DOCUMENTATION OF STATION/AGENCY MAGNITUDE PROCEDURES

(Modified from the SUMMARY OF IASPEI MAGNITUDE WORKING GROUP RECOMMENDATIONS ON DETERMINING EARTHQUAKE MAGNITUDES FROM DIGITAL DATA, updated version 2011; see http://www.iaspei.org/commissions/CSOI/Summary_WG-Recommendations_20110909.pdf)

This document is to outline the procedures adopted by seismological agencies to compute magnitudes of seismic events.

Agency Name: WEL - GNS Science/GeoNet

Please list the magnitudes computed and corresponding phase type analyzed in the table below (example provided). Add as many rows as required.

Magnitude type (nomenclature used at the agency)	**Full name**	**Wave type analyzed**
mB	Broad band body wave magnitude.	P-waves
Mw(mB)	Estimation of the moment magnitude Mw based on mB using the Mw vs. mB regression of Bormann and Saul (2008). *Bormann, P. and Saul, J., 2008. The New IASPEI Standard Broadband Magnitude mB. Seism. Res. Lett., 79, 699-705. IASPEI New Manual of Seismological Observatory Practice (NMSOP) Vol. 1 & 2, 2002. Ed. P. Bormann, GeoForschungZentrum Potsdam, Potsdam.*	P-waves
MLv	Local magnitude calculated on the vertical component using a correction term to fit with the standard ML. See: http://www.seiscomp3.org/doc/jakarta/current/apps/global_mlv.html *Richter, C.F., 1935. An instrumental earthquake magnitude scale. Bull. Seism. Soc. Am., 25, 1-32.*	Regional S waves
Summary Magnitude M	Summary magnitude: magnitude which is a weighted sum of MLv and Mw(mB) See:http://www.seiscomp3.org/doc/jakarta/current/apps/scmag.html	

For each magnitude type computed at the agency, please specify:

1. The equations that are used for calculating each magnitude type and
   a: specify if distance is measured as epicentral distance or hypocentral distance;
   b: specify the distance range for which the equation is applied;
   c: specify restrictions on hypocentral focal-depth, if any.

MLv:

The MLv amplitude calculation is very similar to the original ML, except that it is measured on the vertical component. The individual station MLv is calculated using the following formula:

```
mag = \log10(A) - \log10(A0)
```

A is the MLv Wood-Anderson amplitude in millimetres. The second term is the empirical calibration function, which in turn is a function of the epicentral distance (see Richter, 1935).
The second term is the empirical calibration function, which in turn is a function of the epicentral distance (see Richter, 1935). This calibration function is configured globally using the default:

(Distance-Value pairs) -> "0 -1.3; 60 -2.8; 400 -4.5;1000 -5.85"

The logA0 configuration string consists of an arbitrary number of distance-value pairs. The distance is in km and the value corresponds to the log10(A0) term above. Within each interval the values are computed by linear interpolation. E.g. for the above default specification, at a distance of 100 km the logA0 value would be:

((-4.5)-(-2.8))*(100-60)/(400-60)-2.8 = -3.0

in other words, at 100 km distance the magnitude would be

mag = \log10(A) - (-3) = \log10(A) + 3

which is according to the original Richter (1935) formula if the amplitude is measured in millimeters. Note that the baseline for logA0 is millimetres for historical reasons, while internally in SeisComP 3 the Wood-Anderson amplitudes are measured and stored micrometres.

a - MLv is computed for epicentral distances
b - MLv is computed from 0 to 1000 km
c - No restrictions on focal depth

mB:

mB is a magnitude based on body waves, but the amplitude is measured in a broad frequency range and longer time windows. Instead of amplitude measurements on displacement data together with the dominant period, the maximum velocity amplitude Vmax is taken directly from velocity-proportional records with $V = 2\pi A/T$. The time

window for the measurement can be determined by the duration of the high-frequency (1-3 Hz) radiation (Bormann & Saul, 2008). This time window usually contains the phases P, pP, sP, PcP, but not PP. According to the long time window and broad frequency range used for amplitude measurements mB saturates not like mb. In SeisComP3 a static time window of 60 s is actually taken for amplitude measurements at stations in the distance range of 5° to 105°.

a – mB is computed for epicentral distances
b - Mlv is computed from 5° to 105° distances
c - No restrictions on focal depth

Equation used:

mB = log10(A/T) + Q
where
A is the zero-to-peak amplitude in microns of the P phase
(not necessarily limited to the first few cycles)
T is period in seconds
Q is a function of distance (degrees) and depth (kilometers)

Q function – see embedded Annex

2. Is any signal-to-noise ratio criterion applied to the analyzed signal?
   MLv: No SNR set (submitted to future change)
   mB: SNR=3 (default)

3. Specify the software used (such as SeisComp, Antelope, Seismic Handler, Seisan, SAC, in-house developed programs) to perform the analyses for magnitude computation.
   SeisComp

4. If the agency is computing magnitudes not based on some amplitude/period measurement (e.g., moment magnitude Mw) please summarize the details of the technique used. For example, is Mw obtained with a centroid moment tensor, W-phase and/or spectral fitting technique?
   GeoNet routinely calculates regional moment tensor solutions based on the time-domain moment tensor routine of Doug Dreger at Berkeley Seismological Laboratory (e.g. Dreger and Helmberger, 1993; Dreger 2003). The procedure as adapted for New Zealand is described in Ristau (2008) and Ristau (2013). The moment tensor solutions are calculated manually with the epicenter fixed, but the inversion carried out over a range of depths to find the best depth. As of 17 February 2015 the catalogue consists of 1639 moment tensor solutions dating from 21 August 2003.

   Estimation of the moment magnitude Mw(mB) based on mB using the Mw vs. mB regression of Bormann and Saul (2008).
   Bormann, P. and Saul, J., 2008. The New IASPEI Standard Broadband Magnitude mB. Seism. Res. Lett., 79, 699-705. IASPEI New Manual of Seismological Observatory

Practice (NMSOP) Vol. 1 & 2, 2002. Ed. P. Bormann, GeoForschungZentrum Potsdam, Potsdam.
Dreger, D.S. (2003). TDMT_INV: Time domain seismic moment tensor INVersion, in International Handbook of Earthquake and Engineering Seismology, W.K. Lee (Editor), Vol. 81B, Academic Press, Boston, 1627 pp.
Dreger, D., and D.V. Helmberger (1993). Determination of source parameters at regional distances with single station or sparse network data, J. Geophys. Res. 98, 8107-8125.
Pasyanos, M.E., D.E. Dreger, and B. Romanowicz (1996). Towards real-time determination of regional moment tensor solutions, Bull. Seism. Soc. Amer. 86, 1255-1269.
Ristau, J. (2008). Implementation of routine regional moment tensor analysis in New Zealand, Seism. Res. Lett. 79, 400-415, doi: 10.1785/gssrl.79.3.400.
Ristau, J. (2013). Update of regional moment tensor analysis for earthquakes in New Zealand and adjacent offshore regions, Bull. Seism. Soc. Amer. 103, 2520-2533, doi: 10.1785/0120120339.

5. Other restrictions on the calculation of a specific magnitude. For example, is the magnitude measured only for earthquakes of a certain size, as defined by an independent measure of earthquake size? Also, are specific magnitudes computed only for seismic events occurring in specific areas?

a. A weighted (summary) magnitude is used combining MLv and Mw(mB).
# Define the coefficients to calculate the weight
# of a magnitude:
# weight = a*magStationCount+b
# Unnamed values define the default values
summaryMagnitude.coefficients.a = MLv:0, Mw(mB):0.4
summaryMagnitude.coefficients.b = MLv:2, Mw(mB):-1

Define the coefficients to calculate the weight of the contribution of a given magnitude to the summary magnitude. The name before the colon must match a contributing magnitude type.
If a magnitude measurement exists then:

weight = a* NetworkMagnitude.stationCount + b.

The Summary magnitude is then calculated as:

$$M = \frac{\sum w_i M_i}{\sum w_i}$$

$$w_i = a_i stacount(M_i) + b_i$$

b. The moment tensor solutions are typically calculated for events with $M_L \geq 4$; however, this varies depending on location and depth. Deeper events (e.g. > 33 km) often need to have $M_L \geq 4.5$, while shallow events with a good signal-to-noise ratio can have moment tensor solutions calculated for $M_L \geq 3.5$. Offshore earthquakes with source-receiver distances > ~400 – 500 km typically need to have $M_L \geq 4.5$.

Detailed questions on the magnitudes based on amplitude/period measurements:

6. How the network (event) magnitude and corresponding uncertainty is obtained? For example, is the network magnitude an arithmetic/trimmed mean, median value of the single station magnitudes?
The network magnitude is computed automatically from a Trimmed mean (removing the Magnitude exceeding 25%)

7. Units of the reported amplitudes. Specify if amplitudes are reported in units of trace-amplitude motion instead of ground motion.
MLv amplitudes values are stored in mm/s
mB amplitudes values are stored in nm/s

8. Time-window in which the amplitude measurement is made for the phase type analyzed. For example, for body wave magnitudes, is the time window a flexible time-interval between the P onset and the PP onset or a fixed time window after the first P onset (e.g. 5 s, 10 s or other)? Similarly, for the surface wave magnitudes, is the time window considered a time-interval spanned by waves having group-velocities between, e.g., 3.2 and 4.0 km/s or is always the maximum velocity amplitude, respectively (A/T)max in the whole surface-wave train in a wide range of periods be measured? If so, give the range of periods.
-Fixed time windows (Automatic procedures)
-mB is within 60 seconds after pick time
-MLv is within 150 seconds after pick time
-Time windows can be shortened and amplitudes can be revised manually by duty officers

9. Orientation of seismograph (horizontal or vertical) from which the measurement is made. For example, is Ms computed using both horizontal and vertical components? Specify also if, as for example might be the case for ML, data from each of the two horizontal components at a single station are used, are data from each component treated as a separate observation in the network magnitude computation, or are the two components first averaged into a station magnitude, which is then treated as a single observation in the network magnitude computation?
MLv and mB are created from the vertical component streams

10. Describe the amplitude-response, filter characteristics, or transfer-function of the seismograph or simulated seismograph through which the amplitude measurement is made. For example, is the IASPEI recommended standard Wood-Anderson seismometer simulation filter with the parameters according to Uhrhammer and Collins (1990) used to compute ML?
mB: broadband measurement on velocity streams
MLv: BandPass of 0.1 s and 3 s .Wood Anderson Filter is automatically applying a norm factor of 2800. Therefore the resulting amplitudes are in the range of micrometre per seconds.

11. Details of measuring amplitude:

a: For example, does the amplitude correspond to 0.5*(peak-to-trough amplitude), where "peak-to-trough amplitude" corresponds to difference between a maximum positive excursion and a maximum negative excursion of the trace, or is the amplitude instead measured as the maximum absolute excursion from the "zero" position of the seismograph trace?
MLv: Stored as Max Absolute value x 2
mB: Stored as Max Absolute x 1
Time of the amplitudes is the one of that measurement

b: for example, if the amplitude corresponds to 0.5*(peak-to-trough amplitude), are the "peak" and "trough" respectively the absolute maximum and absolute minimum values of the entire wave-group, or are they the adjacent peak and trough corresponding to the maximum trace excursion that is associated with a single zero-crossing?

c: for example, are displacement amplitude(A) and period(T) measured at the time of maximum A or at the time of the maximum of the quotient (A/T)?

12. Details of measuring period. For example, is it the time between the neighboring peaks, respectively troughs or twice the time span measured between the largest peak and adjacent trough at which the double amplitude has been measured?

13. To what part of a phase the amplitude-measurement time refers. For example, is the amplitude-measurement time the time of the zero-crossing associated with a peak-to-adjacent trough measurement or is it the time of an absolute maximum or absolute minimum?
Time of an absolute maximum

Finally, please add publications as well as internal reports or web links that can be quoted to describe the magnitude procedures adopted at the agency and/or any other relevant information which may not have been included in the questions above.

##########################Annex#####################################

MB computation- Q function

```
************************************************************************
 *   Copyright (C) by GFZ Potsdam
*
 *
*
 *   You can redistribute and/or modify this program under the
*
 *   terms of the SeisComP Public License.
*
 *
*
 *   This program is distributed in the hope that it will be useful,
*
 *   but WITHOUT ANY WARRANTY; without even the implied warranty of
*
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
*
 *   SeisComP Public License for more details.
*

************************************************************************
/



#include <math.h>
#include <seiscomp3/seismology/magnitudes.h>

/* See http://jclahr.com/science/software/magnitude/mb/index.html
 *
```

```
 * These are the original Gutenberg and Richter (1956) calibration
functions
 * as they are still (2007) used at the NEIC
 *
 * The Q values for a distance of 5 degrees are in the 4th column, i.e.
 * at __qmb[x][3]
 */

/*
static float __qmb[17][108] = {
    { 5.6, 5.8, 6.1, 6.4, 6.5, 7.0, 7.0, 7.2, 7.3, 7.2, 7.1, 7.0, 6.6, 6.3,
5.9, 5.9, 5.9, 6.0, 6.1, 6.1, 6.2, 6.3, 6.4, 6.5, 6.5, 6.5, 6.6, 6.6, 6.6,
6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.5, 6.5, 6.4, 6.4, 6.5, 6.5, 6.5, 6.6, 6.7,
6.8, 6.9, 6.9, 6.8, 6.7, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.8, 6.8, 6.9, 6.9,
6.9, 7.0, 7.0, 7.0, 7.0, 7.0, 7.0, 7.0, 7.0, 6.9, 6.9, 6.9, 6.9, 6.8, 6.8,
6.9, 6.9, 6.9, 6.8, 6.7, 6.8, 6.9, 7.0, 7.0, 7.0, 6.9, 7.0, 7.1, 7.0, 7.0,
7.1, 7.1, 7.2, 7.1, 7.2, 7.3, 7.4, 7.5, 7.5, 7.3, 7.4, 7.4, 7.5, 7.6, 7.7,
7.8, 7.8, 7.9, 8.0 },
    { 0.0, 0.0, 0.0, 6.3, 6.5, 6.8, 7.0, 7.0, 7.1, 7.0, 7.0, 6.9, 6.5, 6.1,
5.9, 5.9, 5.9, 6.0, 6.1, 6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.4, 6.5, 6.5, 6.6,
6.6, 6.7, 6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.7,
6.7, 6.8, 6.8, 6.8, 6.8, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.8, 6.8, 6.9, 6.9,
6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.8, 6.8, 6.8, 6.8,
6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.9, 7.0, 7.0, 7.0, 7.0, 7.1, 7.1, 7.0,
7.1, 7.2, 7.2, 7.2, 7.2, 7.2, 7.3, 7.3, 7.3, 7.3, 7.3, 7.4, 7.5, 7.6, 7.7,
7.8, 7.8, 7.9, 8.0 },
    { 0.0, 0.0, 0.0, 6.0, 6.2, 6.5, 6.6, 6.8, 6.9, 6.9, 6.8, 6.7, 6.5, 6.0,
5.9, 5.9, 5.9, 6.0, 6.1, 6.1, 6.2, 6.2, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.5,
6.5, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.5, 6.5, 6.5, 6.5, 6.6,
6.7, 6.7, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.9, 6.9, 6.9,
6.9, 6.9, 6.9, 6.8, 6.8, 6.8, 6.8, 6.8, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7,
6.7, 6.8, 6.8, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.9, 7.0, 7.0, 7.0, 7.1, 7.1,
7.2, 7.2, 7.2, 7.2, 7.2, 7.3, 7.3, 7.3, 7.3, 7.3, 7.4, 7.4, 7.5, 7.6, 7.7,
7.8, 7.9, 7.9, 8.0 },
    { 0.0, 0.0, 0.0, 6.0, 6.1, 6.4, 6.5, 6.6, 6.8, 6.7, 6.7, 6.5, 6.1, 6.0,
5.9, 6.0, 6.0, 6.0, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.5,
6.5, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.6, 6.6, 6.6,
6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.9, 6.9, 6.9, 6.9,
6.9, 6.9, 6.8, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7,
6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.9, 6.9, 7.0, 7.0,
7.1, 7.2, 7.2, 7.2, 7.2, 7.3, 7.3, 7.3, 7.3, 7.4, 7.4, 7.5, 7.5, 7.7, 7.8,
7.8, 7.9, 7.9, 8.0 },
    { 0.0, 0.0, 0.0, 6.0, 6.0, 6.3, 6.4, 6.6, 6.7, 6.6, 6.5, 6.4, 6.0, 6.0,
6.0, 6.0, 6.0, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.5,
6.5, 6.5, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6,
6.6, 6.6, 6.6, 6.7, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.8, 6.9, 6.9,
6.8, 6.8, 6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6,
6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.7, 6.7, 6.7, 6.8, 6.8, 6.8, 6.9, 6.9,
7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.3, 7.3, 7.4, 7.4, 7.4, 7.5, 7.6, 7.7, 7.8,
7.8, 7.9, 7.9, 8.0 },
    { 0.0, 0.0, 0.0, 6.0, 6.0, 6.2, 6.4, 6.4, 6.4, 6.4, 6.3, 6.2, 6.1, 6.1,
6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.5, 6.5, 6.5,
6.5, 6.6, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.4, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7,
6.7, 6.7, 6.7, 6.7, 6.7, 6.7, 6.6, 6.6, 6.6, 6.6, 6.6, 6.5, 6.5, 6.5, 6.5,
6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.6, 6.6, 6.7, 6.8, 6.8, 6.8,
6.9, 6.9, 7.0, 7.0, 7.1, 7.2, 7.2, 7.3, 7.3, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8,
7.9, 7.9, 8.0, 8.0 },
    { 0.0, 0.0, 0.0, 5.9, 6.0, 6.0, 6.0, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2,
6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5,
6.5, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2,
```

```
6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5,
6.5, 6.6, 6.6, 6.6, 6.6, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5,
6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.6, 6.7, 6.7,
6.8, 6.8, 6.9, 6.9, 7.0, 7.1, 7.1, 7.2, 7.2, 7.2, 7.3, 7.4, 7.5, 7.7, 7.8,
7.9, 7.9, 8.0, 8.0 },
    { 0.0, 0.0, 0.0, 5.7, 5.7, 5.8, 5.8, 5.8, 5.9, 5.9, 6.0, 6.0, 6.1, 6.2,
6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1,
6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5,
6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.6, 6.6, 6.6, 6.7, 6.7,
6.7, 6.8, 6.8, 6.9, 7.0, 7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.3, 7.4, 7.6, 7.8,
7.9, 8.0, 8.0, 8.0 },
    { 0.0, 0.0, 0.0, 5.6, 5.7, 5.7, 5.8, 5.8, 5.9, 6.0, 6.0, 6.1, 6.1, 6.1,
6.2, 6.2, 6.2, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2, 6.1, 6.1, 6.0, 6.0, 6.0, 6.1, 6.1,
6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3,
6.3, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5,
6.5, 6.5, 6.5, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7,
6.7, 6.7, 6.8, 6.9, 6.9, 7.0, 7.0, 7.1, 7.2, 7.2, 7.3, 7.3, 7.4, 7.6, 7.8,
7.9, 8.0, 8.0, 8.0 },
    { 0.0, 0.0, 0.0, 5.7, 5.7, 5.8, 5.9, 6.0, 6.1, 6.2, 6.2, 6.2, 6.2, 6.1,
6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1,
6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0,
6.0, 6.0, 6.0, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.3,
6.3, 6.3, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5,
6.5, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6, 6.7, 6.7, 6.7, 6.7, 6.7,
6.7, 6.8, 6.8, 6.9, 6.9, 7.0, 7.0, 7.1, 7.1, 7.2, 7.3, 7.3, 7.4, 7.6, 7.8,
7.9, 8.0, 8.0, 8.0 },
    { 0.0, 0.0, 0.0, 5.7, 5.8, 5.9, 6.0, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2,
6.2, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1,
6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1,
6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2,
6.3, 6.3, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.5,
6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.6, 6.6, 6.6, 6.6,
6.7, 6.7, 6.8, 6.9, 6.9, 6.9, 7.0, 7.1, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7,
7.8, 7.9, 8.0, 8.0 },
    { 0.0, 0.0, 0.0, 5.8, 5.9, 5.9, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2,
6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2,
6.2, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3,
6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.5, 6.5, 6.6, 6.6, 6.7,
6.7, 6.8, 6.8, 6.9, 6.9, 7.0, 7.0, 7.1, 7.2, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7,
7.8, 7.8, 7.8, 7.9 },
    { 0.0, 0.0, 0.0, 5.8, 5.8, 5.9, 6.0, 6.1, 6.2, 6.2, 6.2, 6.3, 6.3, 6.4,
6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.3, 6.2, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2,
6.2, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.4, 6.4, 6.5, 6.5, 6.6, 6.6, 6.7,
6.7, 6.8, 6.8, 6.9, 6.9, 7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7,
7.7, 7.7, 7.8, 7.8 },
    { 0.0, 0.0, 0.0, 5.8, 5.8, 5.9, 5.9, 6.0, 6.1, 6.1, 6.2, 6.2, 6.2, 6.3,
6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.2, 6.2, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.1, 6.2,
6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2,
6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.4, 6.4, 6.5, 6.5, 6.6, 6.7, 6.7, 6.8, 6.8,
6.9, 6.9, 7.0, 7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.4, 7.5, 7.6, 7.6, 7.7, 7.7,
7.7, 7.7, 7.7 },
```

```
    { 0.0, 0.0, 0.0, 5.7, 5.8, 5.8, 5.8, 5.9, 5.8, 5.9, 6.0, 6.0, 6.1, 6.1,
6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1, 6.0, 6.0, 6.0, 6.0,
6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.4, 6.4, 6.5, 6.5, 6.6, 6.7, 6.7,
6.8, 6.8, 6.8, 7.0, 7.0, 7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.3, 7.4, 7.5, 7.6,
7.6, 7.6, 7.7, 7.7 },
    { 0.0, 0.0, 0.0, 5.7, 5.7, 5.7, 5.8, 5.8, 5.8, 5.8, 5.8, 5.9, 5.9, 5.9,
6.0, 6.1, 6.1, 6.1, 6.2, 6.2, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4,
6.4, 6.4, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2, 6.2, 6.2, 6.2, 6.2,
6.2, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1, 6.1, 6.1, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0,
6.0, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.5, 6.5, 6.5, 6.6, 6.7,
6.7, 6.8, 6.9, 6.9, 7.0, 7.0, 7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.3, 7.4, 7.5,
7.5, 7.5, 7.6, 7.6 },
    { 0.0, 0.0, 0.0, 5.7, 5.7, 5.7, 5.7, 5.7, 5.7, 5.8, 5.8, 5.8, 5.8, 5.8,
5.9, 5.9, 6.0, 6.0, 6.0, 6.0, 6.1, 6.1, 6.1, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3,
6.3, 6.4, 6.4, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.2, 6.2,
6.2, 6.2, 6.2, 6.2, 6.1, 6.1, 6.1, 6.1, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0, 6.0,
6.0, 6.0, 6.0, 6.1, 6.1, 6.1, 6.1, 6.2, 6.2, 6.2, 6.2, 6.2, 6.3, 6.3, 6.3,
6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.5, 6.5,
6.6, 6.7, 6.7, 6.8, 6.9, 6.9, 7.0, 7.0, 7.0, 7.1, 7.1, 7.2, 7.2, 7.3, 7.4,
7.4, 7.4, 7.5, 7.5 }
};

*/

/* Preliminary, unpublished Q function of Saul & Bormann (2007) */
static float __qmb[17][108] = {
    { // 0 km
0.0, 0.0, 0.0, 5.75, 5.88, 5.96, 6.01, 6.10, 6.14, 6.17, 6.17, 6.16, 6.15,
6.15, 6.10, 6.06, 6.05, 6.04, 6.05, 6.08, 6.11, 6.16, 6.23, 6.30, 6.39,
6.48, 6.55, 6.61, 6.64, 6.66, 6.67, 6.67, 6.67, 6.67, 6.66, 6.66, 6.65,
6.66, 6.66, 6.67, 6.67, 6.69, 6.69, 6.70, 6.71, 6.72, 6.72, 6.73, 6.74,
6.75, 6.76, 6.77, 6.77, 6.78, 6.79, 6.79, 6.80, 6.81, 6.82, 6.82, 6.83,
6.83, 6.84, 6.84, 6.84, 6.85, 6.85, 6.85, 6.86, 6.87, 6.87, 6.87, 6.87,
6.88, 6.88, 6.88, 6.89, 6.89, 6.90, 6.89, 6.89, 6.89, 6.90, 6.92, 6.94,
6.97, 7.01, 7.05, 7.09, 7.15, 7.20, 7.25, 7.31, 7.37, 7.42, 7.46, 7.50,
7.55, 7.59, 7.64, 7.68, 7.73, 7.75, 7.78, 7.78, 7.78, 7.78, 7.78
    },
    { // 25 km
0.0, 0.0, 0.0, 5.77, 5.85, 5.92, 5.99, 6.09, 6.16, 6.21, 6.23, 6.22, 6.20,
6.19, 6.15, 6.12, 6.10, 6.11, 6.12, 6.14, 6.17, 6.21, 6.26, 6.33, 6.41,
6.50, 6.57, 6.63, 6.65, 6.68, 6.69, 6.69, 6.69, 6.69, 6.69, 6.69, 6.69,
6.69, 6.68, 6.68, 6.68, 6.68, 6.69, 6.70, 6.71, 6.72, 6.74, 6.74, 6.74,
6.75, 6.75, 6.76, 6.77, 6.77, 6.78, 6.78, 6.78, 6.78, 6.79, 6.79, 6.80,
6.81, 6.81, 6.81, 6.82, 6.82, 6.83, 6.83, 6.83, 6.83, 6.83, 6.83, 6.83,
6.84, 6.84, 6.84, 6.85, 6.85, 6.85, 6.86, 6.86, 6.86, 6.86, 6.88, 6.90,
6.94, 6.98, 7.02, 7.06, 7.12, 7.16, 7.21, 7.26, 7.31, 7.36, 7.41, 7.45,
7.50, 7.54, 7.58, 7.62, 7.65, 7.67, 7.68, 7.68, 7.68, 7.68, 7.68
    },
    { // 50 km
0.0, 0.0, 0.0, 5.70, 5.86, 5.92, 5.96, 6.03, 6.10, 6.12, 6.15, 6.17, 6.16,
6.14, 6.13, 6.10, 6.08, 6.10, 6.12, 6.14, 6.16, 6.21, 6.24, 6.32, 6.40,
6.49, 6.54, 6.60, 6.63, 6.66, 6.69, 6.70, 6.69, 6.68, 6.68, 6.65, 6.65,
6.66, 6.66, 6.65, 6.66, 6.65, 6.65, 6.66, 6.68, 6.69, 6.71, 6.73, 6.74,
6.75, 6.76, 6.75, 6.74, 6.74, 6.75, 6.75, 6.76, 6.77, 6.77, 6.76, 6.76,
6.76, 6.76, 6.76, 6.78, 6.77, 6.79, 6.78, 6.78, 6.78, 6.79, 6.78, 6.78,
6.79, 6.79, 6.78, 6.78, 6.78, 6.78, 6.79, 6.79, 6.79, 6.80, 6.81, 6.83,
```

```
6.87, 6.92, 6.97, 7.02, 7.07, 7.12, 7.16, 7.21, 7.27, 7.33, 7.38, 7.41,
7.45, 7.49, 7.51, 7.54, 7.56, 7.58, 7.60, 7.60, 7.60, 7.60, 7.60
    },
    { // 75 km
0.0, 0.0, 0.0, 5.73, 5.80, 5.82, 5.88, 6.01, 6.10, 6.14, 6.15, 6.14, 6.14,
6.09, 6.04, 6.03, 6.03, 6.03, 6.07, 6.11, 6.13, 6.18, 6.22, 6.27, 6.37,
6.46, 6.53, 6.59, 6.62, 6.64, 6.66, 6.67, 6.67, 6.66, 6.65, 6.63, 6.64,
6.63, 6.63, 6.64, 6.63, 6.63, 6.64, 6.65, 6.66, 6.66, 6.67, 6.69, 6.72,
6.74, 6.75, 6.74, 6.74, 6.72, 6.72, 6.73, 6.72, 6.72, 6.73, 6.72, 6.73,
6.73, 6.74, 6.74, 6.74, 6.73, 6.74, 6.74, 6.74, 6.74, 6.75, 6.75, 6.76,
6.77, 6.78, 6.78, 6.77, 6.76, 6.76, 6.76, 6.74, 6.75, 6.77, 6.79, 6.82,
6.87, 6.93, 6.99, 7.03, 7.08, 7.13, 7.17, 7.21, 7.28, 7.34, 7.39, 7.41,
7.47, 7.53, 7.57, 7.62, 7.63, 7.64, 7.64, 7.64, 7.64, 7.64, 7.64
    },
    { // 100 km
0.0, 0.0, 0.0, 5.67, 5.73, 5.73, 5.83, 5.96, 6.02, 6.10, 6.20, 6.18, 6.15,
6.10, 6.05, 6.03, 6.01, 6.03, 6.05, 6.09, 6.10, 6.14, 6.19, 6.25, 6.33,
6.43, 6.50, 6.56, 6.59, 6.61, 6.61, 6.61, 6.60, 6.59, 6.57, 6.57, 6.57,
6.57, 6.56, 6.59, 6.58, 6.59, 6.59, 6.60, 6.59, 6.60, 6.60, 6.60, 6.61,
6.63, 6.65, 6.66, 6.67, 6.67, 6.67, 6.67, 6.68, 6.68, 6.68, 6.66, 6.66,
6.66, 6.66, 6.66, 6.66, 6.66, 6.65, 6.65, 6.66, 6.66, 6.68, 6.68, 6.69,
6.69, 6.69, 6.70, 6.70, 6.70, 6.71, 6.71, 6.71, 6.72, 6.73, 6.75, 6.77,
6.82, 6.86, 6.92, 6.98, 7.03, 7.07, 7.12, 7.17, 7.22, 7.27, 7.32, 7.34,
7.38, 7.42, 7.47, 7.51, 7.54, 7.54, 7.56, 7.56, 7.56, 7.56, 7.56
    },
    { // 150 km
0.0, 0.0, 0.0, 5.64, 5.68, 5.70, 5.78, 5.90, 5.98, 6.03, 6.08, 6.03, 5.94,
5.88, 5.91, 5.92, 5.95, 6.00, 6.03, 6.04, 6.06, 6.09, 6.12, 6.18, 6.24,
6.33, 6.41, 6.47, 6.52, 6.51, 6.51, 6.49, 6.48, 6.45, 6.45, 6.45, 6.46,
6.46, 6.46, 6.47, 6.46, 6.46, 6.46, 6.47, 6.47, 6.50, 6.50, 6.52, 6.51,
6.52, 6.53, 6.53, 6.52, 6.54, 6.55, 6.55, 6.56, 6.56, 6.56, 6.55, 6.56,
6.56, 6.57, 6.56, 6.57, 6.57, 6.56, 6.56, 6.57, 6.57, 6.58, 6.59, 6.59,
6.59, 6.60, 6.60, 6.61, 6.62, 6.63, 6.63, 6.64, 6.65, 6.66, 6.68, 6.71,
6.75, 6.78, 6.84, 6.89, 6.94, 6.99, 7.04, 7.10, 7.14, 7.20, 7.27, 7.30,
7.33, 7.38, 7.42, 7.46, 7.49, 7.51, 7.53, 7.53, 7.53, 7.53, 7.53
    },
    { // 200 km
0.0, 0.0, 0.0, 5.49, 5.50, 5.56, 5.68, 5.76, 5.86, 5.85, 5.84, 5.73, 5.70,
5.68, 5.75, 5.77, 5.84, 5.89, 5.96, 5.94, 5.97, 6.01, 6.07, 6.13, 6.22,
6.29, 6.34, 6.38, 6.42, 6.41, 6.41, 6.40, 6.38, 6.35, 6.35, 6.36, 6.36,
6.37, 6.38, 6.39, 6.38, 6.37, 6.35, 6.34, 6.35, 6.37, 6.38, 6.41, 6.40,
6.41, 6.42, 6.42, 6.42, 6.44, 6.44, 6.42, 6.44, 6.44, 6.44, 6.46, 6.48,
6.47, 6.47, 6.45, 6.43, 6.43, 6.42, 6.43, 6.46, 6.47, 6.47, 6.49, 6.50,
6.51, 6.52, 6.53, 6.53, 6.54, 6.55, 6.54, 6.55, 6.55, 6.56, 6.58, 6.61,
6.64, 6.69, 6.73, 6.76, 6.81, 6.85, 6.89, 6.95, 7.03, 7.08, 7.17, 7.21,
7.22, 7.32, 7.35, 7.42, 7.45, 7.47, 7.47, 7.47, 7.47, 7.47, 7.47
    },
    { // 250 km
0.0, 0.0, 0.0, 5.49, 5.49, 5.57, 5.73, 5.82, 5.90, 5.85, 5.76, 5.66, 5.66,
5.64, 5.73, 5.76, 5.87, 5.88, 5.95, 5.94, 5.98, 6.02, 6.09, 6.15, 6.22,
6.26, 6.30, 6.35, 6.38, 6.38, 6.37, 6.35, 6.31, 6.28, 6.28, 6.29, 6.28,
6.30, 6.30, 6.28, 6.28, 6.28, 6.28, 6.30, 6.32, 6.32, 6.33, 6.34, 6.31,
6.33, 6.33, 6.35, 6.34, 6.38, 6.37, 6.38, 6.39, 6.38, 6.38, 6.40, 6.43,
6.43, 6.46, 6.43, 6.41, 6.39, 6.39, 6.39, 6.41, 6.44, 6.44, 6.45, 6.46,
6.47, 6.47, 6.50, 6.51, 6.53, 6.54, 6.56, 6.56, 6.56, 6.57, 6.59, 6.60,
6.63, 6.68, 6.73, 6.75, 6.80, 6.84, 6.89, 6.95, 7.03, 7.09, 7.17, 7.24,
7.27, 7.31, 7.35, 7.42, 7.43, 7.44, 7.47, 7.47, 7.47, 7.47, 7.47
    },
    { // 300 km
0.0, 0.0, 0.0, 5.52, 5.47, 5.55, 5.68, 5.76, 5.82, 5.84, 5.77, 5.70, 5.72,
5.72, 5.77, 5.77, 5.83, 5.84, 5.87, 5.88, 5.93, 5.99, 6.09, 6.20, 6.24,
```

```
6.27, 6.30, 6.31, 6.32, 6.32, 6.30, 6.28, 6.24, 6.23, 6.21, 6.21, 6.21,
6.23, 6.22, 6.22, 6.23, 6.22, 6.23, 6.24, 6.24, 6.25, 6.25, 6.23, 6.21,
6.24, 6.25, 6.28, 6.30, 6.34, 6.31, 6.34, 6.33, 6.31, 6.32, 6.35, 6.36,
6.38, 6.42, 6.41, 6.42, 6.42, 6.40, 6.39, 6.41, 6.43, 6.41, 6.42, 6.44,
6.44, 6.45, 6.48, 6.50, 6.52, 6.53, 6.54, 6.53, 6.53, 6.52, 6.55, 6.57,
6.61, 6.66, 6.70, 6.72, 6.76, 6.80, 6.84, 6.90, 6.97, 7.03, 7.11, 7.18,
7.22, 7.27, 7.30, 7.36, 7.38, 7.41, 7.42, 7.42, 7.42, 7.42, 7.42
    },
    { // 350 km
0.0, 0.0, 0.0, 5.58, 5.51, 5.55, 5.64, 5.74, 5.75, 5.77, 5.74, 5.72, 5.75,
5.78, 5.79, 5.79, 5.82, 5.82, 5.83, 5.87, 5.93, 5.99, 6.06, 6.15, 6.16,
6.18, 6.23, 6.27, 6.25, 6.27, 6.25, 6.21, 6.19, 6.21, 6.20, 6.20, 6.19,
6.18, 6.16, 6.17, 6.15, 6.16, 6.19, 6.20, 6.18, 6.21, 6.20, 6.20, 6.17,
6.20, 6.20, 6.24, 6.25, 6.31, 6.29, 6.32, 6.31, 6.29, 6.29, 6.30, 6.30,
6.35, 6.38, 6.38, 6.42, 6.42, 6.39, 6.40, 6.40, 6.40, 6.38, 6.37, 6.38,
6.38, 6.40, 6.43, 6.46, 6.49, 6.50, 6.52, 6.51, 6.51, 6.52, 6.56, 6.58,
6.61, 6.66, 6.71, 6.73, 6.77, 6.81, 6.85, 6.91, 6.97, 7.03, 7.09, 7.17,
7.22, 7.27, 7.30, 7.35, 7.36, 7.38, 7.38, 7.38, 7.38, 7.38, 7.38
    },
    { // 400 km
0.0, 0.0, 0.0, 5.44, 5.44, 5.46, 5.51, 5.61, 5.61, 5.71, 5.72, 5.74, 5.76,
5.79, 5.78, 5.78, 5.78, 5.80, 5.81, 5.86, 5.88, 5.99, 6.05, 6.15, 6.15,
6.16, 6.18, 6.21, 6.18, 6.19, 6.19, 6.17, 6.14, 6.17, 6.15, 6.14, 6.14,
6.14, 6.12, 6.14, 6.13, 6.13, 6.16, 6.18, 6.15, 6.16, 6.16, 6.17, 6.14,
6.18, 6.18, 6.21, 6.22, 6.28, 6.27, 6.28, 6.28, 6.26, 6.27, 6.26, 6.27,
6.30, 6.32, 6.32, 6.38, 6.38, 6.36, 6.38, 6.38, 6.37, 6.36, 6.37, 6.37,
6.39, 6.40, 6.42, 6.44, 6.46, 6.46, 6.47, 6.48, 6.49, 6.49, 6.53, 6.57,
6.62, 6.66, 6.71, 6.74, 6.77, 6.81, 6.86, 6.92, 6.99, 7.06, 7.12, 7.20,
7.24, 7.29, 7.32, 7.36, 7.37, 7.41, 7.39, 7.39, 7.39, 7.39, 7.39
    },
    { // 450 km
0.0, 0.0, 0.0, 5.38, 5.43, 5.49, 5.52, 5.57, 5.59, 5.66, 5.73, 5.76, 5.77,
5.78, 5.79, 5.81, 5.82, 5.84, 5.85, 5.89, 5.95, 6.02, 6.06, 6.11, 6.15,
6.18, 6.18, 6.17, 6.17, 6.18, 6.16, 6.16, 6.15, 6.14, 6.15, 6.14, 6.14,
6.14, 6.14, 6.14, 6.14, 6.16, 6.15, 6.14, 6.14, 6.14, 6.14, 6.16, 6.17,
6.18, 6.19, 6.21, 6.23, 6.23, 6.24, 6.25, 6.26, 6.26, 6.26, 6.27, 6.27,
6.29, 6.31, 6.31, 6.32, 6.33, 6.34, 6.35, 6.34, 6.35, 6.35, 6.36, 6.37,
6.38, 6.39, 6.41, 6.41, 6.42, 6.44, 6.44, 6.46, 6.47, 6.49, 6.52, 6.54,
6.58, 6.62, 6.66, 6.69, 6.73, 6.78, 6.81, 6.87, 6.92, 6.99, 7.04, 7.10,
7.14, 7.19, 7.22, 7.25, 7.27, 7.28, 7.32, 7.32, 7.32, 7.32, 7.32
    },
    { // 500 km
0.0, 0.0, 0.0, 5.37, 5.40, 5.44, 5.48, 5.54, 5.57, 5.65, 5.71, 5.75, 5.75,
5.78, 5.78, 5.80, 5.82, 5.85, 5.85, 5.90, 5.94, 6.00, 6.04, 6.08, 6.12,
6.14, 6.14, 6.14, 6.14, 6.15, 6.15, 6.14, 6.13, 6.13, 6.14, 6.14, 6.12,
6.11, 6.11, 6.11, 6.11, 6.12, 6.11, 6.11, 6.11, 6.12, 6.13, 6.15, 6.16,
6.17, 6.18, 6.20, 6.21, 6.21, 6.22, 6.23, 6.23, 6.23, 6.23, 6.25, 6.25,
6.26, 6.30, 6.29, 6.31, 6.32, 6.34, 6.34, 6.34, 6.34, 6.34, 6.35, 6.36,
6.37, 6.37, 6.38, 6.38, 6.39, 6.41, 6.42, 6.44, 6.45, 6.47, 6.50, 6.52,
6.57, 6.61, 6.65, 6.68, 6.72, 6.77, 6.81, 6.86, 6.93, 6.99, 7.04, 7.10,
7.13, 7.18, 7.20, 7.23, 7.25, 7.28, 7.34, 7.34, 7.34, 7.34, 7.34
    },
    { // 550 km
0.0, 0.0, 0.0, 5.42, 5.44, 5.48, 5.50, 5.55, 5.57, 5.63, 5.66, 5.70, 5.72,
5.75, 5.78, 5.80, 5.84, 5.88, 5.91, 5.95, 5.98, 6.05, 6.10, 6.13, 6.15,
6.16, 6.15, 6.14, 6.12, 6.13, 6.12, 6.12, 6.11, 6.11, 6.11, 6.10, 6.10,
6.10, 6.10, 6.10, 6.09, 6.10, 6.09, 6.09, 6.10, 6.11, 6.13, 6.15, 6.17,
6.19, 6.20, 6.22, 6.22, 6.21, 6.21, 6.21, 6.22, 6.21, 6.21, 6.23, 6.23,
6.25, 6.28, 6.28, 6.29, 6.30, 6.31, 6.30, 6.30, 6.31, 6.30, 6.31, 6.32,
6.33, 6.34, 6.35, 6.37, 6.38, 6.40, 6.41, 6.43, 6.45, 6.47, 6.50, 6.53,
```

```
6.57, 6.61, 6.65, 6.68, 6.73, 6.77, 6.81, 6.86, 6.92, 6.96, 7.01, 7.07,
7.11, 7.15, 7.18, 7.20, 7.23, 7.25, 7.29, 7.29, 7.29, 7.29, 7.29
    },
    { // 600 km
0.0, 0.0, 0.0, 5.43, 5.44, 5.48, 5.50, 5.53, 5.53, 5.59, 5.61, 5.65, 5.69,
5.74, 5.78, 5.82, 5.88, 5.94, 6.01, 6.02, 6.08, 6.15, 6.20, 6.22, 6.21,
6.21, 6.20, 6.16, 6.13, 6.13, 6.11, 6.11, 6.12, 6.12, 6.11, 6.10, 6.10,
6.11, 6.11, 6.10, 6.11, 6.09, 6.10, 6.10, 6.10, 6.12, 6.14, 6.16, 6.17,
6.18, 6.19, 6.21, 6.21, 6.20, 6.19, 6.19, 6.19, 6.19, 6.20, 6.21, 6.21,
6.22, 6.25, 6.26, 6.27, 6.27, 6.28, 6.26, 6.27, 6.27, 6.26, 6.26, 6.28,
6.29, 6.30, 6.31, 6.34, 6.36, 6.37, 6.39, 6.42, 6.45, 6.48, 6.51, 6.53,
6.56, 6.60, 6.64, 6.68, 6.72, 6.76, 6.80, 6.85, 6.91, 6.96, 7.01, 7.07,
7.11, 7.14, 7.18, 7.19, 7.22, 7.25, 7.28, 7.28, 7.28, 7.28, 7.28
    },
    { // 650 km
0.0, 0.0, 0.0, 5.41, 5.31, 5.38, 5.37, 5.37, 5.43, 5.45, 5.51, 5.58, 5.64,
5.69, 5.76, 5.85, 5.89, 5.98, 6.10, 6.09, 6.19, 6.24, 6.23, 6.21, 6.20,
6.20, 6.16, 6.11, 6.10, 6.07, 6.03, 6.05, 6.10, 6.10, 6.10, 6.13, 6.14,
6.15, 6.13, 6.14, 6.15, 6.14, 6.15, 6.14, 6.13, 6.15, 6.16, 6.18, 6.17,
6.18, 6.18, 6.19, 6.20, 6.18, 6.19, 6.19, 6.19, 6.20, 6.20, 6.21, 6.21,
6.21, 6.25, 6.24, 6.26, 6.27, 6.28, 6.28, 6.28, 6.27, 6.26, 6.25, 6.27,
6.28, 6.29, 6.29, 6.30, 6.32, 6.33, 6.35, 6.39, 6.43, 6.47, 6.49, 6.52,
6.55, 6.58, 6.64, 6.69, 6.73, 6.77, 6.82, 6.88, 6.92, 6.96, 7.03, 7.08,
7.10, 7.12, 7.16, 7.20, 7.21, 7.25, 7.28, 7.28, 7.28, 7.28, 7.28
    },
    { // 700 km
0.0, 0.0, 0.0, 4.89, 5.10, 5.29, 5.29, 5.27, 5.27, 5.44, 5.64, 5.67, 5.67,
5.79, 5.79, 5.89, 6.17, 6.05, 6.14, 6.16, 6.13, 6.16, 6.11, 6.08, 6.09,
6.07, 6.04, 6.00, 6.03, 6.01, 6.03, 6.05, 6.05, 6.08, 6.12, 6.15, 6.17,
6.21, 6.23, 6.21, 6.22, 6.19, 6.17, 6.17, 6.16, 6.16, 6.12, 6.10, 6.09,
6.10, 6.12, 6.11, 6.10, 6.11, 6.11, 6.11, 6.09, 6.10, 6.19, 6.20, 6.21,
6.20, 6.23, 6.20, 6.21, 6.19, 6.23, 6.22, 6.22, 6.20, 6.20, 6.23, 6.24,
6.27, 6.28, 6.33, 6.35, 6.39, 6.42, 6.47, 6.53, 6.54, 6.58, 6.58, 6.60,
6.62, 6.64, 6.69, 6.70, 6.75, 6.74, 6.76, 6.83, 6.88, 6.96, 6.96, 7.00,
7.03, 7.07, 7.13, 7.18, 7.18, 7.21, 7.24, 7.24, 7.24, 7.24, 7.24
    }
};

#define max(x,y) ((x)>(y)?(x):(y))
#define min(x,y) ((x)<(y)?(x):(y))

static double
bmagnz(double amplitude, double period, double delta, double depth, int
*err)
{
        int j, k;
        double mb, s1, s2, q1, Q;

        *err = 0;

        if (depth < 0. || depth > 700. || delta < 5 || delta > 105) {
                *err = -1;
                return 0.;
        }

        if (depth < 100) {
                k  = (int) (depth/25.) + 2;
                s1 = 0.04*(depth-25.*(k-2));
        }
        else {
                k = min((int)(depth/50.)+4, 17);
```

```
                s1 = .02*(depth-50.*(k-4));
        }
        j  = max(min((int)(delta), 108), 2);
        s2 = delta-j;
        q1 = __qmb[k-2][j-2] + s1*(__qmb[k-1][j-2]-__qmb[k-2][j-2]);
        Q  = q1+s2*((__qmb[k-2][j-1]+s1*(__qmb[k-1][j-1]-__qmb[k-2][j-1]))-
q1);

        mb = log10(amplitude/period) + Q;

        return mb;
}

namespace Seiscomp {
namespace Magnitudes {

bool
compute_mb(
        double amplitude, // in micrometers
        double period,    // in seconds
        double delta,     // in degrees
        double depth,     // in kilometers
        double *mag)
{
        int err;
        double m = bmagnz(amplitude, period, delta, depth, &err);
        if (err) return false;

        *mag = m;

        return true;
}

bool
compute_mb_fromVelocity(
        double amplitude, // in micrometers/second
        double delta,     // in degrees
        double depth,     // in kilometers
        double *mag)
{
        int err;
        double m = bmagnz(amplitude, 2*M_PI, delta, depth, &err);
        if (err) return false;

        *mag = m;

        return true;
}

} // namespace Magnitudes
} // namespace Seiscomp
```