Environmental Science & Technology
Addition/Correction
pubs.acs.org/est

Correction to Surface Area and the Seabed Area, Volume, Depth, Slope, and Topographic Variation for the World's Seas, Oceans, and Countries

Mark J. Costello,*,† Mark Smith,‡ and Witold Fraczek‡

†Institute of Marine Science, University of Auckland, P. Bag 92019, Auckland 1142, New Zealand

‡Esri, 360 New York St., Redlands, CA 92373, USA

Environ. Sci. Technol. **2010**, *44*, 8821−8828; DOI: 10.1021/es1012752

Costello et al.[1] calculated the area and volume of depth zones in the world's ocean using the 3D Analyst tool of ArcGIS 10.1 by Esri. After publication it was noticed that the total for the depth zones did not equal that for the ocean as a whole. It appeared that the tool underestimated total area and volume by 9%. This was because the tool treats the bathymetry surface as a lattice of points rather than a raster data set of cells. This results in using cell centers rather than the full cell area. The consequence is that the analysis area is reduced by $^1/_2$ cell width around the edges of its entire perimeter. When calculating the volume each of the depth zones was reduced by $^1/_2$ cell around their perimeters, accumulating an underestimation error. To resolve this we recalculated the areas and volumes using ArcPy and NumPy in a python script (Appendix) based on the equation presented by Eakins and Sharman[2] that calculates volume and area using the full size of the cell. This method is also free of potential projection distortion over large spatial extents as it calculates the area in square meters of every raster cell of the input data relative to its latitude, accounting for the curvature of the earth. The area of each cell is multiplied by the value of the depth to calculate the volume in cubic meters of water under each raster cell. All relevant cells are summed to calculate the total volume. With this method the sum of the volumes of all 13 depth zones compares favorably with total volume estimates in Costello et al.[1] and that calculated by Eakins and Sharman[2] of 1 335 000 000 km^3 using 1 arc-minute bathymetry data. We present an improved estimate of ocean volume and area to correct Table 2 in Costello et al.[1] (Table 1).

Table 1. New Calculations of the Area and Volume of Depth Zones in the World Ocean

depth zone (m)	surface area (Km^2)	volume (Km^3)
0−100	19 198 985	812 384
101−500	14 726 125	3 705 676
501−1000	7 049 560	5 160 457
1001−2000	16 065 330	24 717 540
2001−3000	30 464 210	78 698 123
3001−4000	79 526 211	283 380 040
4001−5000	116 759 229	524 077 399
5001−6000	73 432 118	392 966 088
6001−7000	2 908 789	17 883 165
7001−8000	309 250	2 279 288
8001−9000	106 492	895 561
9001−10000	32 234	297 962
10001−10898	1970	20 131
sum	360 580 510	1 334 893 822
rounded	360 600 000	1 334 900 000

APPENDIX

The script used to calculate the total ocean area and volume using 64-bit python and NumPy which are installed with the ArcGIS for Desktop Background Processing or ArcGIS for Server.

```
# SumDepth_ByZone.py
#
# Calculate the volume of the oceans using the formula found at:
#   http://www.ngdc.noaa.gov/mgg/global/etopo1_ocean_volumes.html
#   Eakins, B.W. and G.F. Sharman, Volumes of the World's Oceans
#        from ETOPO1, NOAA National Geophysical Data Center,
#        Boulder, CO, 2010
#
# NOTES: Assumes input raster dataset(s) have the land areas masked
# appropriately
#
# August, 2014
# March 2015    Add display surface area values
#
# esri Applications Prototype Lab


import arcpy
import numpy as np
import math
import sys

# path to data workspace
dPath = r'd:\Projects\OceanVolume\Data\depth_zones'

# raster dataset name or prefix: all raster datasets which match will be
used
# be sure this prefix finds only those you would like to use
inRasPrefix = 'zone_'

arcpy.env.workspace = dPath
rasList = arcpy.ListRasters(inRasPrefix + '*')
if len(rasList) == 0:
    print ("Raster dataset not found...Returning")
    sys.exit("Please check the data path")

# some constants
a = 6378137.0   # equatorial radius in m
e = 0.08181919  # Eccentricity

a2 = a**2
e2 = e**2

e1 = 1. - e2

totVol = 0
totArea = 0
totRounded = 0
for inRas in rasList:
    # get the inRaster properties
    rasXMin = arcpy.Raster(inRas).extent.XMin
    rasXMax = arcpy.Raster(inRas).extent.XMax
    rasYMin = arcpy.Raster(inRas).extent.YMin
    rasYMax = arcpy.Raster(inRas).extent.YMax
    rasNoDataVal = arcpy.Raster(inRas).noDataValue
    rasCellHeight = arcpy.Raster(inRas).meanCellHeight
    rasCellWidth = arcpy.Raster(inRas).meanCellWidth
    d0 =  math.radians(rasCellHeight)
    dl =  math.radians(rasCellWidth)
    d0dl = d0*dl

    # cell center
    topCellCenter = rasYMax - (rasCellHeight / 2)

    # create the depths array
    myArray = arcpy.RasterToNumPyArray(inRas, arcpy.Point(rasXMin,
rasYMin))

    # Clone and transpose the lats array
    y = np.linspace(topCellCenter, rasYMin + (rasCellHeight / 2),
myArray.shape[0] )
    latsArray = np.array([y,]*myArray.shape[1]).transpose()
    del y
```


DOI: 10.1021/acs.est.5b01942
Environ. Sci. Technol. 2015, 49, 7071−7072

```
    # mask the depths array - need to mask out continental areas
    # Assume depths are meters. take only between noData and 0
    depths = np.ma.masked_outside(myArray, rasNoDataVal + 1, 0)
    del myArray

    # Calculate the area
    dA = np.ma.array(a2 * np.cos(np.radians(latsArray))*(e1) * d0dl /  \
        (1 - e2 *(np.sin(np.radians(latsArray)))**2)**2, \
          mask = np.ma.getmask(depths))

    # calculate the volume
    dV = dA * (-depths)

    # make result cubic kilometers and square kilometers rather than meters
    result = np.sum(dV) * 1.0e-9
    surfArea = np.sum(dA) * 1.0e-6

    print("{0} is: {1:,d} km3".format(inRas, int(result)))
    print("{0} is: {1:,d} km2\n".format(inRas, int(surfArea)))

    totVol += result
    totArea += surfArea
    totRounded += round(int(surfArea), -3)

# Also print the non-rounded total volume and area
print("Total volume not rounded is: {0:,d} km3".format(int(totVol)))
print ("Total area not rounded is: {0:,d} km2".format(int(totArea)))
print ("Total area rounded sums is: {0:,d} km2\n".format(totRounded))
print(str('Process Complete...'))
```

NumPy uses in-memory arrays so the 64-bit version is required to build the global ocean bathymetry array. The equation came from Eakins and Sharman.[1]

■ REFERENCES

(1) Costello, M. J.; Cheung, A.; De Hauwere, N. Surface area and the seabed area, volume, depth, slope, and topographic variation for the world's seas, oceans, and countries. *Environ. Sci. Technol.* **2010**, *44* (23), 8821−8828.

(2) Eakins, B. W., Sharman, G. F. 2010. *Vol.s of the World's Oceans from ETOPO1, NOAA*; National Geophysical Data Center, Boulder, CO, http://www.ngdc.noaa.gov/mgg/global/etopo1_ocean_volumes.html (accessed February 23, 2015).

DOI: 10.1021/acs.est.5b01942
Environ. Sci. Technol. 2015, 49, 7071−7072