Land-surface recharge program: User Guide and Documentation

David Scott

December 2013

Introduction

Environment Canterbury has, for several years, applied soil water balance models to the estimation of land-surface recharge and irrigation demand. The procedures to apply these models over spatially extensive areas have been encoded in a series of FORTRAN programs which have been successively adapted to meet specific project requirements. These programs were written to be used by the programmer and often had similar or identical names (variants of recharge_demand.for). The resulting proliferation of similar, but different, programs has created a potentially confusing legacy.

This report describes a revised version of **recharge_demand** which has been extended to accommodate the main variants that had been catered for in the multiple versions of the program. An initialization file allows the user to select the desired options without having to be familiar with FORTRAN programming. The revisions have also involved adapting earlier versions of the program to operate within the new file management environment that was established within Environment Canterbury in August 2013.

Background

The soil water budget model which is incorporated in the **recharge_demand** program is illustrated in Figure 1. Surface runoff is assumed to be negligible – generally a reasonable assumption on the Canterbury Plains. The program applies the following logic to determine drainage and, where required, irrigation demand:

- $AET_i/PET_i = f(W_{i-1}/WHC,A)$
- $I_i = f(W_{i-1}/WHC,T,E)$
- $W_i = W_{i-1} + R_i + I_i - AET_i$
- If Wi > WHC then
    - $D_i = WHC – W_i$
    - $W_i = WHC$

where

- R_i = daily rainfall on day i
- PET_i = potential evapotranspiration on day i
- AET_i = actual evapotranspiration on day i
- W_i = soil moisture level on day i

- WHC = soil water holding capacity
- $f(W_{i-1}/WHC,A)$ is a function relating AET to PET and soil moisture condition
- D_i = drainage on day i
- I_i = irrigation application depth on day i
- $f(W_{i-1}/WHC,T,E)$ is a function which determines irrigation application depth
- T is a user-defined term which controls whether irrigation is triggered
- E is a user-defined term which determines the quantum of irrigation

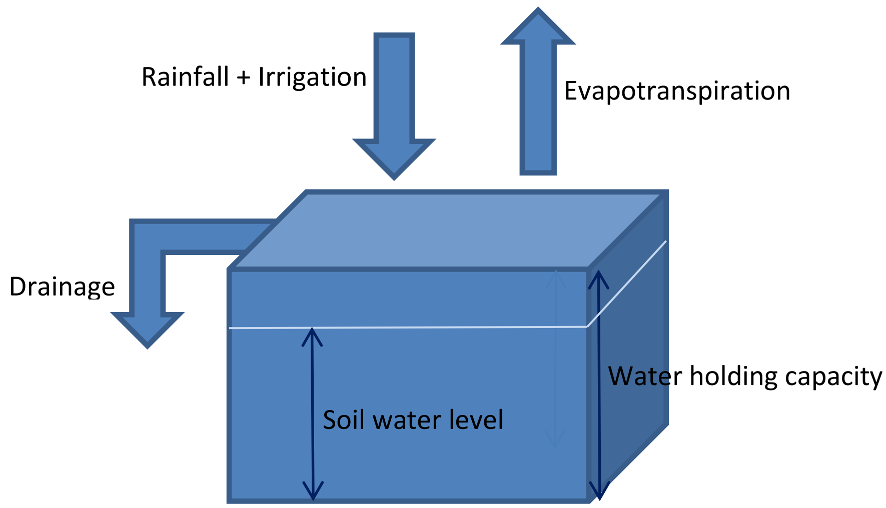

Figure 1: Schematic representation of soil water balance model

The function used to determine actual evapotranspiration is from Minhas et al. (1974):

$$AET = PET \times [\frac{1 - \exp\left(-\frac{A \times W}{WHC}\right)}{1 - 2 \times \exp(-A) - \exp\left(-\frac{A \times W}{WHC}\right)}]$$

The user-defined coefficient A (sometimes referred to as the curve number) determines the shape of the curve on an AET/PET vs W/WHC plot as illustrated in Figure 2.

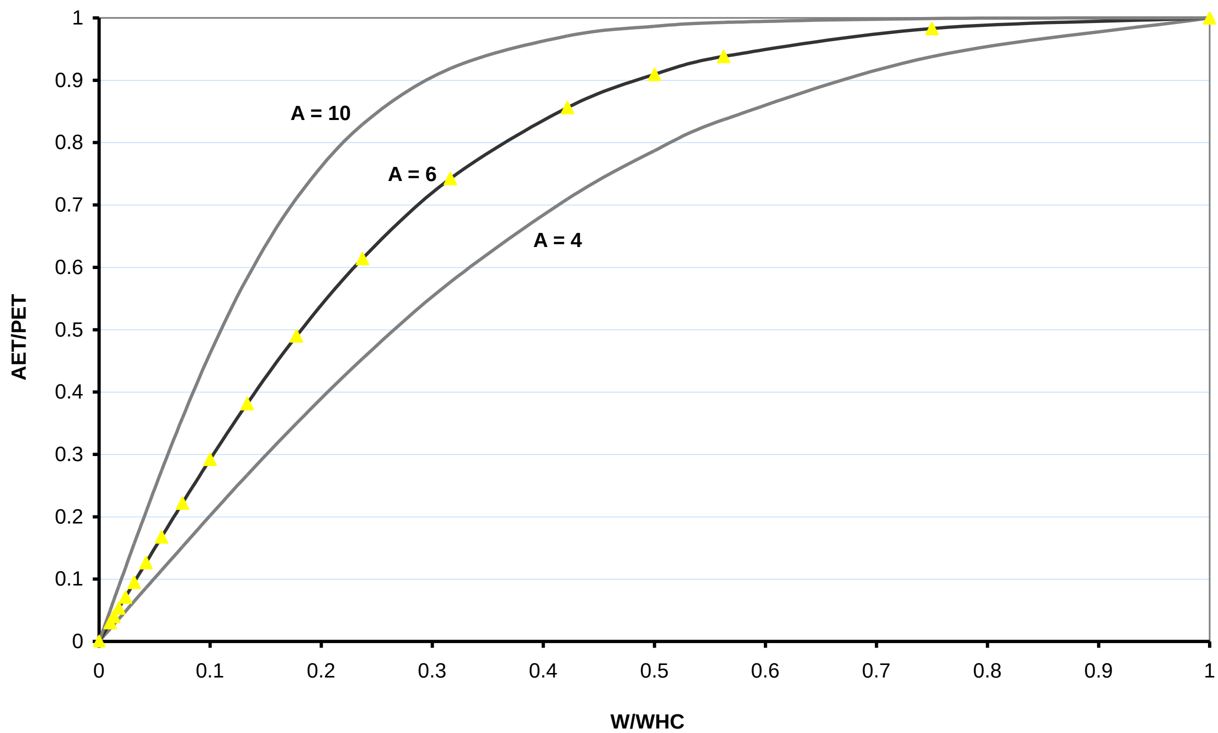

Figure 2: Alternative relationships between actual and potential evapotranspiration as determined by user-defined curve number (A)

The calculations outlined above are relatively simple and are easily implemented in a spreadsheet for a single location. The purpose of the **recharge_demand** program is to streamline the application of the water budget method over spatially extensive areas (e.g. groundwater allocation zones or catchments) which involves discretization of the area into many sub-areas (often using a regular grid), each with representative soil and land-use properties. The program also facilitates the use of NIWA's Virtual Climate Network (VCN) data by identifying the closest VCN station and using the daily climate data for that station in the water budget calculations. Because of this dependence on the VCN data, the application of the **recharge_demand** program relies upon the continued updating of Environment Canterbury's copy of the NIWA climate data. Procedures for maintaining an up-to-date copy are recorded on the Groundwater Resources Section Sharepoint site in the following folder:

- fileservices02\managedshares\data\virtualclimate\procedures\updating\

Though the water budget method used in the **recharge_demand** program is relatively simple it has been shown to provide results that are comparable with more complex methods (Thorpe and Scott 1999, White et al. 2003, Hong et al. 2005). The method has been applied in several Environment Canterbury studies, e.g. for the calculation of long-term average groundwater allocation zone recharge rates (Scott 2004) and for the estimation of the possible history of recharge and demand to support interpretation of response to development (Williams 2010).

Nevertheless, users should be aware of the limitations of the method. Thorley and Ettema (2007) demonstrated that runoff should not be neglected in some of the loess-covered zones of South Canterbury. They proposed alternative (lower) groundwater allocation volumes by incorporating maximum infiltration rates when evaluating average long-term recharge rates. Aitchison-Earl (pers. com.) has subsequently evaluated a range of alternative modifications of the simple model in order to represent the limited infiltration capacity of the loess soils (Appendix B).

Other limiting assumptions of the **recharge_demand** program are that:

- the distribution of land-use does not change in time
- water is available to apply irrigation whenever the triggering condition is encountered.

These assumptions create difficulties when considering situations where the irrigated area and type of irrigation has changed significantly over time. Analysis of those situations requires additional land-use history data and some modification of the way in which the **recharge_demand** program outputs are used. Thorley et al. (2010) illustrate how land-use history and restraints on irrigation availability were incorporated in an analysis of groundwater dynamics in the Rakaia-Ashburton Plains.

User Guide

The executable version of the **recharge_demand** program is located in the following folder:

- fileservices02\managedshares\data\waterbudget

The program reads the following user-defined input files from that folder:

- recharge_demand.ini
- grid_cell.prn

These two files, which are described in detail below, must be customized for each particular application of the **recharge_demand** program. NB read/write privileges to the waterbudget folder are limited to a few named individuals. If a user is unable to update the user-defined files they should request that their name be added to the list of users for that particular folder.

The other input files required by the **recharge_demand** program are located on another folder of the shared drive and no user-intervention is required (or possible).

Once the user-defined input files have been prepared the procedure can be accessed by opening a command window in the waterbudget folder (navigate to that folder using Windows Explorer and select the "Open Command Window Here" option which is made available by holding down the shift key while clicking the right-button on the mouse). The program can then be invoked by entering the command "**recharge_demand**" at the command prompt which should result in output similar to that in Figure 3.

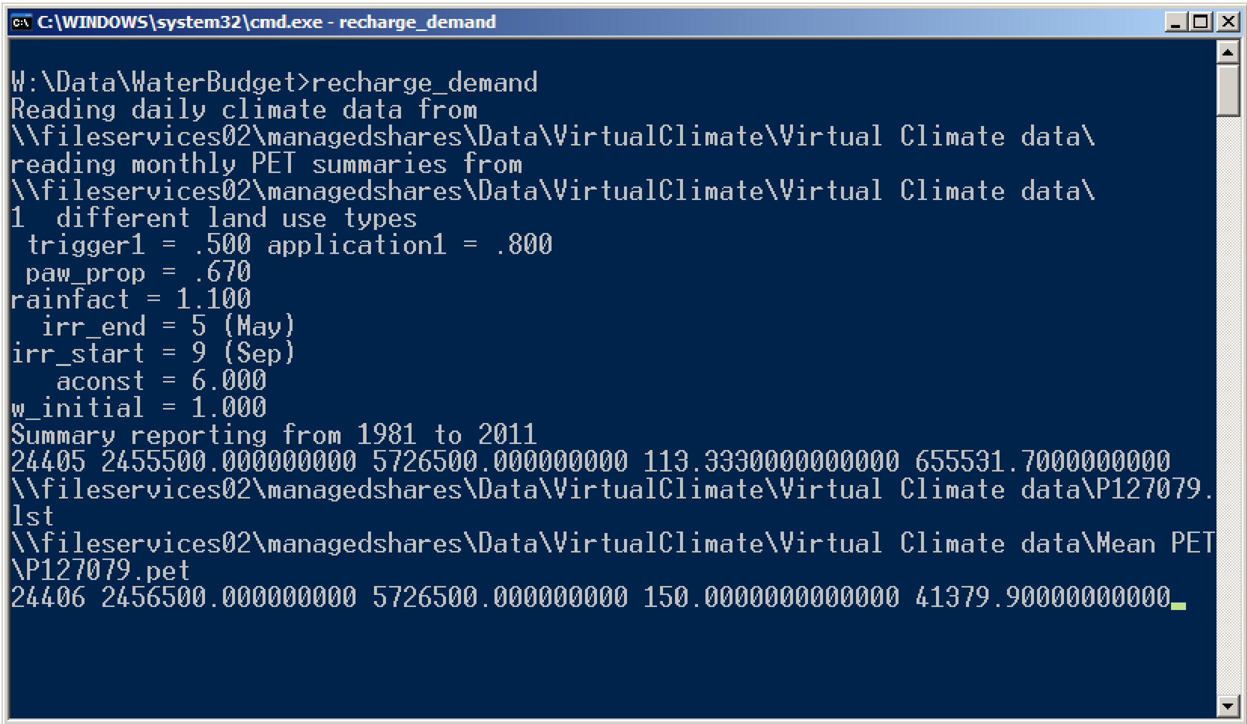

Figure 3: Screen shot of command prompt execution of recharge_demand

Initialization file (recharge_demand.ini)

The initialization file allows the user to specify the following items:

- the name of the primary climate data folder
- the name of the folder containing supplementary potential evapotranspiration data
- **the number of different land use types and the parameters pertaining to those land uses**
- the water holding capacity as a proportion of profile available water (PAW)
- a ground-level gauge correction
- the duration of the irrigation season
- the constant A (curve number) to be used in the AET/PET vs W/WHC relationship
- the initial soil water condition
- specification of a period for summary reporting.

If the initialization file does not exist prior to running the **recharge_demand** program, the program will create a new version of the file using default values (as shown in Figure 4).

```
name of data_directory
name of pet_directory
  1 number of landuse types
  0.0 array of irrigation trigger values
  0.8 array of irrigation application parameters
  0.67 proportion of PAW value to be used as water holding capacity
  1.1 ground-level gauge correction
  5 month following end of irrigation season
  9 month preceeding start of irrigation season
  6.0 constant in AET/PET vs W/WHC relationship
  1.0 initial soil moisture level as proportion of WHC
  1981 30 initial year and no of years for summary reporting
```

Figure 4: Default initialization file

In most circumstances, many of these default values are likely to be satisfactory – and the same will apply for a pre-existing initialization file that has been used in a previous analysis. The exceptions (highlighted in bold font) are as follows:

- the 1st line 'name of data_directory' must be replaced with the following character string \\fileservices02\managedshares\Data\VirtualClimate\Virtual Climate data\
- the 2nd line 'name of pet_directory' must be replaced with the following character string \\fileservices02\managedshares\Data\VirtualClimate\Virtual Climate data\Mean PET\
- lines 3 to 5 relating to land use must be set to match the particular circumstances.

The term 'landuse types' is used to refer to different types of irrigation (including the option of no irrigation. The program requires that at least one landuse type is specified and accommodates a maximum of 3 – hence the only valid values for that item are 1, 2 or 3.

An irrigation trigger value must be defined for each landuse type as follows:

- a zero trigger value indicates that the landuse type is dryland (i.e. no irrigation)
- the absolute value of a non-zero trigger value is used to specify the soil moisture level as the proportion of water holding capacity below which irrigation will be triggered (i.e. the trigger value should be in the range -1.0 to 1.0). A value of 0.5 has typically been assumed to represent spray irrigation.
- a positive trigger value indicates that the quantum of irrigation will be dependent on the soil moisture deficit as determined by the corresponding irrigation application parameter
- a negative trigger value indicates that the quantum of irrigation will be fixed by the corresponding landuse type irrigation application parameter.

An irrigation application parameter must also be defined for each landuse type:

- if the corresponding trigger value is zero, the application factor is irrelevant and is not used. Nevertheless, a value must be included
- if the corresponding trigger value is positive, the quantum of irrigation is set as the current soil moisture deficit divided by the application parameter. Hence, an application factor less than 1.0

results in a surplus (and resulting drainage) whereas a value greater than 1.0 will leave the soil moisture in deficit (deficit irrigation).

- if the corresponding trigger value is negative, the quantum of irrigation is set equal to the application parameter. Hence, in this case, the application factor is a depth.

The following example will, hopefully, clarify the above definitions. If an initialization file contains the following items in lines 3 to 5:

3		
0	0.5	-0.6
0	0.8	100

then three landuse types are specified as follows:

- landuse type 1
  - dryland (no irrigation, application parameter is ignored)
- landuse type 2
  - irrigation will be triggered when soil moisture deficit reaches 0.5 of water holding capacity and (because trigger > 0) the quantum of irrigation will be equal to the soil moisture deficit/0.8
- landuse type 3
  - irrigation will be triggered when soil moisture deficit reaches 0.6 of water holding capacity and (because trigger < 0) the quantum of irrigation will be 100.

The remaining items in the initialization file are as follows:

- line 6 specified the relationship between water holding capacity and Profile Available Water
  - a value of 0.67 has typically been adopted to reflect an effective root zone depth that is 2/3^{rds} of the depth over which PAW has been assessed
- line 7 specifies a constant to be used as a ground-level gauge correction
  - a value of 1.1 applies a 10% increase to standard raingauge values
- line 8 specifies the month following the end of the irrigation season
  - a value of 5 indicates that the irrigation season ends at the end of April
- line 9 specifies the month preceeding the start of the irrigation season
  - a value of 9 indicates that the irrigation season starts at the beginning of October
- line 10 is the constant (A) in the AET/PET vs W/WHC relationship
  - a value of 6.0 has generally been adopted
- line 11 specifies initial soil moisture level as a proportion of WHC
  - a value of 1.0 indicates that initial soil moisture levels are equal to WHC (i.e. are fully saturated)
- line 12 specifies the initial year and the number of years for which average values are to be reported
  - values of 1981 30 indicate that 30 year averages from 1981 will be reported.

Gr id cell properties file (grid_cell.prn)

The grid cell properties file contains one line per sub-area specifying the following items:

- an integer cell identification number
- the x coordinate of the cell
- the y coordinate of the cell
- the profile available water for the cell (mm)
- the surface areas of the landuse types for the cell (m^2).

The x and y coordinates must be NZMG and are conventionally established at cell centroids. Regardless of how many landuse types are represented in any individual cell the areas must be included for as many landuse types as indicated in the initialization file.

Output

While running, the program continues to report progress as illustrated in Figure 3, echoing the input line from the grid_cell.prn file and identifying the relevant VCN files. On completion, the program also displays the following items for the reporting period as specified in the initialization file:

- total_area (m^2)
- rainfall (mm/yr)
- PET (mm/yr)
- AET (mm/yr)
- Demand (mm/yr)
- Drainage (mm/yr)

The program writes a single output file (recharge_demand.csv) containing the following items (in comma separated variable format to facilitate further analysis and plotting using Microsoft Excel):

- date (yyyymm format)
- rain (total rainfall for the month)
- pet (total potential evapotranspiration for the month)
- aet (total actual evapotranspiration for the month)
- demand (total irrigation demand for the month)
- drain (total drainage for the month).

All of the above monthly totals are reported in terms of depths (in mm). The aet, demand and drain values are spatially weighted sums which take account of the landuse type areas specified in the grid_cell.prn input file.

Running the program

The recharge_demand program uses fixed names for all input and output files. This has been done to simplify the programming task. It does, however, create potential complications for the user.

For a single project area the user should copy and appropriately rename the input files for subsequent analysis and, if needed archiving.

In the event that a series of project areas has to be analysed at the same time it is possible to use batch file commands to expedite matters, as illustrated below for the hypothetical situation where the input files are:

- for case1 are case1.ini and case1.prn
- for case2 are case2.ini and case2.prn
- for case3 are case3.ini and case3.prn

A batch file to run the program for these 3 cases and at the same time do the necessary file copying could contain the following lines:

```
copy case1.ini recharge_demand.ini
copy case1.prn grid_cell.prn
recharge_demand > output1.txt
copy recharge_demand.csv case1.csv
copy case2.ini recharge_demand.ini
copy case2.prn grid_cell.prn
recharge_demand > output2.txt
copy recharge_demand.csv case2.csv
copy case3.ini recharge_demand.ini
copy case3.prn grid_cell.prn
recharge_demand > output3.txt
copy recharge_demand.csv case3.csv
```

Running the above batch file would result in the creation of csv and txt files for each case. NB the ">" symbol (pipe out) indicates that output that would normally be displayed to the screen will, instead, be written to the named file. This would allow the user to leave the job running in batch mode and inspect the text outputs for any error messages or to record the summary data.

InExamples

Single cell, single landuse type

Multiple cells, multiple landuse types

References

Hong Y-S, PA White and DM Scott (2005): Automatic rainfall recharge model induction by evolutionary computational intelligence. Water Resources Research, 41, W08422, doi:10.1029/2004WR003577.

Minhas BS, KW Parikh, and TN Srinavasan (1974): Toward the structure of a production function for wheat yields with dated input of irrigation water. Water Resources Research, Vol 10, No 3, pp 383-393.

Rushton KR, VHM Eilers, RC Carter (2006): Improved soil moisture balance methodology for recharge estimation. Journal of Hydrology 318, 379-399.

Scott DM (2004): Groundwater allocation limits: land-based recharge estimates. Environment Canterbury Report No. U04/97.

Thorley MJ, VJ Bidwell and DM Scott (2010): Land-surface recharge and groundwater dynamics – Rakaia-Ashburton Plains. Environment Canterbury Report No. R09/55.

Thorley M, M Ettema (2007): Review of water allocation limits for the south Canterbury downlands. Unpublished Environment Canterbury technical report No. U07/09.

Thorpe, HR and DM Scott (1999): An evaluation of four soil moisture models for estimating natural ground water recharge. Journal of Hydrology (NZ), Vol. 38, No. 2, pp. 179-209.

White PA, Y-S Hong, DL Murray, DM Scott and HR Thorpe (2003): Evaluation of regional models of rainfall recharge to groundwater by comparison with lysimeter measurements, Canterbury, New Zealand. Journal of Hydrology (NZ), 42(1), 39-64.

Williams H (2010): Groundwater resources in the Te Waihora/Lake Ellesmere catchment: management issues and options. Environment Canterbury Report No. R10/05.

Appendices

Appendix A: Fortran code – Recharge_demand.for

```
c-------PROGRAM RECHARGE_DEMAND
c      ************************************************************************
c      This program is an adaptation of the soil moisture budget method that was
c      applied to the calculation of the land-surface recharge figures used to
c      establish interim allocation limits for groundwater allocation zones
c     (as described in ECan report U04/97).
c
c      It has been adapted to be used for the periodic update of monthly recharge
c      totals and assumes the following:
c      - the program will be run within a folder containing the following files:
c            recharge_demand.ini (contains program initialisation data)
c            grid_cell.prn (defines the soil property, location and sub-area for grid
c            cells within the zone of interest)
c      - daily rainfall and pet will be read from files stored in the primary folder
c        specified in the initialisation file
c      - location of the NIWA virtual climate sites will be read from coords.prn
c        (in the same folder as the daily climate data)
c      - for the period prior to 1972 daily pet will be estimated from the average
c        monthly totals calculated for 1972 to July 2007 stored in the secondary
c        folder specified in the initialisation file
c
c      David Scott
c      December 2013
c      ************************************************************************
c      SPECIFICATIONS
c      ------------------------------------------------------------------------
       implicit real*8 (a-h,o-z)
       dimension     rain_t(1200), pet_t(1200), aet_t(1200),
      *              drain_t(1200), demand_t(1200),
      *              trigger(3), application(3), area(3), w(3)
       character*132 line, data_directory, pet_directory, file_in, header
       character*7   climate
       character*3   month_chr(12)
       integer       year, month, day, year_old, month_old,
      *              month_num(12), year0
       real*8        lat, long, month_pet(12)
       integer*4     date, dates(1200)
       logical       irrig, exists, first

       data month_num/31,28,31,30,31,30,31,31,30,31,30,31/
       data month_chr/'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Se
      *p','Oct','Nov','Dec'/

       nmax = 1200

c.......read initialisation file details
       inquire(file='recharge_demand.ini',exist=exists)
       if( .not. exists) then
          write(*,*) ' recharge_demand.ini file did not exist'
          write(*,*) ' add details to newly created file and rerun'
          open(10,'recharge_demand.ini')
          write(10,*) ' name of data_directory'
          write(10,*) ' name of pet_directory'
          write(10,*) ' 1 number of landuse types'
          write(10,*) ' 0.0 array of irrigation trigger values'
```

```
      write(10,*) ' 0.8 array of irrigation application parameters'
      write(10,*) ' 0.67 proportion of PAW value to be used as water
     *holding capacity'
      write(10,*) ' 1.1 ground-level gauge correction'
      write(10,*) ' 5 month following end of irrigation season'
      write(10,*) ' 9 month preceeding start of irrigation season'
      write(10,*) ' 6.0 constant in AET/PET vs W/WHC relationship'
      write(10,*) ' 1.0 initial soil moisture level as proportion of
     *WHC'
      write(10,*) ' 1981 30 initial year and no of years for summary
     *reporting'
      go to 99
   endif

   open(10,'recharge_demand.ini')
   read(10,1,err=90) data_directory
 1 format(a90)
   file_in = charnb(data_directory)
   inquire(file=file_in,exist=exists)
   if( .not. exists) then
      write(*,*) 'Specified data directory does not exist'
      go to 99
   endif
   write(*,*) 'Reading daily climate data from '
   write(*,*) charnb(data_directory)

   file_in = charnb(data_directory)//'coords.prn'
   inquire(file=file_in,exist=exists)
   if( .not. exists) then
      write(*,*) 'coords.prn file does not exist in specified data di
     *rectory'
      go to 99
   endif

   read(10,1,end=90) pet_directory
   file_in = charnb(pet_directory)
   inquire(file=file_in,exist=exists)
   if( .not. exists) then
      write(*,*) 'Specified PET directory does not exist'
      go to 99
   endif
   write(*,*) 'Reading monthly PET summaries from '
   write(*,*) charnb(data_directory)

   read(10,*,err=90) n_landuse     ! number of different land use types (max=3)
   if(n_landuse .gt. 3) then
      write(*,*) ' Number of different land use types > 3'
      go to 99
   endif
   write(*,*) n_landuse, ' different land use types'
   read(10,*,err=90) (trigger(i),i=1,n_landuse)
   read(10,*,err=90) (application(i),i=1,n_landuse)
   do i = 1,n_landuse
      write(*,2) i, trigger(i), i, application(i)
 2    format('  trigger',i0,' = ',f0.3,' application',i0,' = ',f0.3)
   end do
   read(10,*,err=90) paw_prop      ! proportion of PAW as water holding capacity
   write(*,3) paw_prop
 3 format('  paw_prop = ',f0.3)
   read(10,*,err=90) rainfact      ! ground-level gauge correction
   write(*,4) rainfact
 4 format('  rainfact = ', f0.3)
```

```
      read(10,* ,err=90) irr_end      ! month following end of irrigation season
      write(*,5) irr_end, month_chr(irr_end)
    5 format('   irr_end = ',i0,' (',a,')')
      read(10,*,err=90) irr_start     ! month preceeding start of irrigation season
      write(*,6) irr_start, month_chr(irr_start)
    6 format(' irr_start = ',i0,' (',a,')')
      read(10,*,err=90) aconst               ! constant in AET/PET vs W/WHC relationship
      write(*,7) aconst
    7 format('    aconst = ',f0.3)
      read(10,*,err=90) w_initial     ! initial soil moisture as proportion of WHC
      write(*,8) w_initial
    8 format(' w_initial = ',f0.3)
      read(10,*,err=90) year0, nyrs   ! initial year and no of years for summary report
      write(*,9) year0, year0 + nyrs
    9 format(' Summary reporting from ',i0,' to ',i0)
      close(10)

c........clear monthly arrays and accumulators
      do i = 1,nmax
         rain_t(i)   = 0.0
         pet_t(i)    = 0.0
         aet_t(i)    = 0.0
         drain_t(i)  = 0.0
         demand_t(i) = 0.0
      end do
      total_vol_rain   = 0.0
      total_vol_pet    = 0.0
      total_vol_aet    = 0.0
      total_vol_demand = 0.0
      total_vol_drain  = 0.0
      mnth_max         = 0

c........open grid_cell.prn
      inquire(file='grid_cell.prn')
      if( .not. exists) then
         write(*,*) ' grid_cell.prn file does not exist in current direc
     *tory'
         go to 99
      endif
      open(10,'grid_cell.prn')

      total_area = 0.0

c.......read the grid cell details
   10 read(10,*,err=91,end=40) id, x0, y0, paw, (area(i),i=1,n_landuse)
      write(*,*) id, x0, y0, paw, (area(i),i=1,n_landuse)
      paw        = paw_prop*paw              ! apply PAW proportion
      areasum = 0.0
      do i = 1,n_landuse
         areasum = areasum + area(i)
      end do
      total_area = total_area + areasum

c........identify closest synthetic climate station (file 11)
      j0 = 0
      k0 = 0
      open(11,'coords.prn')
      distance = 1e12
   11 read(11,*,end=20,err=92) x, y, j, k
      write(climate,21) j, k
      file_in = charnb(data_directory)//charnb(climate)//'.lst'
      inquire(file=file_in,exist=exists)
```

```
      if( .not. exists ) go to 11
      r = ((x - x0)/1000.0)**2 + ((y - y0)/1000.0)**2
      if(r .lt. distance) then
         distance = r
         j0       = j
         k0       = k
      endif
      go to 11
   20 close(11)
      if(j0 .eq. 0 .or. k0 .eq. 0) then              ! no station identified
         write(*,*) 'no station identified for ', id, x0, y0
         write(*,*) 'NB: Grid cell locations must be specified in NZMG'
         go to 99
      endif

c........open climate station record (file 12) and corresponding average pet (file 13)
      write(climate,21) j0, k0
   21 format('P',2i3.3)
      file_in = charnb(data_directory)//charnb(climate)//'.lst'
      write(*,*) file_in
      open(12,charnb(file_in),status='old')
      file_in = charnb(pet_directory)//charnb(climate)//'.pet'
      write(*,*) file_in
      inquire(file=file_in,exist=exists)
      if( .not. exists ) then
          write(*,*) ' no PET summary file for ', charnb(climate)
          go to 99
      endif
      open(13,charnb(file_in),status='old')

c........read average monthly pet values - to use prior to 1972
      do i = 1,12
         read(13,*,err=93) month_pet(i)
      end do

c.........initialise parameters
      do i = 1,n_landuse
         w(i) = w_initial*paw
      end do

c........step through daily climate record
      mnth  = 1
      first = .true.
   31 read(12,32,end=35,err=94) line
   32 format(a)
      read(line(11:80),*) lat, long, date
      if( first ) then
          year_old  = 0
          month_old = 0
          first     = .false.
      endif
      year  = date/10000
      month = (date - 10000*year)/100
      day   = date - 10000*year - 100*month

      if(year .lt. 1972) then
         read(line(11:80),*) lat, long, date, rain   ! no pet data for pre-1972
         if(month .eq. 2 .and. mod(year,4) .eq. 0) then     ! adjust for leap-year
            pet = month_pet(month)/29
         else
            pet = month_pet(month)/month_num(month)
         endif
```

```
      else
         read(line(11:80),*) lat, long, date, rain, pet
      endif

      rain = rainfact*rain                    ! apply ground-level gauge correction
      if(pet .lt. 0.0) then
         pet = 0.0                            ! correct negative pet estimates
      endif

      if(month .ge. irr_end .and. month .le. irr_start) then
         irrig = .false.                      ! outside irrigation season
      else
         irrig = .true.
      endif

      if(day .eq. 1) then                             ! start a new month
         dates(mnth) = 100*year_old + month_old              ! store year/month date
         year_old    = year
         month_old   = month
         mnth        = mnth + 1
         if(mnth .gt. nmax) then
            write(*,*) 'month count exceeds ', nmax
            go to 99
         endif
      endif

c........step through specified land use types
      aet_tot    = 0.0
      demand_tot = 0.0
      drain_tot  = 0.0
      do i = 1,n_landuse
         appl = 0.0                                   ! default condition (dryland)
         if(trigger(i) .gt. 0.0) then                 ! deficit dependent irrigation
            if( irrig ) then
               if(w(i) .lt. trigger(i)*paw) then
                  appl = (paw - w(i))/application(i) ! apply deficit/efficiency
               endif
            endif
         elseif(trigger(i) .lt. 0.0) then             ! fixed quantum irrigation
            if( irrig ) then
               if(w(i) .lt. -trigger(i)*paw) then
                  appl = application(i)                      ! apply specified depth
               endif
            endif
         endif
         call budget(aconst,paw,rain,pet,aet,w(i),appl,drainage)
         aet_t(mnth)    = aet_t(mnth) + aet*area(i)/1e3
         demand_t(mnth) = demand_t(mnth) + appl*area(i)/1e3
         drain_t(mnth)  = drain_t(mnth) + drainage*area(i)/1e3
         aet_tot        = aet_tot + aet*area(i)/1e3
         demand_tot     = demand_tot + appl*area(i)/1e3
         drain_tot      = drain_tot + drainage*area(i)/1e3
      end do
      rain_t(mnth) = rain_t(mnth) + rain*areasum/1e3
      pet_t(mnth)  = pet_t(mnth) + pet*areasum/1e3

c.......accumulate totals for specified standard period (year0 for nyrs)
      if(year .ge. year0 .and. year .lt. year0 + nyrs) then
         total_vol_rain   = total_vol_rain + rain*areasum/1E3
         total_vol_pet    = total_vol_pet + pet*areasum/1E3
         total_vol_aet    = total_vol_aet + aet_tot
         total_vol_demand = total_vol_demand + demand_tot
```

```
         total_vol_drain  = total_vol_drain + drain_tot
      endif

      go to 31

c.......end of record, close files and get next grid cell
   35 close(12)
      close(13)
      dates(mnth) = 100*year_old + month_old
      if(mnth .gt. mnth_max) then
         mnth_max = mnth
      endif
      go to 10

c.......write the monthly recharge series - cubic metres converted to mm
   40 write(*,*) 'total area = ', total_area
      open(14,'recharge_demand.csv')

      write(14,*) 'date,rain,pet,aet,demand,drain'
      A = 1e3/total_area
      do i = 2,mnth_max
         write(14,41) dates(i), A*rain_t(i), A*pet_t(i), A*aet_t(i),
     *                A*demand_t(i), A*drain_t(i)
   41    format(i6,9(',',f8.2))
      end do

c........write accumulated volumes for specified period (as mm/year)
      A = 1e3/total_area/real(nyrs)
      write(*,*) ' Summary for the ', nyrs,' yr period from ', year0
      write(*,*) '   Rainfall: ', A*total_vol_rain,' (mm/yr)'
      write(*,*) '        PET: ', A*total_vol_pet,' (mm/yr)'
      write(*,*) '        AET: ', A*total_vol_aet,' (mm/yr)'
      write(*,*) '     Demand: ', A*total_vol_demand,' (mm/yr)'
      write(*,*) '   Drainage: ', A*total_vol_drain,' (mm/yr)'
      go to 99

   90 write(*,*) ' error reading recharge_demand.ini file'
      go to 99
   91 write(*,*) ' error reading grid cell details'
      go to 99
   92 write(*,*) ' error reading coords.prn'
      go to 99
   93 write(*,*) ' error reading monthly pet values'
      go to 99
   94 write(*,*) ' error reading daily climate record'
   99 stop
      end

c-------SUBROUTINE BUDGET
      subroutine budget(aconst,paw,rain,pet,aet,w,appl,drainage)
c     ***********************************************************************
c     Apply soil moisture balance calculation for single time-step
c     Returns AET, updated soil moisture level and drainage
c     ***********************************************************************
c     SPECIFICATIONS:
c     -----------------------------------------------------------------------
      implicit real*8 (a-h,o-z)

      smfac    = (1.0 - exp(-aconst*w/paw))/
     *             (1.0 - 2.0*exp(-aconst) + exp(-aconst*w/paw))
      aet      = smfac*pet
      w        = w + rain + appl - aet
```

```
drainage = 0.0
if(w .gt. paw) then
   drainage = w - paw
   w        = paw
elseif(w .lt. 0.0) then
   w = 0.0
endif

return
end
```

Appendix B: Modifications of the simple model to represent limited infiltration capacity

The soil moisture budget calculations that have been used to estimate recharge on the Canterbury Plains have been based on the assumption that runoff can be neglected. This assumption is less appropriate for soil infiltration capacity is limited – as in some areas of South Canterbury where there are significant areas with loess soil. The sub-catchment water budget calculations for the South Canterbury Coastal Streams area have allowed for runoff by introducing a term that defines a maximum drainage rate: runoff is assumed to occur when drainage reaches that maximum rate. The highlighted lines in the text box below show the changes to the Fortran code that were made to implement this approach.

```
      subroutine budget(paw,rain,pet,aet,w,appl,drainage,
     *rate_max,runoff)
c     this version (1) limits the drainage from the soil unit to rate_max
      implicit real*8 (a-h,o-z)
      a        = 6.0          ! constant in aet/pet vs w/paw relationship
      smfac    = (1.0 - exp(-a*w/paw))/
     *           (1.0 - 2.0*exp(-a) + exp(-a*w/paw))
      aet      = smfac*pet
      w        = w + rain + appl - aet
      drainage = 0.0
      runoff   = 0.0
      if(w .gt. paw) then
         drainage = w - paw
         if(drainage .gt. rate_max) then
            runoff   = drainage - rate_max
            drainage = rate_max
         endif
         w = paw
      elseif(w .lt. 0.0) then
         w = 0.0
      endif
      return
      end
```