1 Purpose

nag_tsa_spectrum_univar_cov (g13cac) calculates the smoothed sample spectrum of a univariate time series using one of four lag windows – rectangular, Bartlett, Tukey or Parzen window.

2 Specification

```c
#include <nag.h>
#include <naggl3.h>

void nag_tsa_spectrum_univar_cov (Integer nx, Integer mtx, double px,
         Integer iw, Integer mw, Integer ic, Integer nc, double c[], Integer kc,
         Integer l, Nag_LoggedSpectra lg_spect, Integer nxg, double xg[],
         Integer *ng, double stats[], NagError *fail)
```

3 Description

The smoothed sample spectrum is defined as

\[
\hat{f}(\omega) = \frac{1}{2\pi} \left( C_0 + 2 \sum_{k=1}^{M-1} w_k C_k \cos(\omega k) \right),
\]

where \( M \) is the window width, and is calculated for frequency values

\[ \omega_i = \frac{2\pi i}{L}, \quad i = 0, 1, \ldots, [L/2], \]

where \([\ ]\) denotes the integer part.

The autocovariances \( C_k \) may be supplied by you, or constructed from a time series \( x_1, x_2, \ldots, x_n \), as

\[ C_k = \frac{1}{n-k} \sum_{t=1}^{n-k} x_t x_{t+k}, \]

the fast Fourier transform (FFT) being used to carry out the convolution in this formula.

The time series may be mean or trend corrected (by classical least squares), and tapered before calculation of the covariances, the tapering factors being those of the split cosine bell:

\[
\frac{1}{2}(1 - \cos(\pi(t - \frac{1}{2})/T)), \quad 1 \leq t \leq T \\
\frac{1}{2}(1 - \cos(\pi(n - t + \frac{1}{2})/T)), \quad n + 1 - T \leq t \leq n \\
1, \quad \text{otherwise},
\]

where \( T = \left\lfloor \frac{np}{2} \right\rfloor \) and \( p \) is the tapering proportion.

The smoothing window is defined by

\[ w_k = W\left( \frac{k}{M} \right), \quad k \leq M - 1, \]

which for the various windows is defined over \( 0 \leq \alpha < 1 \) by rectangular:

\[ W(\alpha) = 1 \]
Bartlett:

\[ W(\alpha) = 1 - \alpha \]

Tukey:

\[ W(\alpha) = \frac{1}{2}(1 + \cos(\pi \alpha)) \]

Parzen:

\[ W(\alpha) = 1 - 6\alpha^2 + 6\alpha^3, \quad 0 \leq \alpha \leq \frac{1}{2} \]

\[ W(\alpha) = 2(1 - \alpha)^3, \quad \frac{1}{2} < \alpha < 1. \]

The sampling distribution of \( \hat{f}(\omega) \) is approximately that of a scaled \( \chi^2_d \) variate, whose degrees of freedom \( d \) is provided by the function, together with multiplying limits \( mu, ml \) from which approximate 95% confidence intervals for the true spectrum \( f(\omega) \) may be constructed as \[ [ml \times \hat{f}(\omega), mu \times \hat{f}(\omega)]. \]

Alternatively, log \( \hat{f}(\omega) \) may be returned, with additive limits.

The bandwidth \( b \) of the corresponding smoothing window in the frequency domain is also provided. Spectrum estimates separated by (angular) frequencies much greater than \( b \) may be assumed to be independent.

4 References


5 Arguments

1: \( nx \) – Integer

*Input*

*On entry:* \( n \), the length of the time series.

*Constraint:* \( nx \geq 1 \).

2: \( mtx \) – Integer

*Input*

*On entry:* if covariances are to be calculated by the function (\( ic = 0 \)), \( mtx \) must specify whether the data are to be initially mean or trend corrected.

\( mtx = 0 \)

For no correction.

\( mtx = 1 \)

For mean correction.

\( mtx = 2 \)

For trend correction.

*Constraint:* if \( ic = 0 \), \( 0 \leq mtx \leq 2 \)

If covariances are supplied (\( ic \neq 0 \)), \( mtx \) is not used.

3: \( px \) – double

*Input*

*On entry:* if covariances are to be calculated by the function (\( ic = 0 \)), \( px \) must specify the proportion of the data (totalled over both ends) to be initially tapered by the split cosine bell taper.

If covariances are supplied (\( ic \neq 0 \)), \( px \) must specify the proportion of data tapered before the supplied covariances were calculated and after any mean or trend correction. \( px \) is required for the calculation of output statistics. A value of 0.0 implies no tapering.

*Constraint:* \( 0.0 \leq px \leq 1.0 \).
4:  iw – Integer  
   On entry: the choice of lag window.
   iw = 1  
      Rectangular.
   iw = 2  
      Bartlett.
   iw = 3  
      Tukey.
   iw = 4  
      Parzen.
   Constraint: 1 ≤ iw ≤ 4.

5:  mw – Integer  
   On entry: M, the ‘cut-off’ point of the lag window. Windowed covariances at lag M or greater are zero.
   Constraint: 1 ≤ mw ≤ nx.

6:  ic – Integer  
   On entry: indicates whether covariances are to be calculated in the function or supplied in the call to the function.
   ic = 0  
      Covariances are to be calculated.
   ic ≠ 0  
      Covariances are to be supplied.

7:  nc – Integer  
   On entry: the number of covariances to be calculated in the function or supplied in the call to the function.
   Constraint: mw ≤ nc ≤ nx.

8:  c[nc] – double  
   On entry: if ic ≠ 0, c must contain the nc covariances for lags from 0 to (nc – 1), otherwise c need not be set.
   On exit: if ic = 0, c will contain the nc calculated covariances.
   If ic ≠ 0, the contents of c will be unchanged.

9:  kc – Integer  
   On entry: if ic = 0, kc must specify the order of the fast Fourier transform (FFT) used to calculate the covariances. kc should be a product of small primes such as \(2^m\) where \(m\) is the smallest integer such that \(2^m ≥ nx + nc\), provided \(m ≤ 20\).
   If ic ≠ 0, that is covariances are supplied, kc is not used.
   Constraint: kc ≥ nx + nc. The largest prime factor of kc must not exceed 19, and the total number of prime factors of kc, counting repetitions, must not exceed 20. These two restrictions are imposed by the internal FFT algorithm used.

10: l – Integer  
    On entry: L, the frequency division of the spectral estimates as \(L\). Therefore it is also the order of the FFT used to construct the sample spectrum from the covariances. l should be a product of
small primes such as $2^m$ where $m$ is the smallest integer such that $2^m \geq 2M - 1$, provided $m \leq 20$.

Constraint: $l \geq 2 \times mw - 1$. The largest prime factor of $l$ must not exceed 19, and the total number of prime factors of $l$, counting repetitions, must not exceed 20. These two restrictions are imposed by the internal FFT algorithm used.

11: $lg\_spect$ – Nag\_LoggedSpectra    
Input

On entry: indicates whether unlogged or logged spectral estimates and confidence limits are required.

$lg\_spect =$ Nag\_Unlogged
Unlogged.

$lg\_spect =$ Nag\_Logged
Logged.

Constraint: $lg\_spect$ = Nag\_Unlogged or Nag\_Logged.

12: $nxg$ – Integer    
Input

On entry: the dimension of the array $xg$.

Constraints:
if $ic = 0$, $nxg \geq \max(\text{kc}, l)$;
if $ic \neq 0$, $nxg \geq 1$.

13: $xg[\text{nxg}]$ – double    
Input/Output

On entry: if the covariances are to be calculated, then $xg$ must contain the $nx$ data points. If covariances are supplied, $xg$ may contain any values.

On exit: contains the $ng$ spectral estimates, $\hat{f}(\omega_i)$, for $i = 0, 1, \ldots, [L/2]$ in $xg[0]$ to $xg[ng - 1]$ respectively (logged if $lg\_spect = $ Nag\_Logged). The elements $xg[i - 1]$, for $i = ng + 1, \ldots, nxg$ contain 0.0.

14: $ng$ – Integer *    
Output

On exit: the number of spectral estimates, $[L/2] + 1$, in $xg$.

15: $\text{stats}[4]$ – double    
Output

On exit: four associated statistics. These are the degrees of freedom in $\text{stats}[0]$, the lower and upper 95% confidence limit factors in $\text{stats}[1]$ and $\text{stats}[2]$ respectively (logged if $lg\_spect = $ Nag\_Logged), and the bandwidth in $\text{stats}[3]$.

16: $fail$ – NagError *    
Input/Output

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE\_ALLOC\_FAIL
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE\_BAD\_PARAM
On entry, argument <value> had an illegal value.
NE_CONFID_LIMITS

The calculation of confidence limit factors has failed.

NE_INT

On entry, \( ic = 0 \) and \( mtx < 0 \): \( mtx = \langle value \rangle \).
On entry, \( ic = 0 \) and \( mtx > 2 \): \( mtx = \langle value \rangle \).
On entry, \( iw = \langle value \rangle \).
Constraint: \( iw = 1, 2, 3 \) or \( 4 \).
On entry, \( mw = \langle value \rangle \).
Constraint: \( mw \geq 1 \).
On entry, \( nx = \langle value \rangle \).
Constraint: \( nx \geq 1 \).

NE_INT_2

On entry, \( l = \langle value \rangle \) and \( mw = \langle value \rangle \).
Constraint: \( l \geq 2 \times mw - 1 \).
On entry, \( mw = \langle value \rangle \) and \( nx = \langle value \rangle \).
Constraint: \( mw \leq nx \).
On entry, \( nc = \langle value \rangle \) and \( mw = \langle value \rangle \).
Constraint: \( nc \geq mw \).
On entry, \( nc = \langle value \rangle \) and \( nx = \langle value \rangle \).
Constraint: \( nc \leq nx \).
On entry, \( nxg = \langle value \rangle \) and \( l = \langle value \rangle \).
Constraint: if \( ic \neq 0 \), \( nxg \geq l \).

NE_INT_3

On entry, \( kc = \langle value \rangle \), \( nx = \langle value \rangle \) and \( nc = \langle value \rangle \).
Constraint: if \( ic = 0 \), \( kc \geq (nx + nc) \).
On entry, \( nxg = \langle value \rangle \), \( kc = \langle value \rangle \) and \( l = \langle value \rangle \).
Constraint: if \( ic = 0 \), \( nxg \geq \max(kc, l) \).

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG. See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly. See Section 3.6.5 in the Essential Introduction for further information.

NE_PRIME_FACTOR

\( kc \) has a prime factor exceeding 19, or more than 20 prime factors (counting repetitions):
\( kc = \langle value \rangle \).
\( l \) has a prime factor exceeding 19, or more than 20 prime factors (counting repetitions):
\( l = \langle value \rangle \).
NE_REAL

On entry, px = \langle\text{value}\rangle.
Constraint: px ≤ 1.0.

On entry, px = \langle\text{value}\rangle.
Constraint: px ≥ 0.0.

NE_SPECTRAL_ESTIMATES

One or more spectral estimates are zero. Consult the values in xg and stats.

7 Accuracy

The FFT is a numerically stable process, and any errors introduced during the computation will normally be insignificant compared with uncertainty in the data.

8 Parallelism and Performance

Not applicable.

9 Further Comments

nag_tsa_spectrum_univar_cov (g13cac) carries out two FFTs of length kc to calculate the covariances and one FFT of length l to calculate the sample spectrum. The time taken by the function for an FFT of length n is approximately proportional to nlog(n) (but see Section 9 in nag_sum_fft_realherm_1d (c06pac) for further details).

10 Example

This example reads a time series of length 256. It selects the mean correction option, a tapering proportion of 0.1, the Parzen smoothing window and a cut-off point for the window at lag 100. It chooses to have 100 auto-covariances calculated and unlogged spectral estimates at a frequency division of 2π/200. It then calls nag_tsa_spectrum_univar_cov (g13cac) to calculate the univariate spectrum and statistics and prints the autocovariances and the spectrum together with its 95% confidence multiplying limits.

10.1 Program Text

/* nag_tsa_spectrum_univar_cov (g13cac) Example Program. */
* Copyright 2014 Numerical Algorithms Group.
* * Mark 7, 2002.
* * Mark 7b revised, 2004.
* */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg13.h>

int main(void)
{
    /* Scalars */
    double px;
    Integer exit_status, i, ic, iw, kc, lf, mtx, mw, nc, ng, nx, nxg;
    NagError fail;

    /* Arrays */
    double *c = 0, *xg = 0;
    double stats[4];
INIT_FAIL(fail);
exit_status = 0;

printf(
"nag_tsa_spectrum_univar_cov (g13cac) Example Program Results\n"
);

/* Skip heading in data file */
#ifdef _WIN32
scanf_s("%*[\n ] ");
#else
scanf("%*[\n ] ");
#endif

#ifdef _WIN32
scanf_s(""NAG_IFMT""%"NAG_IFMT""%*[\n ] ", &nx, &nc);
#else
scanf(""NAG_IFMT""%"NAG_IFMT""%*[\n ] ", &nx, &nc);
#endif

if (nx > 0 && nc > 0)
{
  mtx = 1;
p[, = 0.1;
iw = 4;
mw = 100;
ic = 0;
k", = 360;
lf = 200;

  if (ic == 0)
    nxg = MAX(k", lf);
  else
    nxg =", l;

  /* Allocate memory */
  if (!(c = NAG_ALLOC(nc, double)) ||
    !(xg = NAG_ALLOC(nxg, double))
  
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  for (i = 1; i <= nx; ++i)
  #ifdef _WIN32
    scanf_s("%lf", &xg[i-1]);
  #else
    scanf("%lf", &xg[i-1]);
  #endif
  #ifdef _WIN32
    scanf_s("%*[\n ] ");
  #else
    scanf("%*[\n ] ");
  #endif

  /* nag_tsa_spectrum_univar_cov (g13cac).
   * Univariate time series, smoothed sample spectrum using
   * rectangular, Bartlett, Tukey or Parzen lag window
   */
  nag_tsa_spectrum_univar_cov(nx, mtx, px, iw, mw, ic, nc, c, kc, lf,
                       Nag_Unlogged, nxg, xg, &ng, stats, &fail);
  if (fail.code != NE_NOERROR)
    
    printf( 
      "Error from nag_tsa_spectrum_univar_cov (g13cac).\n", fail.message);
    exit_status = 1;
    goto END;
  }
}
printf("\n");
printf("Covariances\n");
for (i = 1; i <= nc; ++i)
{
    printf("%11.4f", c[i-1]);
    if (i % 6 == 0 || i == nc)
        printf("\n");
}
printf("\n");
printf("Degrees of freedom =%4.1f Bandwidth =%7.4f\n",
        stats[0], stats[3]);
printf("\n");
printf("95 percent confidence limits - Lower =%7.4f "
        "Upper =%7.4f\n", stats[1], stats[2]);
printf("\n");
printf(" Spectrum Spectrum Spectrum\n"                Spectrum\n");
printf(" estimate estimate estimate\n"                estimate\n");
for (i = 1; i <= ng; ++i)
{
    printf("%4"NAG_IFMT"%10.4f", i, xg[i-1]);
    if (i % 4 == 0 || i == ng)
        printf("\n");
}

END:
NAG_FREE(c);
NAG_FREE(xg);

return exit_status;

10.2 Program Data

nag_tsa_spectrum_univar_cov (g13cac) Example Program Data
256 100
5.0 11.0 16.0 23.0 36.0 58.0 29.0 20.0 10.0 8.0 3.0 0.0
0.0 2.0 11.0 27.0 47.0 63.0 60.0 39.0 28.0 26.0 22.0 11.0
21.0 40.0 78.0 122.0 103.0 73.0 47.0 35.0 11.0 5.0 16.0 34.0
70.0 81.0 111.0 101.0 73.0 40.0 20.0 16.0 5.0 11.0 22.0 40.0
60.0 80.9 83.4 47.7 47.8 30.7 12.2 9.6 10.2 32.4 47.6 54.0
62.9 85.9 61.2 45.1 36.4 20.9 11.4 37.8 69.8 106.1 100.8 81.6
66.5 34.8 30.6 7.0 19.8 92.5 154.4 84.8 68.1 38.5 22.8
10.2 24.1 82.9 132.0 130.9 118.1 89.9 66.6 60.0 46.9 41.0 21.3
16.0 6.4 41.0 6.8 14.5 34.0 45.0 43.1 47.5 42.5 28.1 10.1
8.1 2.5 0.0 1.4 5.0 12.2 13.9 35.4 45.8 41.1 30.1 23.9
15.6 6.6 4.0 1.8 8.5 16.6 36.3 49.6 64.2 67.0 70.9 47.8
27.5 8.5 13.2 56.9 121.5 138.3 103.2 85.7 64.6 36.7 24.2 10.7
15.0 40.1 61.5 98.5 124.7 96.3 66.6 64.5 54.1 39.0 20.6 6.7
4.3 22.7 54.8 93.8 95.8 77.2 59.1 44.0 47.0 30.5 16.3 7.3
37.6 74.0 139.0 111.2 101.6 66.2 44.7 17.0 11.3 12.4 3.4 6.0
32.3 54.3 59.7 63.7 63.5 52.2 25.4 13.1 6.8 6.3 7.1 35.6
73.0 85.1 78.0 64.0 41.8 26.2 26.7 12.1 9.5 2.7 5.0 24.4
42.0 63.5 53.8 62.0 48.5 43.9 18.6 5.7 3.6 1.4 9.6 47.4
57.1 103.9 80.6 63.6 37.6 26.1 14.2 5.8 16.7 44.3 63.9 69.0
77.8 64.9 35.7 21.2 11.1 5.7 8.7 36.1 79.7 114.4 109.6 88.8
67.8 47.5 30.6 16.3 9.6 33.2 92.6 151.6 136.3 134.7 83.9 69.4
31.5 13.9 4.4 38.0
### 10.3 Program Results

**nag_tsa_spectrum_univar_cov (g13cac) Example Program Results**

**Covariances**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1152.9733</td>
<td>937.3289</td>
<td>494.9243</td>
<td>14.8648</td>
<td>-342.8548</td>
</tr>
<tr>
<td></td>
<td>-469.2733</td>
<td>-236.6896</td>
<td>109.0608</td>
<td>441.3498</td>
<td>637.4571</td>
</tr>
<tr>
<td></td>
<td>454.0505</td>
<td>154.5960</td>
<td>-136.8016</td>
<td>-343.3911</td>
<td>-421.8441</td>
</tr>
<tr>
<td></td>
<td>-241.1943</td>
<td>-55.6140</td>
<td>129.4067</td>
<td>267.4248</td>
<td>311.8293</td>
</tr>
<tr>
<td></td>
<td>56.4402</td>
<td>-146.4689</td>
<td>-320.9948</td>
<td>-406.4077</td>
<td>-375.6384</td>
</tr>
<tr>
<td></td>
<td>-132.6214</td>
<td>11.0791</td>
<td>126.4843</td>
<td>171.3391</td>
<td>122.6284</td>
</tr>
<tr>
<td></td>
<td>-3.4126</td>
<td>73.2521</td>
<td>98.0831</td>
<td>71.8949</td>
<td>17.0985</td>
</tr>
<tr>
<td></td>
<td>-76.7900</td>
<td>-110.5354</td>
<td>-126.1383</td>
<td>-121.1043</td>
<td>-103.9362</td>
</tr>
<tr>
<td></td>
<td>-10.8678</td>
<td>58.5009</td>
<td>116.4587</td>
<td>140.0961</td>
<td>129.5928</td>
</tr>
<tr>
<td></td>
<td>-35.5487</td>
<td>-135.3894</td>
<td>-203.7149</td>
<td>-216.2161</td>
<td>-152.7723</td>
</tr>
<tr>
<td></td>
<td>99.3397</td>
<td>188.9594</td>
<td>204.9047</td>
<td>148.4056</td>
<td>34.4975</td>
</tr>
<tr>
<td></td>
<td>-208.5982</td>
<td>-252.4128</td>
<td>-223.7600</td>
<td>-120.8640</td>
<td>23.3565</td>
</tr>
<tr>
<td></td>
<td>227.7642</td>
<td>228.5123</td>
<td>172.3820</td>
<td>87.4911</td>
<td>-21.2370</td>
</tr>
<tr>
<td></td>
<td>-176.3634</td>
<td>-165.1218</td>
<td>-75.1308</td>
<td>67.1634</td>
<td>195.7290</td>
</tr>
<tr>
<td></td>
<td>290.8258</td>
<td>225.3811</td>
<td>104.0784</td>
<td>-44.4731</td>
<td>-162.7355</td>
</tr>
<tr>
<td></td>
<td>-165.2444</td>
<td>-48.5473</td>
<td>118.8872</td>
<td>265.0045</td>
<td></td>
</tr>
</tbody>
</table>

**Degrees of freedom = 9.0**  
**Bandwidth = 0.1165**

**95 percent confidence limits - Lower = 0.4731 Upper = 3.3329**

<table>
<thead>
<tr>
<th>Spectrum</th>
<th>Spectrum</th>
<th>Spectrum</th>
<th>Spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>estimate</td>
<td>estimate</td>
<td>estimate</td>
<td>estimate</td>
</tr>
<tr>
<td>1</td>
<td>210.4696</td>
<td>2</td>
<td>428.2020</td>
</tr>
<tr>
<td>2</td>
<td>706.1605</td>
<td>6</td>
<td>393.4052</td>
</tr>
<tr>
<td>3</td>
<td>170.1320</td>
<td>10</td>
<td>133.0442</td>
</tr>
<tr>
<td>4</td>
<td>141.5173</td>
<td>14</td>
<td>194.3041</td>
</tr>
<tr>
<td>5</td>
<td>985.3130</td>
<td>18</td>
<td>2023.1574</td>
</tr>
<tr>
<td>6</td>
<td>1669.9001</td>
<td>22</td>
<td>1012.1320</td>
</tr>
<tr>
<td>7</td>
<td>441.9977</td>
<td>26</td>
<td>300.1985</td>
</tr>
<tr>
<td>8</td>
<td>79.1533</td>
<td>30</td>
<td>49.4882</td>
</tr>
<tr>
<td>9</td>
<td>27.5111</td>
<td>34</td>
<td>59.4429</td>
</tr>
<tr>
<td>10</td>
<td>116.6737</td>
<td>38</td>
<td>87.3142</td>
</tr>
<tr>
<td>11</td>
<td>46.6097</td>
<td>42</td>
<td>53.6206</td>
</tr>
<tr>
<td>12</td>
<td>25.6285</td>
<td>46</td>
<td>24.8555</td>
</tr>
<tr>
<td>13</td>
<td>27.3351</td>
<td>50</td>
<td>22.4443</td>
</tr>
<tr>
<td>14</td>
<td>12.0207</td>
<td>54</td>
<td>12.6846</td>
</tr>
<tr>
<td>15</td>
<td>12.6103</td>
<td>58</td>
<td>7.9511</td>
</tr>
<tr>
<td>16</td>
<td>3.4182</td>
<td>62</td>
<td>3.2359</td>
</tr>
<tr>
<td>17</td>
<td>10.0610</td>
<td>66</td>
<td>7.9483</td>
</tr>
<tr>
<td>18</td>
<td>5.5751</td>
<td>70</td>
<td>7.8491</td>
</tr>
<tr>
<td>19</td>
<td>10.1386</td>
<td>74</td>
<td>6.3158</td>
</tr>
<tr>
<td>20</td>
<td>1.8026</td>
<td>78</td>
<td>1.0103</td>
</tr>
<tr>
<td>21</td>
<td>4.0822</td>
<td>82</td>
<td>4.6221</td>
</tr>
<tr>
<td>22</td>
<td>4.8498</td>
<td>86</td>
<td>6.3964</td>
</tr>
<tr>
<td>23</td>
<td>4.4444</td>
<td>90</td>
<td>5.2131</td>
</tr>
<tr>
<td>24</td>
<td>5.8722</td>
<td>94</td>
<td>7.9268</td>
</tr>
<tr>
<td>25</td>
<td>4.5495</td>
<td>98</td>
<td>5.2696</td>
</tr>
</tbody>
</table>

**Mark 25 g13cac.9 (last)**