mkmod

Willem A. Schreüder
Principia Mathematica
August 2011

What is mkmod

- Program to build MODFLOW input files for the ESPAM model
- Replaces the ***readinp.for*** program used with ESPAM 1.x
  - "Tail end" of the recharge tool
- Operates at the entity-model cell level
  - Maps entity level inputs to model cells
  - Time increments are stress periods (months)

What is new in mkmod

- Implements new algorithms for calculating recharge and pumping
  - On farm algorithm
  - Soil Moisture Accounting
  - Returns calculated instead of specified
- Adds output options
  - Steady state initial stress period
  - Separate recharge and well packages
  - Summary tables

Whom to blame for mkmod

- Greg Sullivan (on farm algorithms)
- Willem Schreüder (implementation)
- Jim Brannon (peer review)
- Allan Wylie (user testing)
- David Blew (just because we always blame him for everything)

mkmod inputs

- Command line options
  - Controls algorithms, file names, etc.
- mkmod control file (.mdl)
  - Specifies model dimensions, units, periods,...
- Data files
  - Output from other tools that pre-process data
  - Input data such as efficiencies

mkmod outputs

- MODFLOW input files
  - Net recharge a.k.a. “well term”
  - Individual water budget items
- Summary tables
  - HTML table by period and entity
- Diagnostic outputs
  - Acres by cell, ...
  - User specified summeries

mkmod flow chart

- Read definitions and time invariant data
- Loop over periods
  - Read data for current period
  - Calculate irrigated recharge/pumping
  - Calculate non-irrigated recharge
  - Distribute canal seepage
  - Distribute trib underflow and perched river
  - Save data

Irrigated recharge/pumping

- Applied water by entity
  - Diversion-Canal Seepage+Off-site Pumping
- Loop over entities
  - Loop over gravity & sprinkler lands
    - Loop over cells
      - Calculate CIR, pumping recharge, soil moisture
        - Algorithms depend on command line switches
        - GW and SW treated differently
        - Done on unit area basis, then multiply by acres
      - Accumulate irrigated acres

Crop Irrigation Requirement (CIR)

- Done by cell under the entity
- Adjusted ET = ETadj(Gr/Sp) * ET(Cell)
- Precip = Precip(Cell)
- CIR = Adjusted ET - Precip

For Groundwater Entities

- If CIR<0, pumping = 0
- Else pumping = CIR/Efficiency(Gr/Sp)
- Recharge = Pumping – CIR
- Notes
  - If CIR<0, Recharge = -CIR
  - Net Recharge = -CIR = Recharge-Pumping
  - Recharge = Precip + (1-Eff)*Pumping
  - Soil moisture never changes

For Groundwater Entities

- If $CIR<0$, pumping = 0
- Else pumping = CIR/Efficiency(Gr/Sp)
- Recharge = Pumping – CIR
- Notes
  - If $CIR<0$, Recharge = -CIR
  - Net Recharge = -CIR = Recharge-Pumping
  - Recharge = Precip + (1-Eff)*Pumping
  - Soil moisture never changes

SW Entities - On Farm Algorithm

- Determine over/under irrigation
  - deficit is the amount of under irrigation
  - excess is the amount of over irrigation
- Adjust soil moisture of deficit/excess
- Recharge = DPin*(1-Eff)*Applied + DPex*Excess
- Runoff = (1-DPin)*(1-Eff)*Applied+(1-DPex)*Excess
- Notes
  - When deficit>0, CIR is not met (ET shorted)
  - When excess>0, Actual Efficiency < Eff(Gr/Sp)

Deficit/Excess Calculation (SW)

- NetApp = Eff(Gr/Sp)*Application - CIR
  - if NetApp>0, excess = NetApp, deficit=0
  - if NetApp<0, excess = 0, deficit = -NetApp
- Optional Soil Moisture Adjustment
  - SMsink = field capacity – soil moisture
  - SMsource = soil moisture – wilting point
  - If deficit>0 & SMsource>0
    - SMchange = - min(SMsource,deficit)
    - deficit = deficit + SMchange
  - If excess>0 & SMsink>0
    - SMchange = min(SMsink,excess)
    - excess = excess - SMchange

SW Entities – V1.x Algorithm

- Recharge = Application – CIR
- Notes
  - Assumes Runoff/Returns are known and subtracted from Applied Water
  - Application efficiency could be 100%
  - Needs manual adjustment to avoid negatives
  - Eff*Applied>=CIR, On Farm algorithm gives the same result, but uses DPex & DPin to partition overage to recharge and runoff

Irrigated Entity Summary

- GW – pump to meet CIR
  - SM constant, net pumping=ET-precip
- SW – Apply Diversion–Leakage+Pumping
  - SM adjusted for excess or deficit
  - Recharge fixed fraction of applied water plus fixed fraction of an excess
  - Runoff fixed fraction of applied water plus fixed fraction of excess
  - Deficit shorts ET
- Cell by cell for all entities, Sprinkler/Gravity

mkmod flow chart

- Read definitions and time invariant data
- Loop over periods
  - Read data for current period
  - **Calculate irrigated recharge/pumping** ← You are here
  - Calculate non-irrigated recharge
  - Distribute canal seepage
  - Distribute trib underflow and perched river
  - Save data

Non-irrigated recharge

- Area = Cell Area – Irrigated Area
  - Calculated when doing irrigated recharge
- Loop over all active cells
  - Recharge = Factor(soil) * NIR * Area

Fixed point & off-site terms

- Loop over all locations
  - Add term to budget
  - Location specified by model cell
  - Budget term specified in .mdl file
  - Wetlands is a special case and have an adjustment factor (currently 1)

Canal Seepage

- Volume specified as total for entity
- Location specified as a list of model cells
- Volume is distributed uniformly to cells

Tributary Underflow & Perched River Seepage

- Volumes specified by reach
- Scale factor for each reach
- Location specified as a list of cells
- Distribute uniformly to cells in reach

Output

- Recharge, pumping, etc are accumulated on a cell by cell basis
  - Net Recharge "well term" is the sum of all arrays on a cell by cell basis
  - Arrays can be saved individually or as sum
- Budget terms are accumulated for each irrigation entity
  - Summarized for all SW and GW entities and by entity in the HTML table

Steady State

- Steady State stresses are average of sequence of stress periods
  - Currently 229 (May 1999) to 240 (April 2000)
- mkmod can calculate this average and save it as the first stress period
  - Combines steady state and transient into one MODFLOW run
  - Requires more memory

Running mkmod

- Written in Perl, so needs a Perl interpreter
- Mac OS/X, Unix, Linux
  - mkmod <options and parameters>
- Windows
  - Invoke perl explicitly
    - perl mkmod <options and parameters>
  - Compile to EXE using PAR-packer

Production mode command line

- mkmod -ss E110712A
  - Parameter -ss omits steady state, saves only "well term" and requires only 72MB RAM
  - Input files are named E110712A.*
  - Output file is E110712A.net
  - Run time about 10 minutes

Input files with data that apply to all periods in the simulation

- .mdl Model units, dimensions, periods, ...
- .cel   Active cells and cell areas
- .sol   Soil type for non-irrigated recharge
- .red Reduction factors (Gravity,Sprinkler)
- .eff   On Farm parameters Eff,DPin,DPex,..

Input files with data by Entity

(Header + data set for each stress period)

- .ent Entity names; sprinkler fractions
- .iar Irrigated acres by cell
- .div Diversions and returns
- .cnl Canal leakage
- .fpt Fixed point pumping
- .off Off-site pumping
- .pch Perch river seepage
- .trb Tributary underflow

Input files with cell arrays

(no header, just array of values for each stress period)

- .pre Precipitation
- .eti   Evapotranspiration
- .nir   Non-irrigated recharge

Model Input File (.mdl)

(See annotated mdl file for details)

- Run title (shown in HTML)
- Units
- Stress periods
- Dimensions
- Adjustment factors
- Fixed point definitions
- Debug output
- Minimum non-irrigated area

Cells (.cel) and Soil Types (.sol)

- cel file
  - IBOUND array (1=active,0=inactive)
  - Area by cell (5280 x 5280)
- .sol file
  - Soil index for non-irrigated recharge
  - 1-max (max <= number of soils in .mdl)
  - Dead cells are ignored

ET Reduction Factors (.red)

- Row 1: gravity reduction factor
- Row 2: sprinkler reduction factor
  - Should be a fraction (0=none)
  - One value per stress period

On Farm Factor File (.eff)

- One row for each entiry
  - Name
  - Maximum efficiency gravity (0-1)
  - Maximum efficiency sprinkler (0-1)
  - DPin (Deep perc fraction of applied water)
  - DPex (Deep perc fraction of excess water)
  - Wilting point
  - Field Capacity
  - Depth of rooting zone

Entity File (.ent)

- Header
  - Entity Identifier (e.g. IESW029)
  - Entity type (SW/GW)
  - ET Adjustment Sprinkler
  - ET Adjustment Gravity
  - Display name
- For each stress period
  - Sprinkler Fraction (one value per entity)

Irrigated Area File (.iar)

- No Header
- For each stress period
  - List of cells (row,col,count)
    - List of entities in that cell
    - List of areas (one per entity)
- Many stress periods repeat previous data

Diversions (.div)

- Header
  - Row listing entities
- Each Stress Period
  - Row of diverted amounts (cubic feet)
  - Row of returns (cubic feet)
    - Returns are zeroed when returns are calculated

Canal Leakage (.cnl)

- Header
  - #Cells, Adjustment, Identifier, Text Name
  - Row Column (rest of row is not used)
    - Could be used to weight spatially
- For each stress period
  - Canal leakage fraction (of diversion)

Fixed Point [Pumping] (.fpt)

- Header
  - Type Layer Row Column Name
  - Current types
    - W Wetlands
    - U Urban Pumping
    - E Exchange Pumping
    - M Mud Lake Pumping
- Stress periods
  - Row of values, one per entry (cubic feet)

Off-site Pumping (.off)

- Header
  - Layer Row Column Entity Name
- For each stress period
  - Row of volumes, one per well (cubic feet)

Tributary Underflow (.trb)
Perched river seepage (.pch)

- Header
  - #Cells Factor Name
  - Row Col
- For each stress period
  - Row of volumes per reach (cubic feet)

Precipitation (.pre)
Evapotranspiration (.eti)
Non-irrigated Recharge (.nir)

- No Header
- For each stress period
  - Array of values for every cell (depth in feet)
  - Values in dead cells are ignored
  - Values adjusted for irrigated/non-irrigated area as appropriate

Command Line

- mkmod [parameters] fileroot

  fileroot sets default file root for input and output files
  -o sets alternate output file root
  --xxx sets alternate input file name, where xxx is he file extension (e.g. --ent foo.ent)
  -a save service areas (entityname.dat)
  -s single output file ('well term')
  -ss No steady state, single output file
  -S No steady state, separate output files
  -sss Production mode only output is well file

Command Line Option -m

- FRS (default): Maximum Farm Efficiency using Runoff for Returns and Soil Moisture
- 1: ESPAM 1.1
- MFE: Maximum Farm Efficiency using Fixed Returns
  - ESPAM 1.1 with on farm
- FER: Maximum Farm Efficiency using Runoff for Returns
  - FRS without soil moisture

Output files (-s)

- Net recharge ['well term'] (.net)
  - MODFLOW well file format (k,i,j,q)
- Summary table (.htm)
  - Summarizes input and output by groups and by entity by stress period
- Summary table (.dat)
  - Main summary table for plotting
- Return flows (.rfl)
- Debug output (.rfx)

Output files (separate terms)

- .ppt precipitation recharge
- .can canal seepage
- .tri tributary underflow
- .gwr groundwater deep percolation
- .swr surface water deep percolation
- .wel well pumping

Output Files (acreage)

- .AGWgr Groundwater Gravity Acres
- AGWsp Groundwater Sprinkler Acres
- ASWgr Surface Water Gravity Acres
- ASWsp Surface Water Sprinkler Acres

Summary Table Contents (.htm)

- Title and definitions
- Total Non-irrigated, GW & SW by period
- Average values by GW entity
- Individual GW entities by period
- Average values by SW entity
- Individual SW entities by period
- *See example file for individual fields*
- *Column numbers correspond to .dat file*

Programing Notes 1

- Major functions
  - Read read next non blank/comment line
  - ReadLine Read & split on spaces
  - ReadCSV Read array from CSV file
  - Open open file and read header
  - Next read next period from file
  - Average Average summary values
  - SaveCell save stresses in cell format
  - SaveArea save stresses in array format

Programming Notes 2

- Major Variables
  - Input Data Structures
    - x{type}{...}
    - Structure varies by input data type
  - Output arrays
    - out{type}[period][cell]
    - Volumes (converted to rates when saving)
  - Summary tabes
    - sum[period]{entity}{type}