


```

    if (inst.eq.466) sf = 1.00061
    if (inst.eq.69) sf = 1.00036
    if (inst.eq.618) sf = 1.00083
    if (inst.eq.431) sf = 1.000772
    if (inst.eq.444) sf = 1.000646
  endif
  write(*,22) sf
  format(' Gravimeter scale factor applied: ',f8.6)
22 else
  backspace(10)
c
c read observation line
c
  if (lfeedback) then
    read(10,*,end=90) ino,iday,t,r1,r2
    rdg = r1 + (r2*1.002771d0 - r2**2*166.534687d-6)/sf
  else
    read(10,*,end=90) ino,iday,t,rdg
  endif
  if (inst.eq.0) stop '*** Error: no header in start of file'
  no = no+1
c
  if (lcoor) then
    call listno(ino,istat,nc,ii)
    if (ii.eq.0) then
      write(*,*) '*** station missing in coordinate list: ',ino
      goto 20
    else
      glat = rlat(ii)
      glon = rlon(ii)
      gh = rh(ii)
    endif
  endif
c
c astronomical tide correction
  ep = epoch1900(iday,t)-izone/24.d0
  tg = tide(ep,glat,glon,gh)
c earth tide correction
  tg = tg*delta + 0.00483 - 0.01573*cos(glat/radeg)**2
c calibration table correction
  g = sf*calib(inst,rdg) + tg
c
  write(30,30) ino,iday,t,no,rdg,tg,g
30 format(i6,2X,i8,',',f5.2,i6,3f10.3,2X,I3)
  endif
c
  goto 20
c
90 write(*,*) 'Number of observations in list: ',no
  write(*,*)
  write(*,*) 'Total no of observation lists: ',nf
  end
c
  double precision function calib(inst,rdg)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c          c a l i b
c
c calibrates LaCoste and Romberg gravity readings with the
c manufacturers tables by linear interpolation.
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  double precision rdg,a1,a2
c
  dimension g867(71),g69(71),g378(71),g466(71),g495(71)
*,g79(71),g85(71),g298(71),g45(71),g618(71),g431(71),g444(71)
c
  data g867 /
*0000.00, 101.10, 202.19, 303.27, 404.36, 505.43,

```

```

*0606.51, 707.59, 808.66, 909.74, 1010.82, 1111.89,
*1212.97, 1314.05, 1415.14, 1516.22, 1617.31, 1718.40,
*1819.50, 1920.60, 2021.71, 2122.82, 2223.95, 2325.09,
*2426.24, 2527.41, 2628.59, 2729.77, 2830.97, 2932.17,
*3033.38, 3134.61, 3235.84, 3337.08, 3438.33, 3539.59,
*3640.86, 3742.14, 3843.43, 3944.73, 4046.04, 4147.36,
*4248.69, 4350.02, 4451.36, 4552.71, 4654.07, 4755.43,
*4856.80, 4958.17, 5059.55, 5160.94, 5262.32, 5363.71,
*5465.09, 5566.48, 5667.86, 5769.24, 5870.61, 5971.98,
*6073.35, 6174.70, 6276.04, 6377.38, 6478.69, 6579.99,
*6681.27, 6782.53, 6883.78, 6985.00, 7086.21 /
c
  data g69 /
*0000.00, 102.94, 205.87, 308.80, 411.72, 514.65,
*0617.57, 720.50, 823.42, 926.34, 1029.26, 1132.19,
*1235.11, 1338.04, 1440.97, 1543.91, 1646.85, 1749.81,
*1852.78, 1955.77, 2058.76, 2161.76, 2264.76, 2367.77,
*2470.78, 2573.80, 2676.84, 2779.90, 2882.98, 2986.08,
*3089.21, 3192.35, 3295.51, 3398.67, 3501.84, 3605.02,
*3708.20, 3811.38, 3914.59, 4017.82, 4121.08, 4224.35,
*4327.63, 4430.93, 4534.23, 4637.53, 4740.84, 4844.15,
*4947.46, 5050.78, 5154.12, 5257.47, 5360.83, 5464.19,
*5567.55, 5670.90, 5774.24, 5877.58, 5980.90, 6084.20,
*6187.51, 6290.81, 6394.10, 6497.37, 6600.63, 6703.88,
*6807.10, 6910.29, 7013.46, 7116.60, 7219.71/
c
  data g378 /
*0000.00, 105.80, 211.61, 317.41, 423.22, 529.04,
*0634.86, 740.69, 846.53, 952.38, 1058.24, 1164.12,
*1270.01, 1375.92, 1481.84, 1587.78, 1693.73, 1799.69,
*1905.66, 2011.65, 2117.66, 2223.68, 2329.72, 2435.78,
*2541.86, 2647.95, 2754.06, 2860.18, 2966.32, 3072.48,
*3178.65, 3284.84, 3391.06, 3497.28, 3603.53, 3709.79,
*3816.06, 3922.35, 4028.65, 4134.97, 4241.30, 4347.64,
*4453.99, 4560.35, 4666.72, 4773.10, 4879.49, 4985.89,
*5092.30, 5198.72, 5305.14, 5411.57, 5518.01, 5624.45,
*5730.89, 5837.34, 5943.78, 6050.23, 6156.67, 6263.10,
*6369.54, 6475.96, 6582.39, 6688.80, 6795.21, 6901.60,
*7007.98, 7114.35, 7220.69, 7327.01, 7433.31/
c
  data g466 /
*0000.00, 104.93, 209.87, 314.82, 419.76, 524.71,
*0629.66, 734.60, 839.54, 944.49, 1049.43, 1154.38,
*1259.34, 1364.30, 1469.27, 1574.25, 1679.23, 1784.23,
*1889.23, 1994.24, 2099.26, 2204.30, 2309.34, 2414.40,
*2519.48, 2624.57, 2729.67, 2834.79, 2939.93, 3045.08,
*3150.24, 3255.42, 3360.62, 3465.83, 3671.05, 3676.29,
*3781.55, 3886.81, 3992.10, 4097.39, 4202.70, 4308.02,
*4413.35, 4518.70, 4624.06, 4729.42, 4834.80, 4940.19,
*5045.58, 5150.98, 5256.38, 5361.79, 5467.20, 5572.62,
*5678.04, 5783.46, 5888.88, 5994.31, 6099.74, 6205.16,
*6310.58, 6415.98, 6521.38, 6626.76, 6732.13, 6837.49,
*6942.83, 7048.15, 7153.45, 7258.72, 7363.96/
c
  data g495 /
*0000.00, 103.30, 206.60, 309.89, 413.17,
*0516.46, 619.75, 723.05, 826.35, 929.65,
*1032.96, 1136.28, 1239.60, 1342.93, 1446.27,
*1549.62, 1652.98, 1756.35, 1859.73, 1963.12,
*2066.52, 2169.94, 2273.36, 2376.80, 2480.25,
*2583.71, 2687.18, 2790.67, 2894.17, 2997.68,
*3101.21, 3204.74, 3308.30, 3411.86, 3515.44,
*3619.03, 3722.64, 3826.26, 3929.89, 4033.53,
*4137.18, 4240.84, 4344.51, 4448.20, 4551.89,
*4655.59, 4759.30, 4863.01, 4966.74, 5070.47,
*5174.20, 5277.95, 5381.69, 5485.44, 5589.18,
*5692.93, 5796.67, 5900.41, 6004.15, 6107.89,
*6211.61, 6315.33, 6419.04, 6522.73, 6626.41,
*6730.07, 6833.72, 6937.35, 7040.95, 7144.54,

```

```
*7248.10/
c
data g79 /
*0000.00, 0103.66, 0207.30, 0310.93, 0414.55, 0518.16,
*0621.77, 0725.38, 0828.98, 0932.58, 1036.19, 1139.79,
*1243.39, 1346.99, 1450.60, 1554.20, 1657.81, 1761.41,
*1865.02, 1968.63, 2072.24, 2175.86, 2279.50, 2383.14,
*2486.79, 2590.46, 2694.13, 2797.80, 2901.48, 3005.17,
*3108.87, 3212.57, 3316.29, 3420.02, 3523.77, 3627.52,
*3731.29, 3835.06, 3938.84, 4042.62, 4146.41, 4250.20,
*4353.99, 4457.79, 4561.59, 4665.40, 4769.20, 4873.00,
*4976.81, 5080.62, 5184.43, 5288.23, 5392.04, 5495.84,
*5599.63, 5703.43, 5807.21, 5910.99, 6014.77, 6118.54,
*6222.30, 6326.05, 6429.79, 6533.51, 6637.22, 6740.92,
*6844.59, 6948.24, 7051.87, 7155.48, 7259.07/
```

```
c
data g85 /
*0000.00, 0103.46, 0206.91, 0310.36, 0413.80, 0517.25,
*0620.70, 0724.16, 0827.62, 0931.07, 1034.53, 1137.98,
*1241.44, 1344.89, 1448.35, 1551.82, 1655.29, 1758.77,
*1862.26, 1965.76, 2069.27, 2172.79, 2276.32, 2379.84,
*2483.37, 2586.89, 2690.43, 2793.97, 2897.53, 3001.10,
*3104.68, 3208.27, 3311.88, 3415.49, 3519.12, 3622.75,
*3726.39, 3830.05, 3933.71, 4037.38, 4141.06, 4244.74,
*4348.43, 4452.13, 4555.83, 4659.54, 4763.26, 4866.99,
*4970.71, 5074.44, 5178.16, 5281.89, 5385.61, 5489.33,
*5593.05, 5696.75, 5800.45, 5904.13, 6007.80, 6111.46,
*6215.11, 6318.74, 6422.37, 6525.97, 6629.56, 6733.13,
*6836.68, 6940.20, 7043.70, 7147.18, 7250.64/
```

```
c
data g298 /
*0000.00, 0105.80, 0211.57, 0317.33, 0423.08, 0528.82,
*0634.55, 0740.28, 0846.00, 0951.73, 1057.45, 1163.17,
*1268.89, 1374.61, 1480.34, 1586.07, 1691.80, 1797.54,
*1903.28, 2009.03, 2114.79, 2220.55, 2326.32, 2432.09,
*2537.88, 2643.67, 2749.47, 2855.28, 2961.10, 3066.92,
*3172.76, 3278.60, 3384.45, 3490.31, 3596.17, 3702.04,
*3807.91, 3913.79, 4019.67, 4125.56, 4231.45, 4337.34,
*4443.24, 4549.14, 4655.04, 4760.94, 4866.84, 4972.74,
*5078.63, 5184.53, 5290.42, 5396.30, 5502.18, 5608.05,
*5713.92, 5819.77, 5925.62, 6031.46, 6137.30, 6243.12,
*6348.92, 6454.71, 6560.48, 6666.23, 6771.96, 6877.66,
*6983.33, 7088.98, 7194.60, 7300.19, 7405.76 /
```

```
c
data g45 /
* 000.00, 104.18, 208.34, 312.48, 416.62, 520.76,
* 624.90, 729.04, 833.17, 937.31, 1041.44, 1145.58,
*1249.72, 1353.86, 1458.01, 1562.16, 1666.31, 1770.47,
*1874.64, 1978.81, 2082.99, 2187.18, 2291.38, 2395.59,
*2499.81, 2604.03, 2708.27, 2812.51, 2916.76, 3021.02,
*3125.28, 3229.54, 3333.81, 3438.09, 3542.36, 3646.64,
*3750.93, 3855.21, 3959.48, 4063.75, 4168.03, 4272.32,
*4376.64, 4480.97, 4585.29, 4689.60, 4793.89, 4898.16,
*5002.42, 5106.65, 5210.87, 5315.08, 5419.28, 5523.48,
*5627.66, 5731.84, 5836.02, 5940.20, 6044.43, 6148.70,
*6252.96, 6357.21, 6461.44, 6565.64, 6669.81, 6773.94,
*6878.03, 6982.10, 7086.13, 7190.12, 7294.08 /
```

```
c
data g618/
*0000.00, 102.58, 205.15, 307.71, 410.27, 512.82,
*0615.36, 717.90, 820.43, 922.96, 1025.49, 1128.02,
*1230.54, 1333.07, 1435.60, 1538.13, 1640.67, 1743.22,
*1845.78, 1948.34, 2050.90, 2153.46, 2256.02, 2358.59,
*2461.18, 2563.78, 2666.39, 2769.01, 2871.62, 2974.24,
*3076.87, 3179.50, 3282.14, 3384.78, 3487.44, 3590.09,
*3692.76, 3795.42, 3898.10, 4000.77, 4103.45, 4206.13,
*4308.81, 4411.50, 4514.20, 4616.89, 4719.59, 4822.29,
*4924.99, 5027.69, 5130.38, 5233.08, 5335.78, 5438.46,
*5541.15, 5643.82, 5746.49, 5849.15, 5951.79, 6054.43,
```

```
*6157.05, 6259.66, 6362.25, 6464.82, 6567.38, 6669.92,
*6772.44, 6874.95, 6977.43, 7079.89, 7182.32/
```

```
c
data g431/
* 0.0000, 105.6280, 211.2450, 316.8520, 422.4500, 528.0390,
* 633.6190, 739.1900, 844.7520, 950.3070, 1055.8550, 1161.3970,
*1266.9351, 1372.4680, 1477.9990, 1583.5280, 1689.0601, 1794.5950,
*1900.1340, 2005.6770, 2111.2251, 2216.7800, 2322.3420, 2427.9099,
*2533.4861, 2639.0691, 2744.6609, 2850.2610, 2955.8689, 3061.4871,
*3167.1150, 3272.7529, 3378.3999, 3484.0559, 3589.7200, 3695.3931,
*3801.0750, 3906.7639, 4012.4609, 4118.1650, 4223.8779, 4329.5981,
*4435.3262, 4541.0620, 4646.8052, 4752.5542, 4858.3101, 4964.0718,
*5069.8379, 5175.6079, 5281.3818, 5387.1558, 5492.9282, 5598.6968,
*5704.4629, 5810.2251, 5915.9819, 6021.7329, 6127.4771, 6233.2148,
*6338.9448, 6444.6660, 6550.3750, 6656.0688, 6761.7471, 6867.4058,
*6973.0439, 7078.6611, 7184.2549, 7289.8262, 7395.3730/
```

```
c
data g444/
* 0.0000, 105.3630, 210.7080, 316.0370, 421.3520, 526.6530,
* 631.9410, 737.2170, 842.4830, 947.7390, 1052.9871, 1158.2280,
*1263.4640, 1368.6949, 1473.9220, 1579.1470, 1684.3680, 1789.5861,
*1894.8020, 2000.0179, 2105.2351, 2210.4541, 2315.6760, 2420.9021,
*2526.1340, 2631.3730, 2736.6211, 2841.8770, 2947.1411, 3052.4131,
*3157.6919, 3262.9780, 3368.2700, 3473.5659, 3578.8660, 3684.1670,
*3789.4680, 3894.7681, 4000.0720, 4105.3828, 4210.7012, 4316.0278,
*4421.3628, 4526.7002, 4632.0371, 4737.3721, 4842.7041, 4948.0332,
*5053.3569, 5158.6748, 5263.9858, 5369.2891, 5474.5840, 5579.8711,
*5685.1489, 5790.4199, 5895.6812, 6000.9341, 6106.1768, 6211.4072,
*6316.6152, 6421.7959, 6526.9492, 6632.0752, 6737.1719, 6842.2402,
*6947.2798, 7052.2900, 7157.2700, 7262.2202, 7367.1372/
```

```
c
i = rdg/100+1
if (inst.eq.867) then
  a1 = g867(i)
  a2 = g867(i+1)
elseif (inst.eq.466) then
  a1 = g466(i)
  a2 = g466(i+1)
elseif (inst.eq.618) then
  a1 = g618(i)
  a2 = g618(i+1)
elseif (inst.eq.69) then
  a1 = g69(i)
  a2 = g69(i+1)
elseif (inst.eq.378) then
  a1 = g378(i)
  a2 = g378(i+1)
elseif (inst.eq.495) then
  a1 = g495(i)
  a2 = g495(i+1)
elseif (inst.eq.79) then
  a1 = g79(i)
  a2 = g79(i+1)
elseif (inst.eq.85) then
  a1 = g85(i)
  a2 = g85(i+1)
elseif (inst.eq.298) then
  a1 = g298(i)
  a2 = g298(i+1)
elseif (inst.eq.45) then
  a1 = g45(i)
  a2 = g45(i+1)
elseif (inst.eq.431) then
  a1 = g431(i)
  a2 = g431(i+1)
elseif (inst.eq.444) then
  a1 = g444(i)
  a2 = g444(i+1)
else
```

Mar 11, 09 11:36

grredu_ny.f

Page 7/10

```

      stop '*** ERROR: Instrument calibration table not included'
    endif
c
    calib = a1 + (a2-a1)*(rdg/100.d0-i+1)
    return
  end
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c           l i s t n o
c
c  subroutine for finding station number in station list
c  'idx' is zero if 'istat' is not in array 'ia' of 'n' elements
c  search begins at index 'idx', e.g. from previous call
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  subroutine listno(istat, ia, n, idx)
    dimension ia(*)
    if (n.le.0) then
      idx = 0
      return
    else
      if (idx.le.0.or.idx.gt.n) idx = 1
      j = idx
10     if (ia(j).eq.istat) goto 20
      j = j+1
      if (j.gt.n) j = 1
      if (j.eq.idx) goto 30
      goto 10
    endif
20    idx = j
    return
30    idx = 0
    return
  end
c
  real*8 function epoch1900(iday,time)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c  The procedure compute the time interval in units of days
c  from epoch 1900 (A.D. 1900 January 0.5) to the epoch given
c  by the parameter which must be a GI-standard date.
c  iday is given as an integer in form '13051998', meaning May 13, 1998.
c  time is a real giving hr and min (e.g. 17.22), and possible minute
c  decimals.
c
c  Example:
c  The Julian day number of run-time can be written on current
c  output by the call
c  write(out, entier epoch_1900(date_time) + 2415020)
c
c  KMS. Fortran version by Rene Forsberg, Oct 91
c
c  Modified by Shfaqat Abbas Khan 20/3-2000
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
  implicit none
  real*8 time, m, ex_year
  integer iday, days, year, month, day, h, i, ii, ex_days
c
c  year = mod(iday,100)
c  month = mod(iday/100,100)
c  day = iday/10000
c
c  year = mod(iday,10000)
c  month = mod(iday/10000,100)
c  day = iday/1000000
c
  if (month.le.0.or.month.gt.12.or.

```

Mar 11, 09 11:36

grredu_ny.f

Page 8/10

```

.day.le.0.or.day.gt.31) then
  write(*,*) '****', day, month, year
  stop 'epoch1900 - illegal date spec'
endif

  if (month.EQ.1) ii=0
  if (month.EQ.2) ii=31
  if (month.EQ.3) ii=59
  if (month.EQ.4) ii=90
  if (month.EQ.5) ii=120
  if (month.EQ.6) ii=151
  if (month.EQ.7) ii=181
  if (month.EQ.8) ii=212
  if (month.EQ.9) ii=243
  if (month.EQ.10) ii=273
  if (month.EQ.11) ii=304
  if (month.EQ.12) ii=334

  continue
c
c  taking intercalary days into ccount
  ex_year=(year-1900)/4
  ex_days=int(ex_year)+1
c
c  remember a intercalary day must not be added before 1st March
  if (month.LE.2) then
    ex_days=ex_days-1
  end if
c
c
c  days = (year-1900)*365 + ex_days + ii + day
c  h = time
c  m = (time-h)*100
c
c  epoch1900 = days + h/24.d0 + m/1440.d0 - 0.5d0
c
  return
end
c
  real*8 function tide(time,lat,lon,height)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c           t i d e
c
c  the procedure compute the tidal acceleration due to
c  the moon and the sun.
c
c  Parameters.
c
c  tide      (return value). the tidal acceleration corresponding
c              to a rigid earth in mgal.
c  time      (call value). time in days from epoch 1900,
c              i.e. Greenwich mean noon on December
c              31, 1899.
c  latitude ( - - ). Terrestrial latitude of point of computa-
c              tion, degrees.
c  longitude( - - ). Terrestrial longitude of point of
c              computation, positive east, degrees.
c  height   ( - - ). Height above sea level (or above ellip-
c              soid) of point of computation, meter.
c
c  Reference:
c
c  I.M.Longman: Tidal accelerations due to the moon and the sun.
c  Journal of Geophysical Research, Volume 64, No 12, December 1959.
c
c  Explanatory supplement to
c  The Astronomical Ephemeris and

```

Mar 11, 09 11:36

grredu_ny.f

Page 9/10

```

c The American Ephemeris and Nautical Almanac.
c Her Majestys Stationery Office, 1961
c
c algol program by Willy Weng, 1976.
c fortran version jan 1992, Rene Forsberg
c (c) KMS
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
implicit none
real*8 lat,lon,time,height
real*8 omega, N, sinii, cosii, pi,
.alfa, l, sigma, coslambda, sinlambda, cosphi,
.tt, tt2, tt3, r, e, el, p, h, c, cl, aprim, alprim,
.resd, resdd, i, ii, m, mm, ss, my, ksi, ksil, ll,
.pl, ny, t, ll, costheta, g0, gm, gs, s
c
pi = 3.1415926536d0
e = 0.054899720d0
c = 3.84402d10
cl = 1.495d13
aprim = 1/(c*(1-e*e))
i = 0.08979719
omega = 0.4093146162
ss = 1.993d33
mm = 7.3537d25
my = 6.670d-8
m = 0.074804d0
c
c computation point
c
coslambda = cos(lat/180*pi)
sinlambda = sin(lat/180*pi)
r = 6.378270d8/sqrt(1+0.006738d0*sinlambda**2)
. + height * 100
ll = lon/180*pi
c
c Julian centuries and series in tt,
c Longman (10), (11), (12), (19), (26) and (27).
c
c Sun:
c
c Longman Exp. Sup (p.98)
c h (12) ll
c pl (26) GAMMA
c el (27) e
c
tt = time/36525
tt2 = tt*tt
tt3 = tt2*tt
s = 4.720023438 + 8399.7093*tt + 4.40695d-5*tt2 + 3.29d-8*tt3
p = 5.835124721 + 71.018009*tt - 1.80546d-4*tt2 - 2.181d-7*tt3
h = 4.881627934 + 628.33195*tt + 5.2796 d-6*tt2
N = 4.523588570 - 33.757153*tt + 3.67488d-5*tt2 + 3.87 d-8*tt3
pl = 4.908229467 + 3.0005264d-2*tt + 7.9024d-6*tt2 + 5.81d-8*tt3
el = 0.01675104 - 4.18d-5 *tt - 1.26d-7 * tt2
c
c reciproc distances
c
alprim= 1/(cl*(1-el*el))
resd = 1/c + aprim*e*cos(s-p)
. + aprim*e*e*cos(2*(s-p))
. + 15d0/8*aprim*m*e*cos(s-2*h+p)
. + aprim*m*m*cos(2*(s-h))
resdd = 1/cl + alprim*el*cos(h-pl)
c
c longitude of moons ascending node
c
cosii = cos(omega)*cos(i)
. - sin(omega)*sin(i)*cos(N)

```

Mar 11, 09 11:36

grredu_ny.f

Page 10/10

```

sinii = sqrt(1-cosii**2)
ii = atan(sinii/cosii)
ny = asin(sin(i)*sin(N)/sinii)
c
c longitude and rigth ascension
c
t = 2 * pi * (time - int(time)) + ll
ksil = t + h
ksi = ksil - ny
ll = h + 2*e1*sin(h-pl)
alfa = 2 * atan((sin(omega)*sin(N)/sinii) /
. (1 + cos(N)*cos(ny))
. + sin(N)*sin(ny)*cos(omega)))
sigma = s - N + alfa
l = sigma + 2*e*sin(s-p)
. + 5d0/4*e*e*sin(2*(s-p))
. + 15d0/4*m*e*sin(s - 2*h + p)
. + 11d0/8*m*m*sin(2*(s -h))
c
c zenith angles
c
costheta = sinlambda*sinii*sin(l)
. + coslambda*(cos(ii/2)**2*cos(1-ksi) +
. sin(ii/2)**2*cos(1+ksi))
cosphi = sinlambda*sin(omega)*sin(ll)
. + coslambda*(cos(omega/2)**2*cos(11-ksil) +
. sin(omega/2)**2*cos(11+ksil))
c
c gravities
c
gs = my*ss*r*resdd**3*(3*cosphi**2 - 1)
gm = my*mm*r*resd**3*(3*costheta**2 - 1)
. + 3/2*my*mm*r**2*resd**4*
. (5*costheta**3 - 3*costheta)
g0 = gm + gs
c
c transformation from the cgs unit gal to mgal
tide = g0 * 1000
return
end

```