# Dewatering – Deep well systems

GEO-SLOPE International Ltd. | www.geo-slope.com

1200, 700 - 6th Ave SW, Calgary, AB, Canada T2P 0T8
Main: +1 403 269 2002 | Fax: +1 888 463 2239

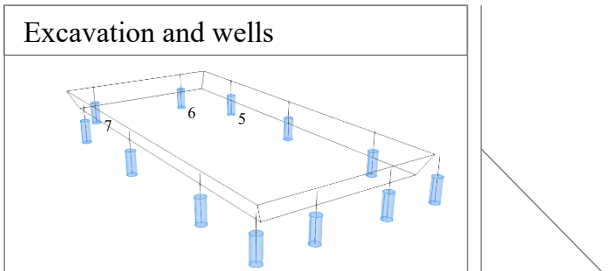
## Introduction

The objective of dewatering is to lower the groundwater table to prevent significant groundwater flow into an excavation, and/or to ensure slope stability. The preferred dewatering system will depend on hydrogeological conditions and construction requirements. In the case of slopes and excavations, the groundwater table can be lowered using a combination of methods, such as deep wells, wellpoints, vacuum wells, and horizontal wells. Deep well systems are often used for dewatering slopes and excavations when large drawdowns are required. This type of system generally consists of an array of pumping wells located near the excavation or slope. The combined effect of the array lowers the groundwater table over a wide area. The active pumping system must be set below the groundwater table whereas the bottom of the wells must be set deep enough to allow flow without excessive head loss.

The objective of this example is to illustrate how to model deep well systems in three-dimensional environments, and to verify the water-flow formulation against a well-known benchmark. To do so, the example will discuss a specific deep well system, and provide insight into an approach for modeling pumping wells.

## Numerical Simulation

The deep well system, taken from Mansur and Kaufman's (1962) chapter on dewatering, consists of a large, 770 foot-long by 370 foot-wide, 40 foot-deep excavation in a 90 foot-deep unconfined sandy aquifer. In order to capture the full effect of the river, the domain was extended 10,000 feet beyond the excavation in each direction. As shown in Figure 1, the centerline of the excavation is located 1000 feet from the edge of the river. The dewatering system consists of twelve pumping wells, located 5 feet from the crown of the excavation slope. The wells are provided with 40 feet of 10-inch diameter screens, and the pumping rate at each well is assumed equal to 1150 gpm (or 2.56 $ft^3$/sec).



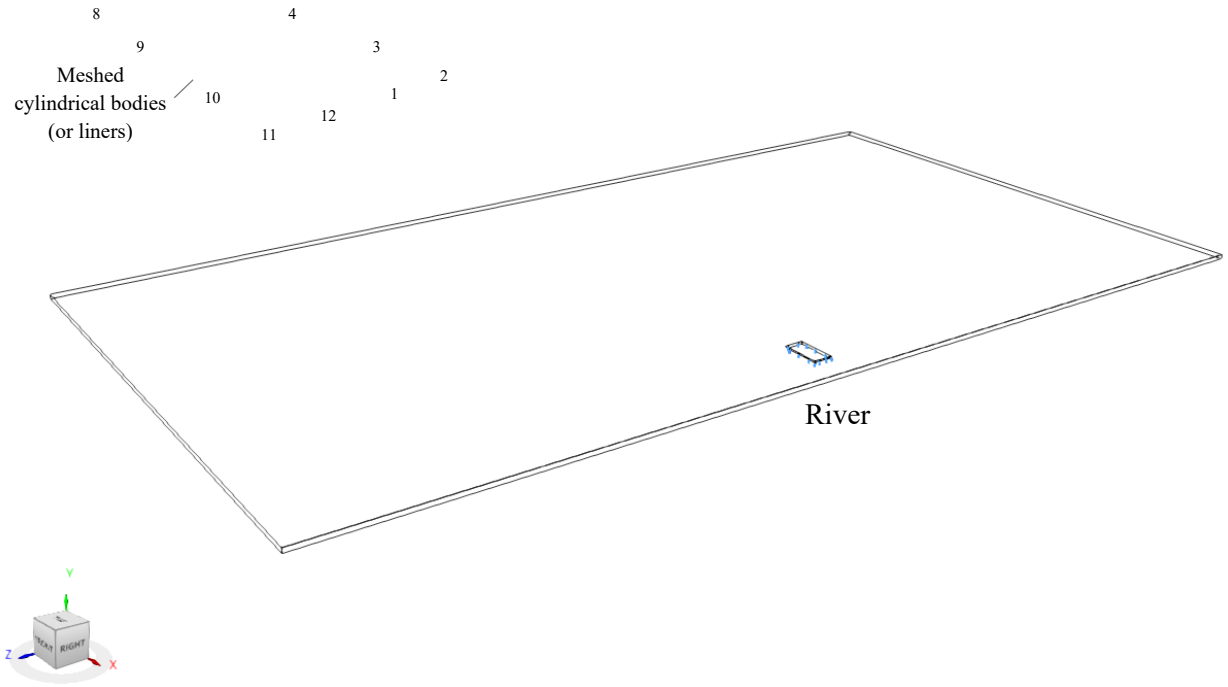



**Figure 1. Problem configuration.**

The Saturated/Unsaturated Material Model was used to describe the sandy aquifer. Although optional in steady-state analyses, the volumetric water content function was used to estimate the hydraulic conductivity function. The volumetric water content function was taken as that of a typical sand with a porosity of 0.35 and a soil-structure compressibility of 5 x $10^{-8}$ /psf. The saturated hydraulic conductivity was set equal to 3.33 x $10^{-3}$ ft/s.

The river level was assumed to remain constant, at 85 feet of elevation, and a potential seepage face was assumed to prevail on the inner surfaces of the excavation. Although the pumping wells can be modeled as three-dimensional cylindrical objects, this can create numerical difficulties in large-scale problems. These difficulties can be alleviated by representing the wells as lines with prescribed water flux and surface perimeter. In this study, each pumping well was modeled as a line representing the effective screen length. The effective screen length accounts for head loss in the wells, and was determined by numerical experimentation. The water flux along the effective screen length was computed as $Q_{pumping}/(PL_e)$ where $P$ is the surface perimeter and $L_e$ is the effective screen length. The downside to this approach is that large hydraulic gradients may prevail near the lines where pumping occurs. This downside was remediated by constraining the mesh using meshed cylindrical bodies (or liners) along the screen length (refer to the insert in Figure 1). In this example, the finite element mesh was generated with 200 ft edge lengths in a tetrahedral pattern, which resulted in 127874 nodes and 678427 elements.

## Results and Discussion

Figure 2 shows the total head isosurfaces and phreatic surface from three different perspectives. As expected, the array of pumping wells prevents groundwater flow from entering the excavation, and results

in lowering of the total head and phreatic surface beneath the excavation. The total head is also found to be highest in the portion of the excavation closest to the river. Overall, the deep well system lowers the phreatic surface, and in so doing, provides additional stability to the side slopes and base of the excavation.

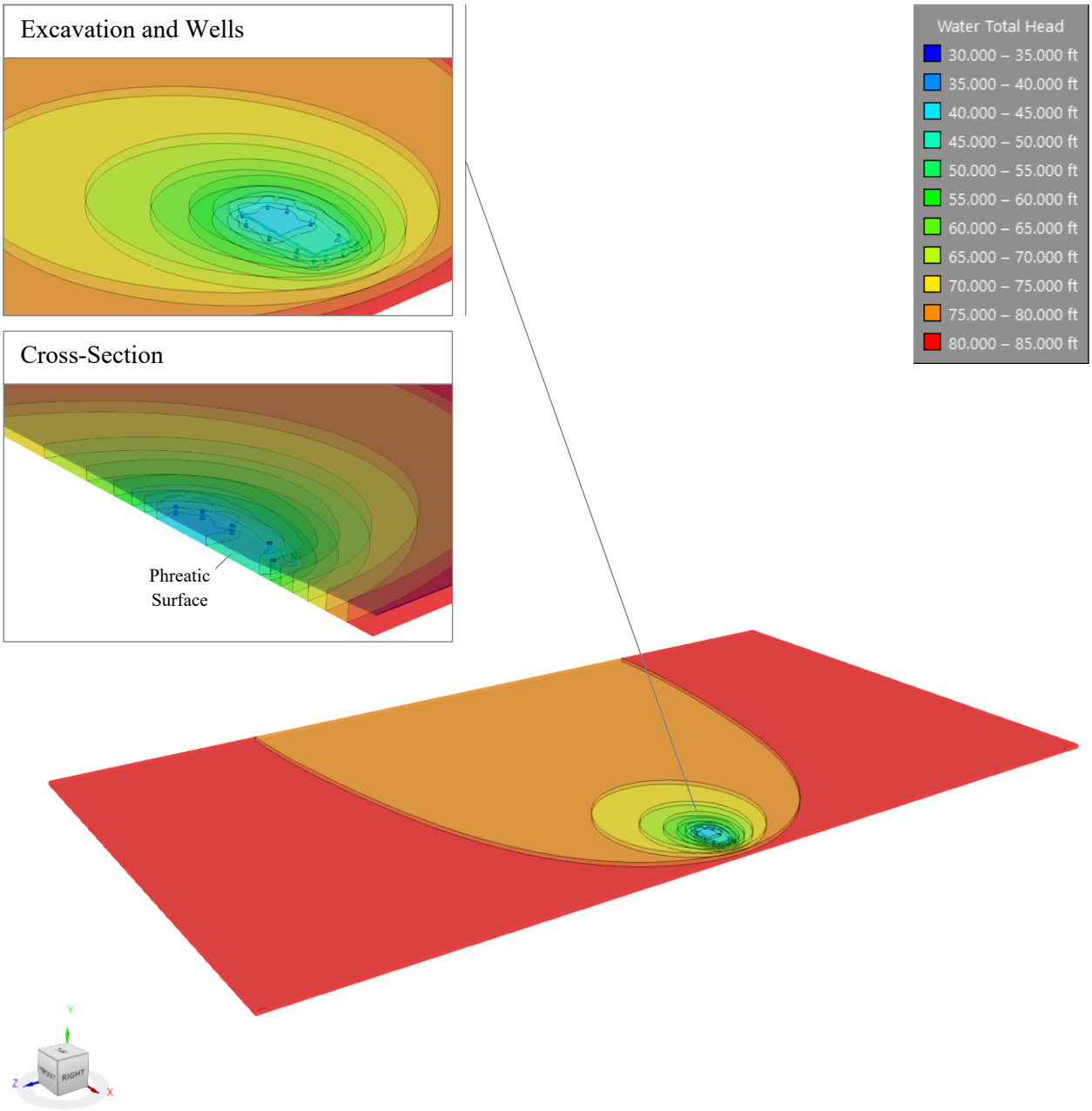



**Figure 2. Total head isosurfaces and phreatic surface.**

Figure 3 shows the total head beneath the center of the excavation, and reveals that it is essentially constant at 44.74 ft.

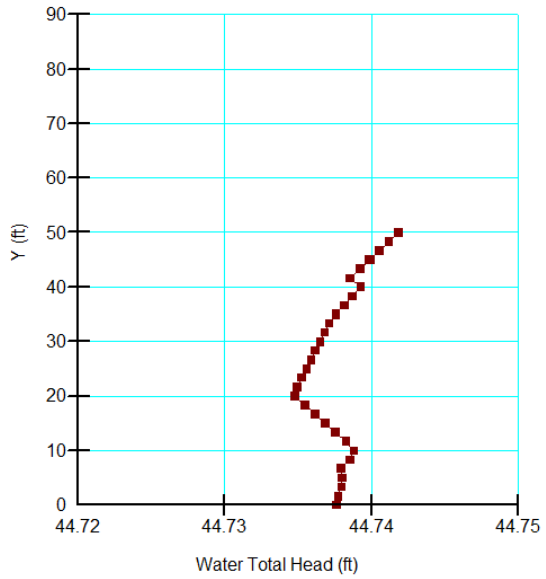
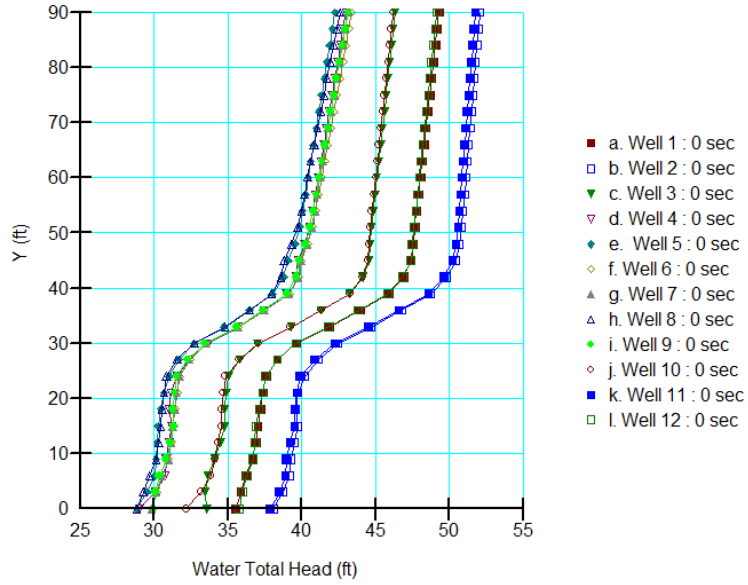



**Figure 3. Total head beneath the center of the excavation.**

Figure 4 shows the total head along the length of the different wells. As in reality, the pumping process results in nearly constant values of total head along the screened portion of the wells. For instance, the total head along the effective screen length of well number four remains approximately constant at 31.52 ft. The total head is also shown to be highest in the wells closest to the river, and symmetrical with respect to the




axis perpendicular to the river.

**Figure 4. Total head along the length of the wells.**

Adequacy of these results can be assessed by comparing them to those obtained using the method of images. As reported by Mansur and Kaufman (1962), the method of images results in values of total head of 44.5 and 32 feet beneath the center of the excavation and at well number four, respectively. These results are in very good agreement with the model results. It must be noted that better agreement can be achieved by decreasing the element edge lengths. It must also be noted that failure to constrain the mesh using cylindrical bodies (or liners) leads to significant errors in the computed values of total head near the wells. This can rapidly be assessed by suppressing the geometry items and their extrusion from the design history.

Figure 5 shows a contour plot of drawdown in plan view. The result was achieved by unsuppressing the clipping planes, setting the Grid Cell Size equal to 10 ft, exporting the elevation of the phreatic surface at each grid point using the Export Isosurface functionality, and running the Python script provided in Appendix. Often preferred by hydrogeologists, the drawdown contour plot provides a fast and easy way of interpreting changes in total head.

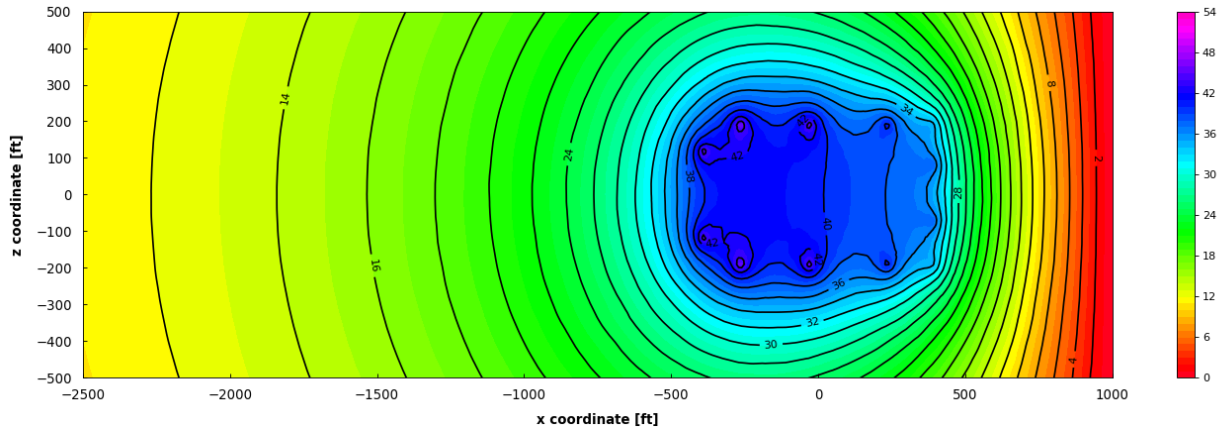



**Figure 5. Drawdown contour plot in plan view.**

## Summary and Conclusions

The capabilities of the software were assessed with a benchmark problem for deep well systems in three-dimensional environments. The results were shown to capture the general trend of the groundwater flow system, and to be in close agreement with those provided by the method of images. Though it was shown that pumping wells could be modeled as lines with prescribed water flux and surface perimeter, it was found that mesh constraints had to be applied around the screened portion of the wells for the sake of accuracy. The history-based approach of the software was also shown to provide a rapid means of assessing the effect of adding mesh constraints to the pumping wells.

## References

Mansur, C.I., and Kaufman, R.I. 1962. Dewatering. In: Leonards G.A. editor, Foundation Engineering, McGraw-Hill Inc.

# Appendix

```
import csv
import numpy as np
import matplotlib.pyplot as plt

#-----------------------------------------------------------------------------------------------------------
# Input
#-----------------------------------------------------------------------------------------------------------

initialWaterTableElevation = 85
currentWaterTableElevation = 'surfaceElevation.csv'
xMin = -2500
xMax =  1000
dx   =  500
yMin =  0
yMax =  54
zMin = -500
zMax =  500
dz   =  100
colorMap = 'gist_rainbow'
viewDataPoints = 'NO'

#-----------------------------------------------------------------------------------------------------------
# Data Manipulation
#-----------------------------------------------------------------------------------------------------------

x  = []
y  = []
z  = []

with open(currentWaterTableElevation,'r') as csvfile:
        surface = csv.reader(csvfile, delimiter=',')
        for row in surface:
                x.append(float(row[0]))
                y.append(float(row[1]))
                z.append(float(row[2]))

dy = [initialWaterTableElevation - x for x in y]

#-----------------------------------------------------------------------------------------------------------
# Contour Plot Generation
#-----------------------------------------------------------------------------------------------------------

plt.figure(figsize = (6 * (xMax - xMin) / (zMax - zMin), 6), dpi = 80, facecolor = 'w')

plt.xlabel('x coordinate [ft]', labelpad = 10, fontsize = 12, fontweight = 'bold')
plt.ylabel('z coordinate [ft]', labelpad = 10, fontsize = 12, fontweight = 'bold')

plt.xlim(xMin, xMax)
plt.ylim(zMin, zMax)
plt.tick_params(axis = 'both', which = 'major', direction = 'in', pad = 10, labelsize = 12)
plt.xticks(np.arange(xMin, xMax + 1, step = dx))
plt.yticks(np.arange(zMin, zMax + 1, step = dz))

plt.tricontourf(x, z, dy, levels = np.arange(0, yMax + 1, 1), cmap = colorMap)
plt.colorbar(boundaries = (yMin, yMax + 1))

contourlines = plt.tricontour(x, z, dy, colors = 'Black', levels = np.arange(0, yMax + 1, 2))
plt.clabel(contourlines, fontsize = 10, fmt = '%1.0f')

if (viewDataPoints == 'YES'): plt.scatter(x, z, s = 1, c = 'Black')
```