This is a dictionary of terms and parameters used in the Soil Moisture Routing Model. Grouping occurs by common function and order of appearance in the script. (Ex. Term – Definition, units, line number of first occurrence)

1. Slope and Aspect for SAM

el – elevation DEM, (m), 29

slope – slope, (m/m), 29

1. Flow Direction

flowunits - A "flow unit" is an elevation difference of one unit for an adjacent cell. An elevation difference of one unit for a diagonal cell is 1/(square root of 2)=0.707 flow units, m/m, 39

north – percent of lateral or surface flow that will leave a cell towards the north, (%), 45  
northeast – 46

east – 47

southeast – 48

south – 49

southwest – 50

west – 51

northwest – 52

1. Residual Moisture Content

residual\_mc\_A – soil moisture in layer A permanently retained against ET and drainage fluxes, %, 67

residual\_mc\_B – soil moisture in layer B permanently retained against ET and drainage fluxes, %, 68

1. Watershed and Linear Reservoir Properties

wshed\_id – number associated with a specific watershed or subwatershed, (), 80

area\_cells – count of cells in a watershed used with gridsize and discharge to convert to a depth, (), 80

res\_vol – the depth of storage in the linear reservoir, (m), 80

res\_coeff – the percentage of reservoir water that becomes baseflow, (%), 80

area – the watershed area, m2, 83

base\_flow – the baseflow of the watershed (taken as 0 initially), (m), 86

roads – a value is assigned to represent road surfaces, (), 104

1. Land use initialization

landuse – values are assigned to each type of land cover/use (1=water, 2-urban/rock/barren, 3-forest/woody wetland, 4-shrub, 5-grass/pasture, 6-row crop), (), 107

max\_canopy\_storage\_amt – water storage capacity for each land cover type, (m)

canopy\_cover – the canopy’s attenuation of solar radiation and resistance to heat transfer according to each landuse type, (%), 120

root\_zone – soil depth where actual ET is accounted for, (m), 159

wiltpt\_amt – the soil moisture at which plants with wilt according to land use type, (m), 170

ETreduction\_mc – the soil water percentage of field capacity that can be extracted by ET from a soil layer, (%), 181

tmax\_rain – temperature above which, all precipitation is assumed to be rain (land use dependent), (°C)

tmin\_snow – temperature below which, all precipitation is assumed to be snow (land use dependent), (°C)

1. Initial Storage Conditions

storage\_amt – soil moisture (initially defined as a percentage of the possible saturation amount), (m), 220

storage\_amt\_A – soil moisture in layer A (initially, the sum of wilt point and a percentage of field capacity), (m), 221

1. Initial Conditions for Snow Accumulation Model

swe – how much water is present if the snowpack was melted (snow water equivalent), (m), 230

snow.age – how old is the snow on the ground for changing albedo, (days), 233

swe.yesterday – the SWE from the prior day time step, (m), 234

albedo – the reflectivity of the snow surface (ground is a constant value, snow changes according to age), (), 235

liquid.water – the amount of liquid water held in the snowpack (if the internal energy of the snowpack and soil layer is not zero then there is no liquid water, otherwise it is assumed at 3% of SWE), (m), 236

ice.content – ice in the snowpack, (m), 237

tsnow\_surf – the temperature of the snow surface (assumes the following: specific heat of the soil is 2.1 KJ/kg/°C, bulk density of the soil is 1000.0 Kg/m3, density of water is 1000.0 Kg/m3, specific heat of the snow is 2.1 KJ/kg/°C, snow skin thickness is 0.02 m, snow skin damping depth is 0.15 m, soil skin damping depth is 0.3 m), (°C), 238

u.surface – the energy content of the surface layer, (KJ/m2), 239

tsnow.pack – the temperature of the snowpack (calculated from energy content), (°C), 240

u.total – internal energy of the snowpack and soil layer, (KJ/m2), 241

Q – Runoff, mm

perc – percolation through the restrictive layer into the linear reservoir, cm

runoff\_total – annual runoff for each water year

runoff\_flow – saturation excess overland flow, cm

saturation\_ - annual saturation, cm

saturation – the percentage of storage that is saturated, %

Psat – days in a water year that saturation occurs, cnt

gridsize – m

time\_step - hrs

temp\_time\_step – number of hours to break up daily time step, hrs

hrly\_tmp –

res\_coeff – the percentage of the linear reservoir that becomes baseflow

base\_flow – cm

landuse –

max\_canopy\_storage\_amt – depth of water a canopy can hold

canopy\_storage\_amt -

canopy\_storage\_amt\_pre – ???

canopy\_cover – type of vegetation cover

kfactor – factor used for the day-degree snowmelt algorithm (unused for this project?)

tbase –

root\_zone –

wiltpt\_amt –

wiltpt\_mc\_A –

wiltpt\_mc\_B –

soil\_depth –

soil\_depth\_A –

soil\_depth\_B –

ETreduction\_mc –

ET\_coeff –

tmax\_rain –

tmin\_snow –

storage\_amt\_ini –

storage\_amt\_pre – ????

storage\_diff –

storage\_amt –

root\_storage\_amt –

storage\_amt\_A –

storage\_amt\_A\_tmp –

storage\_amt\_B –

sat\_amt –

sat\_amt\_A –

sat\_amt\_B –

fieldcap\_amt –

fieldcap\_amt\_A –

fieldcap\_amt\_B –

swe –

snowmelt –

snow.age –

swe.yesterday –

albedo –

liquid.water –

ice.content –

tsnow\_surf –

u.surface – KJ/m2

tsnow.pack – C

u.total – KJ/m2

mass\_balance\_total –

date –

year –

tmax –

tmin –

tavg –

precip –

pet\_snotel –

pet –

rain –

snow –

throughfall –

cc\_forest –

cc\_partial –

cc\_open –

output –

t0 –

t1 –

t2 –

t3 –

tdew –

cloud –

rh\_snow – roughness, s/m

rh\_veg – roughness, s/m

l\_turb –

roads –

road\_runoff –

Kfc\_A –

Kfc\_B –

Ksat\_matrix\_A –

Ksat\_matrix\_B –

Ksat\_mpore\_A –

Ksat\_mpore\_B –

rh – wind roughness, s/m

albedo –

t\_diff –

coefdh –

coefbh –

linke\_value –

beam\_rad –

diff\_rad –

refl\_rad –

r.sun – ??

q.srad – short wave radiation, KJ/m^2

q.lw – long wave radiation, KJ/m^2

q.latent – latent heat release, KJ/m^2

q.vap – heat of vaporization, KJ/m^2

q.sensible – sensible heat flux, KJ/m^2

q.rain.ground - the combined conductive heat from rain, snow, and ground

q.total – net radiation, KJ/m^2

vap.d.snow\_sat –

vap.d.air –

condens –

refreeze –

liquid.water –

ice.content –

water\_input – the depth of snowmelt and throughfall, cm

temp\_sum –

actualET\_daily\_flow –

actualET\_flow –

perc\_daily\_flow –

runoff\_daily\_flow –

lateral\_daily\_out –

lateral\_daily\_in –

lateral\_flow\_in –

lateral\_flow –

input\_daily –

SM\_test -

mass\_daily\_balance –

mass\_balance\_total –

input\_daily\_balance –

flowunits –

north –

east –

west –

south –

southeast –

southwest –

northeast –

northwest –

precip\_cm\_ -

rain\_cm\_ -

actualET\_flow\_cm\_ -

canopy\_evap\_cm\_ -

snowmelt\_cm\_ -

reservoir\_vol – the

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 8 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Storage amount output

print `r.stats.zonal base=MASK cover=storage\_amt out=temp8 method=sum --o`;

print $storage\_amt\_cm\_{$wshed\_id} = `r.stats -A input=temp8 nv= `;

$storage\_amt\_cm\_{$wshed\_id} = $storage\_amt\_cm\_{$wshed\_id}\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 9 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Throughfall output

print `r.stats.zonal base=MASK cover=throughfall out=temp9 method=sum --o`;

print $throughfall\_cm\_{$wshed\_id} = `r.stats -A input=temp9 nv= `;

$throughfall\_cm\_{$wshed\_id} = $throughfall\_cm\_{$wshed\_id}\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 10 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Canopy Storage Amount output

print `r.stats.zonal base=MASK cover=canopy\_storage\_amt out=temp10 method=sum --o`;

print $canopy\_storage\_amt\_cm\_{$wshed\_id} = `r.stats -A input=temp10 nv= `;

$canopy\_storage\_amt\_cm\_{$wshed\_id} = $canopy\_storage\_amt\_cm\_{$wshed\_id}\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 11 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Percolation output

print `r.stats.zonal base=MASK cover=perc\_daily\_flow out=temp11 method=sum --o`;

print $perc\_cm\_{$wshed\_id} = `r.stats -A input=temp11 nv= `;

$perc\_cm\_{$wshed\_id} = $perc\_cm\_{$wshed\_id}\*1;

$perc\_cms\_{$wshed\_id} = $perc\_cm\_{$wshed\_id}/100.0\*$gridsize\*$gridsize/($time\_step\*3600.0); # Calculate the percolation in m3/s for sum of all cells

$perc\_mm\_{$wshed\_id} = $perc\_cm\_{$wshed\_id}\*10/$area\_cells; # Convert percolation into area average depth (mm)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 17 swe temporary test \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Snow Water Equivalent (average) output

print `r.stats.zonal base=MASK cover=swe out=temp17 method=sum --o`;

print $swe\_cm\_{$wshed\_id} = `r.stats -A input=temp17 nv= `;

$swe\_cm\_{$wshed\_id} = $swe\_cm\_{$wshed\_id}\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 22 condens temporary test \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Condens (average) output

print `r.stats.zonal base=MASK cover=condens out=temp22 method=sum --o`;

print $condens\_cm\_{$wshed\_id} = `r.stats -A input=temp22 nv= `;

$condens\_cm\_{$wshed\_id} = $condens\_cm\_{$wshed\_id}\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 23 snow temporary test \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Snowfall (average) output

print `r.stats.zonal base=MASK cover=snow out=temp23 method=sum --o`;

print $snow\_cm\_{$wshed\_id} = `r.stats -A input=temp23 nv= `;

$snow\_cm\_{$wshed\_id} = $snow\_cm\_{$wshed\_id}\*1;

=head

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 18 swe temporary test \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Snow Water Equivalent (at SNOTEL) output by MZ 20190109

print `v.what.rast map=snotel raster=swe column=SWE`; # get the swe value at the snotel point and store it in the column SWE

print `v.sample input=snotel column=SWE output=swe\_snotel raster=swe method=bilinear --o`; # sample a swe raster map at snotel vector point using bilinear method

print $swe\_snotel\_cm = `v.db.select -c map=swe\_snotel columns=pnt\_val`; # the value on the snotel point

print $swe\_snotel\_samples\_cm = `v.db.select -c map=swe\_snotel columns=rast\_val`; # the value calculated by bilinear method

$swe\_snotel\_cm = $swe\_snotel\_cm\*1;

$swe\_snotel\_samples\_cm = $swe\_snotel\_samples\_cm\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 19 point swe output \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Swe for point 1 to 6 output by MZ 20200604

print `r.mapcalc 'swe\_1 = swe\*point\_null\_1' --o`;

print `r.mapcalc 'swe\_2 = swe\*point\_null\_2' --o`;

print `r.mapcalc 'swe\_3 = swe\*point\_null\_3' --o`;

print `r.mapcalc 'swe\_4 = swe\*point\_null\_4' --o`;

print `r.mapcalc 'swe\_5 = swe\*point\_null\_5' --o`;

print `r.mapcalc 'swe\_6 = swe\*point\_null\_6' --o`;

print `r.stats.zonal base=point\_null\_1 cover=swe\_1 out=temp19 method=sum --o`;

print $swe\_1 = `r.stats -A -n -N input=temp19`\*1;

print `r.stats.zonal base=point\_null\_2 cover=swe\_2 out=temp19 method=sum --o`;

print $swe\_2 = `r.stats -A -n -N input=temp19`\*1;

print `r.stats.zonal base=point\_null\_3 cover=swe\_3 out=temp19 method=sum --o`;

print $swe\_3 = `r.stats -A -n -N input=temp19`\*1;

print `r.stats.zonal base=point\_null\_4 cover=swe\_4 out=temp19 method=sum --o`;

print $swe\_4 = `r.stats -A -n -N input=temp19`\*1;

print `r.stats.zonal base=point\_null\_5 cover=swe\_5 out=temp19 method=sum --o`;

print $swe\_5 = `r.stats -A -n -N input=temp19`\*1;

print `r.stats.zonal base=point\_null\_6 cover=swe\_6 out=temp19 method=sum --o`;

print $swe\_6 = `r.stats -A -n -N input=temp19`\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 20 point snow output \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Snowfall for point 1 to 5 output by MZ 20190607

print `r.mapcalc 'snow\_1 = snow\*point\_null\_1' --o`;

print `r.mapcalc 'snow\_2 = snow\*point\_null\_2' --o`;

print `r.mapcalc 'snow\_3 = snow\*point\_null\_3' --o`;

print `r.mapcalc 'snow\_4 = snow\*point\_null\_4' --o`;

print `r.mapcalc 'snow\_5 = snow\*point\_null\_5' --o`;

print `r.stats.zonal base=point\_null\_1 cover=snow\_1 out=temp20 method=sum --o`;

print $snow\_1 = `r.stats -A -n -N input=temp20`\*1;

print `r.stats.zonal base=point\_null\_2 cover=snow\_2 out=temp20 method=sum --o`;

print $snow\_2 = `r.stats -A -n -N input=temp20`\*1;

print `r.stats.zonal base=point\_null\_3 cover=snow\_3 out=temp20 method=sum --o`;

print $snow\_3 = `r.stats -A -n -N input=temp20`\*1;

print `r.stats.zonal base=point\_null\_4 cover=snow\_4 out=temp20 method=sum --o`;

print $snow\_4 = `r.stats -A -n -N input=temp20`\*1;

print `r.stats.zonal base=point\_null\_5 cover=snow\_5 out=temp20 method=sum --o`;

print $snow\_5 = `r.stats -A -n -N input=temp20`\*1;

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 21 point actualET output \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Actual ET for point 1 to 5 output by MZ 20190607

print `r.mapcalc 'actualET\_1 = actualET\_daily\_flow\*point\_null\_1' --o`;

print `r.mapcalc 'actualET\_2 = actualET\_daily\_flow\*point\_null\_2' --o`;

print `r.mapcalc 'actualET\_3 = actualET\_daily\_flow\*point\_null\_3' --o`;

print `r.mapcalc 'actualET\_4 = actualET\_daily\_flow\*point\_null\_4' --o`;

print `r.mapcalc 'actualET\_5 = actualET\_daily\_flow\*point\_null\_5' --o`;

print `r.stats.zonal base=point\_null\_1 cover=actualET\_1 out=temp21 method=sum --o`;

print $actualET\_1 = `r.stats -A -n -N input=temp21`\*1;

print `r.stats.zonal base=point\_null\_2 cover=actualET\_2 out=temp21 method=sum --o`;

print $actualET\_2 = `r.stats -A -n -N input=temp21`\*1;

print `r.stats.zonal base=point\_null\_3 cover=actualET\_3 out=temp21 method=sum --o`;

print $actualET\_3 = `r.stats -A -n -N input=temp21`\*1;

print `r.stats.zonal base=point\_null\_4 cover=actualET\_4 out=temp21 method=sum --o`;

print $actualET\_4 = `r.stats -A -n -N input=temp21`\*1;

print `r.stats.zonal base=point\_null\_5 cover=actualET\_5 out=temp21 method=sum --o`;

print $actualET\_5 = `r.stats -A -n -N input=temp21`\*1;

=cut

# comment point output script MZ 20190715

=head

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 12 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Streamflow calculations

$reservoir\_vol\_{$wshed\_id}=$reservoir\_vol\_{$wshed\_id}+$perc\_cms\_{$wshed\_id}

-$base\_flow\_{$wshed\_id};

$base\_flow\_{$wshed\_id}=$reservoir\_coeff\_{$wshed\_id} \* $reservoir\_vol\_{$wshed\_id};

$Q\_{$wshed\_id}=$base\_flow\_{$wshed\_id}+$runoff\_cms\_{$wshed\_id}; # m3/s for sum of all cells

$Q\_mm\_{$wshed\_id} = $Q\_{$wshed\_id}\*86400\*1000/($area\_cells\*$gridsize\*$gridsize); #convert Q into average depth in mm

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 13 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Mass balance daily output$lateral\_in\_cm\_{$wshed\_id} $lateral\_out\_cm\_{$wshed\_id} $mass\_cm\_{$wshed\_id}

print `r.stats.zonal base=MASK cover=mass\_daily\_balance out=temp12 method=sum --o`;

print $mass\_cm\_{$wshed\_id} = `r.stats -A -n -N input=temp12`;

$mass\_cm\_{$wshed\_id} = $mass\_cm\_{$wshed\_id}\*1; # sum of all cells in cm

$mass\_mm\_{$wshed\_id} = $mass\_cm\_{$wshed\_id}\*10/$area\_cells; # Convert into area average depth (mm)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 14 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Lateral in daily

print `r.stats.zonal base=MASK cover=lateral\_daily\_in out=temp13 method=sum --o`;

print $lateral\_in\_cm\_{$wshed\_id} = `r.stats -A -n -N input=temp13`;

$lateral\_in\_cm\_{$wshed\_id} = $lateral\_in\_cm\_{$wshed\_id}\*1; # sum of all cells in cm

$lateral\_in\_mm\_{$wshed\_id} = $lateral\_in\_cm\_{$wshed\_id}\*10/$area\_cells; # Convert into area average depth (mm)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 15 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Lateral out daily

print `r.stats.zonal base=MASK cover=lateral\_daily\_out out=temp14 method=sum --o`;

print $lateral\_out\_cm\_{$wshed\_id} = `r.stats -A -n -N input=temp14`;

$lateral\_out\_cm\_{$wshed\_id} = $lateral\_out\_cm\_{$wshed\_id}\*1; # sum of all cells in cm

$lateral\_out\_mm\_{$wshed\_id} = $lateral\_out\_cm\_{$wshed\_id}\*10/$area\_cells; # Convert into area average depth (mm)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 16 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Input daily balance

print `r.stats.zonal base=MASK cover=input\_daily\_balance out=temp15 method=sum --o`;

print $input\_cm\_{$wshed\_id} = `r.stats -A -n -N input=temp15`;

$input\_cm\_{$wshed\_id} = $input\_cm\_{$wshed\_id}\*1; # sum of all cells in cm

$input\_mm\_{$wshed\_id} = $input\_cm\_{$wshed\_id}\*10/$area\_cells; # Convert into area average depth (mm)

=cut

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Output File \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Create subwatershed streamflow output file

# open(OUT, ">>Q\_subwshed\_$wshed\_id") || die("Cannot Open File");

# print OUT "$wshed\_id $date $year $runoff\_cms\_{$wshed\_id} $perc\_cms\_{$wshed\_id} $base\_flow\_{$wshed\_id} $Q\_{$wshed\_id} \n";

# close(OUT);

# Create mass balance output file

open(OUT, ">>M\_subwshed\_$wshed\_id") || die("Cannot Open File");

# print OUT "$wshed\_id $date $year $precip\_cm\_{$wshed\_id} $rain\_cm\_{$wshed\_id} $canopy\_storage\_amt\_cm\_{$wshed\_id} $canopy\_evap\_cm\_{$wshed\_id} $throughfall\_cm\_{$wshed\_id} $snowmelt\_cm\_{$wshed\_id} $actualET\_flow\_cm\_{$wshed\_id} $perc\_cm\_{$wshed\_id} $runoff\_cm\_{$wshed\_id} $storage\_amt\_cm\_{$wshed\_id} $road\_runoff\_cm\_{$wshed\_id} $mass\_cm\_{$wshed\_id} \n";

# print OUT "$wshed\_id $date $year $precip\_cm\_{$wshed\_id} $rain\_cm\_{$wshed\_id} $canopy\_storage\_amt\_cm\_{$wshed\_id} $canopy\_evap\_cm\_{$wshed\_id} $throughfall\_cm\_{$wshed\_id} $snowmelt\_cm\_{$wshed\_id} $actualET\_flow\_cm\_{$wshed\_id} $perc\_cm\_{$wshed\_id} $runoff\_cm\_{$wshed\_id} $storage\_amt\_cm\_{$wshed\_id} $swe\_cm\_{$wshed\_id} $swe\_snotel\_cm $swe\_snotel\_samples\_cm \n"; # for output 80 to 93 MZ 20190715

print OUT "$wshed\_id $date $year $precip\_cm\_{$wshed\_id} $rain\_cm\_{$wshed\_id} $canopy\_storage\_amt\_cm\_{$wshed\_id} $canopy\_evap\_cm\_{$wshed\_id} $throughfall\_cm\_{$wshed\_id} $snowmelt\_cm\_{$wshed\_id} $actualET\_flow\_cm\_{$wshed\_id} $perc\_cm\_{$wshed\_id} $runoff\_cm\_{$wshed\_id} $storage\_amt\_cm\_{$wshed\_id} $road\_runoff\_cm\_{$wshed\_id} $swe\_cm\_{$wshed\_id} $condens\_cm\_{$wshed\_id} $snow\_cm\_{$wshed\_id} \n"; # for output from 94 MZ 20190715

close(OUT);

# Create subwatershed streamflow output file in depth (mm) 05/12/2016

# open(OUT, ">>R\_subwshed\_$wshed\_id") || die("Cannot Open File");

# print OUT "$wshed\_id $date $year $Q\_mm\_{$wshed\_id} $perc\_mm\_{$wshed\_id} \n";

# close(OUT);

# Create point output

# open(OUT, ">>M\_point\_$wshed\_id") || die("Cannot Open File");

# print OUT "$wshed\_id $date $year $swe\_1 $swe\_2 $swe\_3 $swe\_4 $swe\_5 $swe\_6 \n";

# close(OUT);

print `g.remove -f type=raster name=MASK`;

};

};

close (WSHEDS);

=head

# Sum up streamflow from multiple subwatersheds

# Print cumulative streamflow output files for each watershed outlet

open (WSHEDS, "<wshed\_list\_7flumes.ini") || "Can't open file\n";

while (<WSHEDS>) {

chop($\_);

($wshed\_id,$area\_cells,$res\_vol,$res\_coeff) = split;

open(SUBWSHEDS, "<subwsheds\_$wshed\_id.ini") || "Can't open file\n";

while (<SUBWSHEDS>) {

chop($\_);

($subwshed) = split;

if ($subwshed ne $wshed\_id) {

$Q\_{$wshed\_id} = $Q\_{$wshed\_id} + $Q\_{$subwshed};

};

};

close(SUBWSHEDS);

open(OUT, ">>Q\_wshed\_$wshed\_id") || die("Cannot Open File");

print OUT "$wshed\_id $date $year $Q\_{$wshed\_id} \n";

close(OUT);

};

close (WSHEDS);

=cut

};

close (WEATHER);