PROBABLE MAXIMUM PRECIPITATION
HMR52 Tool Powered by
ARCgis
Presented by Dan Vogler, Donald Cowart and Graham Story, P.E.
INTRODUCTION

• Our nation’s dams and levees - Will they hold back extreme rainfall, surging tides, and wind driven waves from future storms?
• What is the Probable Maximum Precipitation (PMP) event?
• Hydrometeorological Report 52 (HMR52), is used to determine basin averaged rainfall due to the PMP.
• HMR52 will compute the storm area and size, representing them as elliptical rings of greater rainfall intensity. These rings are “referenced” with a storm center and basin boundaries.
• ARCGIS through python scripting will:
  - 1) geo-reference the storm center with real coordinates,
  - 2) compute basin boundaries,
  - 3) sub-areally “clip” the elliptical rings to the basin boundary, and
  - 4) create a grid of this PMP rainfall ready for hydrologic model input.
Flooding Damage
Flooding Damage

PROBABLE MAXIMUM PRECIPITATION HMR52 TOOL POWERED BY ARCGIS
Flood Damage
Probable Maximum Flood (PMF)

- Is the flood due to the Probable Maximum Precipitation (PMP) event.
- Is the maximum event for the Inflow Design Flood (IDF). This event is used for design criteria for the overflow spillway.
- Overflow spillway should be able to pass the PMF and still keep the structure (dam, STA, reservoir, etc.) intact.
Probable Maximum Precipitation (PMP)

- Is the rainfall event that is the theoretical maximum for the area
- Is derived from over 600 measured extreme rainfall events
- Utilizes Depth-Area-Duration
- Seasonal Variation
- Geographical Variation
- [http://www.nws.noaa.gov/oh/hdsc/studies/pmp.html](http://www.nws.noaa.gov/oh/hdsc/studies/pmp.html)
- Florida uses HMR51 as the basis for PMP (east of 105\textsuperscript{th} parallel)
- USACE developed the HMR 52 program to compute the PMP
The Hydrometeorological Designs Studies Center develops the technical basis of PMP estimates and provides guidelines for developing the PMP for a specific area.

All the Hydrometeorological Reports (HMRs) can be downloaded from NOAA’s Hydrometeorological Designs Studies Center webpage, http://www.weather.gov/oh/hdsc/. You will need to download and install the DjVu plug-in to view these documents, which are scanned images. For example, HMR 51 discusses how PMP values were developed for the Eastern US and contains maps of PMP index values, while HMR 52 describes how to compute the PMP for a specific drainage area.
General Approach Used by the NWS for Developing PMP Estimates

Over 500 storms were analyzed when developing PMP estimates for the Eastern US. One of the figures on this slide shows the location of the 30 most influential storm events. The other figure shows depth-area-duration values for event number 77 centered over Warner, OK.
General Approach Used by the NWS for Developing PMP Estimates

Moisture maximization refers to the process of increasing measured storm rainfall for the storm location and season to account for higher atmospheric moisture than was available in the actual storm. Orographic influences were not considered in the development of the PMP in the East (except in Tennessee Valley), however, they were considered for Western states.

These figures show an example of moisture maximization and storm transposition using the storm centered at Warner, OK. The storm was transposed to four locations, A, B, C, and D, and the precipitation adjusted based on the maximum water available at each location and the water available during the storm event.
Regional PMP maps were developed using enveloping curves. Not all areas have experienced extreme precipitation, so in order to prevent discontinuities, smooth curves were used to define PMP amounts. Guidance for determining the shape of enveloping curves was obtained from other kinds of rainfall data like greatest monthly precipitation, greatest weekly precipitation, and 1% rainfall maps.
PMP maps for the Eastern US are found in HMR 51 and HMR 52. A separate map was created for storms of duration 1, 6, 12, 24, 48, and 72 hours and area from 10-20000 sq miles (31 maps). The figures on this slide show the 6- and 72-hour PMP for a storm of 10 sq miles. These maps implicitly contain depth-area and depth-duration information.

Other HMRs contain PMP maps for western regions. For example, HMR 59 contains a map for the 10 square mile 24 hour PMP. Instead of containing maps for multiple depths and durations, HMR 59 contains depth area and depth-duration tables for computing the PMP for your specific watershed.

Orographic affects – HMR 47, 55A, 57, and 59 incorporate orographic influences on PMP estimates.
Snowmelt – HMR 36 and EM 1110-2-1406 contain guidelines for rain on snow.
General Considerations for Developing the PMP

Storm area – think of this as the area of maximum intensity. This is important because rainfall for contributing areas downstream of point of interest (reservoir) is important. Storm area is maximized to produce greatest inflow into project area, however, residual rainfall is needed for streamflow from downstream areas.
The figure on this slide shows a 3D view of the spatial pattern for the most intense 6-hour increment.
Recommended guidance for distributing the PMP depths in time can be found in HMR 52. The guidance is based on examining 28 storms. HMR 52 recommends arranging individual 6-hr increments such that they decrease progressively to either side of the greatest 6-hr increment and placing the four greatest 6-hr increments at any position in the sequence except within the first 24-hr period of the storm sequence.

Following these guidelines, the goal is to develop a time pattern that maximizes runoff. It makes sense to place the more intense precipitation toward the end of the 72-hour event. You can maximize runoff by applying the most intense rainfall when the watershed is saturated. Additionally, historic storms in your area could be used as a source for developing the temporal pattern.
The HMR52 program will intersect the PMP pattern with subbasin boundaries and compute area-average hyetographs. Hyetographs can be written to an HEC-DSS file where they can be read by HEC-HMS. The program will compute the PMP hyetographs and the downstream residual precipitation hyetographs.
HMR52 Program

HMR52 – computes subbasin average PMP hyetographs using:

- X,Y coordinates
- Depth-Area-Duration values from HMR 51
- Storm area
- Storm orientation
- Temporal pattern

The HMR52 program is a command line program that requires an input file containing, among many pieces of information, X,Y coordinates of basin and subbasin boundaries and depth-area-duration information from HMR 51. Using the basin centroid, the program finds the storm area and orientation that maximizes precipitation over the basin. The user has the option to define a different storm center. The user also has the option of defining the temporal pattern.
PROBABLE MAXIMUM PRECIPITATION HMR52 TOOL POWERED BY ARCGIS
import arcgisscripting, sys, os, fileinput, math

gp = arcgisscripting.create()
gp.workspace = "."
gp.overwriteoutput = 1

# Set local variables
StormCenter_X = gp.GetParameterAsText(0)
StormCenter_Y = gp.GetParameterAsText(1)
BasinShape = gp.GetParameterAsText(2)
Basin_Points = "basin_points.shp"
Basin_Name = gp.GetParameterAsText(3)
Storm_Orientation = gp.GetParameterAsText(4)
fcname = gp.GetParameterAsText(5)
output_workspace = gp.GetParameterAsText(6)
gp.workspace = output_workspace
miles2meters = 1609.344*1609.344
deg2rad = math.pi / 180
try:
    # Process: Shape to BX And BY coordinate conversion

gp.FeatureVerticesToPoints (BasinShape, Basin_Points, "all")
print "Reducing Basin Shape to Points"
desc = gp.Describe(Basin_Points)
shapefieldname = desc.ShapeFieldName
printrows = gp.SearchCursor(Basin_Points)
printrow = printrows.Next()
i = 15
j = 0
k = 0
xy = {}
print "Generating Input"
# Reduce Points to every 15th point to get under 100 point limit
while printrow:
    feat = printrow.GetValue(shapefieldname)
pnt = feat.GetPart()
    if i %15 == 0:
        #print j,k
        xy[(j,k)] = pnt.X
        xy[(j,k+1)] = pnt.y
        j = j + 1
        k = k + 1
    i = i + 1
printrow = printrows.Next()
temp = sys.stdout
HMR 52 Input Generation

```python
temp = sys.stdout
gp.workspace = "."
sys.stdout = open("HMR52.DAT", 'w')
print "ID HMR52 INPUT DATA FOR PMF CALCULATION"
print ":FREE"
print "BN ", Basin_Name
print "BS 0.00018939"
i = 0
while i < j:
    item_1 = str(xy[(i,i)][0:6])
    print "BX",item_1
    i = i + 1
i = 0
while i < j:
    item_1 = str(xy[(i,i+1)][0:6])
    print "BY ",item_1
    i = i + 1
print "HO ", Storm_Orientation
print "HP 10 32.0 38.7 47.1 51.8 55.7"
print "HP 200 24.6 31.2 39.5 44.3 48.8"
print "HP 1000 18.2 24.9 33.2 37.7 41.3"
print "HP 5000 10.1 15.0 21.9 26.6 30.7"
print "HP 10000 7.6 11.8 17.6 22.5 26.5"
print "HP 20000 5.6 9.2 13.6 18.0 22.0"
print "SC ",StormCenter_X, StormCenter_Y
print "ST 60 .270"
print "PL 3"
print "ZW A=C44 B=RESERV C=PRECIP-INC D=07FEB2005 F=PMP"
print "ZZ"
```

#------------------------------------ END INPUT GENERATION; NOW RUN THE MODEL
HMR 52 Execution

sys.stdout = temp

    sys_command = "copy C:\HEC\HMR52\HMR52.EXE ."
    os.system(sys_command)
    sys_command = "copy C:\HEC\HMR52\HMR52T.DAT ."
    os.system(sys_command)
    sys_command = "del .\HMRDSS.DSS"
    os.system(sys_command)
    sys_command = "HMR52.EXE INPUT=HMR52.DAT OUTPUT=HMR52.OUT DSSFILE=HMRDSS.DSS"
    os.system(sys_command)
    gp.workspace = output_workspace
    sys_command = "copy .\HMR52.* " + gp.workspace
    os.system(sys_command)

gp.workspace = "."

#file_out = gp.workspace + "HMR52.OUT"
#print file_out
#-----------------------------------END MODEL SIMULATION ; NOW PARSE THE OUTPUTS ---------
Parsing HMR 52 Output

for line in fileinput.input(gp.workspace + '\HMR52.OUT'):
    if 'PREFERRED ORIENTATION ' in line:
        pref_orient = str(line)[64:67]
        print pref_orient
    if '    A      ' in line:
        # print line
        a_split = line.split()
        a_split[1] = 10*miles2meters
        # print a_split[1]
        # Array of arrays of output ellipse values
    if B thru S in line
        gp.workspace = output_workspace
        isoheys = [s_split, r_split, q_split, p_split, o_split, n_split, m_split, l_split, k_split, j_split, i_split, h_split, g_split, f_split, e_split, d_split, c_split, b_split, a_split]
        hours = [19]#1,2,3,4,5,6,7,8,9,10,11,12]
for time in hours:
    try:
        fcnames = fcname + str(time) + ".shp"
        gp.createFeatureClass(gp.workspace, fcnames, 'POLYGON')
        print "Create feature class succeeded!"
        gp.addfield(fcnames, "gridcode", "double", "4", "2")
        print "Add Field succeeded!"
        gp.mask = BasinShape
        print "Mask added successfully!"
    except:
        print gp.GetMessages(2)
for ring in isoheyts:
    values = time
    print values
    # Compute a & b from the ellipse area
    b = math.sqrt(ring[1]/((math.pi)*2.5))
    a = (2.5*b)
    print ring[1], value
    # Convert preferred orientation of radians and set unit circle to N(0:360)
    beta = float(pref_orient) * deg2rad + 90
    sinbeta = math.sin(beta)
    cosbeta = math.cos(beta)
    steps = 360
    i = 0
    pntObj = gp.CreateObject("Point")
    arrayObj = gp.CreateObject("Array")
    rows = gp.insertCursor(fcnames)
while i < 360:
    # Convert the rotating angles to radians
    alpha = i*deg2rad
    sinalpha = math.sin(alpha)
    cosalpha = math.cos(alpha)
    # Convert rotated coordinate for ellipse to storm center offset
    X = float(StormCenter_X) + (float(a)*cosalpha*cosbeta - float(b)*sinalpha*sinbeta)
    Y = float(StormCenter_Y) + (float(a)*cosalpha*sinbeta + float(b)*sinalpha*cosbeta)
    i = i + (360 / steps)
    row = rows.NewRow()
    # Populate Point Object.
    pntObj.ID = i
    pntObj.x = X
    pntObj.y = Y
    # pntObj.gridcode = value
    # print i, X, Y
    arrayObj.add(pntObj)
    # IF point object include
    # row.shape = pntObj
    # row.ID = pntObj.ID
    # rows.InsertRow(row)
    row.shape = arrayObj
    row.gridcode = value
    rows.InsertRow(row)
Parsing HMR 52 Output – “Georeferencing” Rainfall Rings
try:
    gridname = fcname + "_max"
    clipname = fcname + "\_max\_" + "clipped" + ".shp"
    gp.Clip_analysis(fcnames,BasinShape,clipname)
    print "Isoheytals clipped to basin boundary sucessfully!"
    gp.FeatureToRaster_conversion(clipname, "GRIDCODE", gridname)
    print "Rasterization complete!"
except:
    #arrayObj.RemoveAll()
    print gp.GetMessages(3)

except:
    # Print error message if an error occurs
    print gp.GetMessages()

del gp
Conclusions

- ArcGIS can be successfully integrated with command line programming.
- Python is a quick and easy tool for interfacing ArcGIS with other Windows programs.
- These tools have created input in needed formats (shapes and grids) for use in hydrologic models.
- These tools can be shared with others to be used in a common framework.