389.7929; 2497.7860; 1585.3809;
        1862.0438; 2008.8546; 697.3099; 571.2517; 598.3465; 461.0977;
        977.1107; 883.9849; 718.3594; 543.8971; 1587.3480; 4957.8130;
        969.6838; 419.9980; 561.9990; 689.5988; 1398.5203; 820.8168;
        875.1716; 1392.4499; 1256.3174; 1362.8590; 1999.2552; 1209.4730;
        1125.0356; 1827.4010; 1014.1540; 880.3944; 873.7375; 951.4432;
        473.0022; 601.0030; 713.9979; 829.2984; 959.7953; 1212.9613;
        958.8743; 1129.4431; 1943.0419; 539.6388; 463.5990; 562.6400;
        736.7584; 1415.4461; 2208.7897; 636.0009; 759.4010; 1078.8382;
        748.6413; 987.6417; 788.0961; 1020.0225; 1230.9235; 440.5174;
        743.0772];
y = [ 255.8394; 310.9587; 485.6800; 402.9974; 495.5608; 633.7978;
```

```

630.7566; 700.4409; 830.9586; 815.3602; 338.0014; 412.3613;
520.0006; 452.4015; 512.7201; 658.8395; 392.5995; 443.5586;
640.1164; 333.8394; 466.9583; 543.3969; 317.7198; 424.3209;
518.9617; 338.0014; 419.6412; 476.3200; 386.3602; 423.2783;
503.3572; 354.6389; 497.3182; 588.5195; 654.5971; 550.7274;
528.3770; 640.4813; 401.3204; 435.9990; 276.5606; 588.3488;
664.1978; 444.8602; 462.8995; 377.7792; 553.1504; 810.8962;
1067.9541; 1049.8788; 522.7012; 1424.8047; 517.9196; 830.9586;
925.5795; 1162.0024; 383.4580; 621.1173; 621.1173; 621.1173;
548.6002; 745.2353; 837.8005; 795.3402; 418.5976; 508.7974;
883.2780; 742.5276; 242.3202; 242.3202; 266.0010; 408.4992;
614.7588; 385.3184; 515.6200; 1138.1620; 993.9630; 299.1993;
750.3202; 572.0807; 907.3969; 811.5776; 427.7975; 649.9985;
860.6002; 1143.4211; 2032.6792; 590.6183; 1570.3911; 483.4800;
600.4804; 696.2021; 774.7962; 390.5984; 612.5619; 708.7622;
296.9192; 1071.4627; 496.5976; 503.3974; 357.6411; 430.3376;
624.6990; 582.5413; 580.2215; 543.8807; 588.6372; 627.9999;
712.1012; 968.3949; 482.5816; 593.1694; 1033.5658; 693.6795;
693.6795; 761.2791; 361.3981; 628.4522; 771.4486; 757.1187;
821.5970; 1022.3202; 679.4407; 538.7491; 679.9981; 977.0033;
561.2015; 728.3997; 372.3186; 361.5210; 620.8006; 819.9964;
360.8780; 395.7608; 442.0001; 404.0384; 670.7993; 297.5702;
353.4882; 383.9376; 284.8008; 431.1000; 801.3518; 448.4513;
577.9111; 570.5210; 865.3205; 444.5578; 680.4198; 576.2779;
708.4787; 734.2356; 433.0010; 327.4188; 485.5198; 305.4390;
468.0008; 459.8177; 863.9199; 831.4407; 534.7610; 392.0502;
934.9752; 813.3081; 263.7100; 769.0838; 630.5863; 645.9874;
319.5584; 348.4518; 614.5068; 662.0096; 1504.3708; 406.2180;
692.1689; 588.1371; 511.2609; 700.5600; 1301.1451; 879.0660;
912.8851; 1509.7812; 484.0605; 399.6703; 444.1001; 248.8101;
527.8014; 500.6313; 436.8107; 374.7990; 726.3921; 1827.2000;
523.4911; 334.9998; 473.2009; 581.2029; 929.7540; 591.1974;
637.5483; 674.9509; 776.7589; 959.5170; 1250.9643; 737.8201;
810.6772; 983.0009; 708.8968; 633.1200; 631.7982; 608.6419;
300.9999; 377.9984; 397.0015; 588.5195; 681.7616; 807.3603;
696.8011; 811.1962; 1305.7201; 442.0001; 353.6013; 468.0008;
526.7573; 890.2390; 1318.8033; 331.0005; 416.4015; 596.8406;
429.0399; 619.6408; 400.7990; 775.0209; 772.7611; 306.5191;
522.6019];
isx = [nag_int(1)];
tau = [0.10; 0.25; 0.50; 0.75; 0.90];
state = zeros(1, 1, nag_int_name);
ip = 2;
b = zeros(2, 5);
iopts = zeros(100, 1, nag_int_name);
opts = zeros(100, 1);
% Initialize the optional argument array
[iopts, opts, ifail] = g02zk( ...
    'Initialize = g02qg', iopts, opts);

% Set optional arguments
[iopts, opts, ifail] = g02zk( ...
    'Return Residuals = Yes', iopts, opts);
[iopts, opts, ifail] = g02zk( ...
    'Matrix Returned = Covariance', iopts, opts);
[iopts, opts, ifail] = g02zk( ...
    'Interval Method = IID', iopts, opts);

% Call the model fitting routine
[df, b, bl, bu, ch, res, state, info, ifail] = ...
    g02qg( ...
        sorder, c1, weight, dat, isx, y, tau, b, iopts, opts, state);

if (ifail == 0)
    % Display the parameter estimates
    for l=1:numel(tau)
        fprintf('\nQuantile: %6.3f\n\n', tau(l));
        fprintf('    Lower    Parameter    Upper\n');
        fprintf('    Limit    Estimate    Limit\n');
        for j=1:2
            fprintf('%3d    %7.3f    %7.3f    %7.3f\n', j, bl(j,l), b(j,l), bu(j,l));
        end
    end
end

```

```

end
fprintf('\nCovariance matrix\n');
for i=1:ip
    fprintf('%10.3e ', ch(1:i, i, 1));
    fprintf('\n');
end
fprintf('\n');
end

if (numel(res) > 0)
    fprintf('First 10 Residuals\n');
    fprintf('
    Quantile\n');
    fprintf('Obs.   %6.3f   %6.3f   %6.3f   %6.3f   %6.3f\n', tau);
    for i=1:10
        fprintf(' %3d %10.5f %10.5f %10.5f %10.5f %10.5f\n', i, res(i, 1:5));
    end
else
    fprintf('Residuals not returned\n');
end
elseif (ifail == 231)
    fprintf('\nAdditional error information (info):\n');
    disp(info);
end

```

9.2 Program Results

g02zk example results

Quantile: 0.100

	Lower Limit	Parameter Estimate	Upper Limit
1	74.946	110.142	145.337
2	0.370	0.402	0.433

Covariance matrix

```

3.191e+02
-2.541e-01  2.587e-04

```

Quantile: 0.250

	Lower Limit	Parameter Estimate	Upper Limit
1	64.232	95.483	126.735
2	0.446	0.474	0.502

Covariance matrix

```

2.516e+02
-2.004e-01  2.039e-04

```

Quantile: 0.500

	Lower Limit	Parameter Estimate	Upper Limit
1	55.399	81.482	107.566
2	0.537	0.560	0.584

Covariance matrix

```

1.753e+02
-1.396e-01  1.421e-04

```

Quantile: 0.750

	Lower Limit	Parameter Estimate	Upper Limit
1	41.372	62.396	83.421
2	0.625	0.644	0.663

Covariance matrix

```

1.139e+02

```

-9.068e-02 9.230e-05

Quantile: 0.900

	Lower Limit	Parameter Estimate	Upper Limit
1	26.829	67.351	107.873
2	0.650	0.686	0.723

Covariance matrix

4.230e+02
-3.369e-01 3.429e-04

First 10 Residuals

Obs.	Quantile				
	0.100	0.250	0.500	0.750	0.900
1	-23.10718	-38.84219	-61.00711	-77.14462	-99.86551
2	-16.70358	-41.20981	-73.81193	-100.11463	-127.96277
3	13.48419	-37.04518	-100.61322	-157.07478	-200.13481
4	36.09526	4.52393	-36.48522	-70.97584	-102.95390
5	83.74310	44.08476	-6.54743	-50.41028	-87.11562
6	143.66660	89.90799	22.49734	-37.70668	-82.65437
7	187.39134	142.05288	84.66171	34.21603	-5.80963
8	196.90443	140.73220	70.44951	7.44831	-38.91027
9	194.55254	114.45726	15.70761	-75.01861	-135.36147
10	105.62394	12.32563	-102.13482	-208.16238	-276.22311
