```
$ONSYMLIST OFFSYMXREF  onlisting
 OPTION LIMROW = 0
 OPTION LIMCOL = 0
 option iterlim =1500
$EOLCOM //

*#####################################################################################
* PMP MODEL WITH THEORETICALLY CORRECT CROP ET AND YIELD FUNCTIONS FROM AG ENGNR LIT
* INFERS IRRIGATION EFFICIENCY BASED ON OBSERVATIONS ON:
*    YIELD, WATER USE, ACREAGE, INPUT AND OUTPUT PRICES, CROPPING PATTERNS

* IRRIGATION EFFICIENCY CALIBRATION IS BASED ON 1ST ORDER CONDITIONS FOR PROFIT MAX
* IRRIGATION EFF IS CALCULATED TO MAKE OBSERVED VMP = PRICE OF INPUTS (LAND AND WATER)
* CALIBRATED IRR EFF IS USED TO IDENTIFY THE CROP WATER PRODUCTION FUNCTION
* RESULT IS SELF-CALIBRATED: INCOME OPTIMIZED CW PRODN FN OUTCOMES MATCH OBSERVED DATA
*######################################################################################
* FRANK A WARD DECEMBER 2007
* NEW MEXICO STATE UNIVERSITY
* DEPARTMENT OF AGRICULTURAL ECONOMICS AG BUSINESS
* FWARD@NMSU.EDU
*######################################################################################

 SETS  I  PRODUCTION PROCESSES /COT_FLOOD, COT_DRIP /
       P  POLICY               /OLD, NEW /

*##################################################################
*     DATA
*##################################################################

parameter  pwater_p(p)   price of water per acre inch
/old            435
 new            435/

parameter  pland_p(p)    price of land per acre
/old            400
 new            400/

table  price_p(I,p)   commodity price
                   old         new
COT_FLOOD          10          10
COT_DRIP           10          10

parameter  Yd_p(I  )      dryland yield
/
COT_FLOOD          200
COT_DRIP           200
/

parameter  Ym_p(I  )      yield under max irrigation for first acre
/
COT_FLOOD           800
COT_DRIP           1000
/

parameter   Im_p(I)     irrigation required for max yield in ac-in per acre

/
COT_FLOOD           24
COT_DRIP            18
/

table   rain_p(I,p)   inches per acre of rain
                    old        new
COT_FLOOD           8.0        8.0
```

```
COT_DRIP             8.0        8.0


table     b_rain_p(I,p)  rain efficiency
                    old         new
COT_FLOOD           0.5         0.5
COT_DRIP            0.5         0.5

parameter  yield_p(I)   observed yield under actual (typically deficit) irrigation

/
COT_FLOOD         260.7900
COT_DRIP          602.1419
/

parameter

divert_rhs_p(p)       water supply available across crops and acres
/old        400
 new        400/

land_rhs_p(p)        land supply in acres
/old         40
 new         40/

parameter land_p(I)  observed land in production under base conditions - used to calibrate

/
cot_flood     17.0140
cot_drip      19.1535
/

parameter I_p(I)     observed (assumed optimized) irrigation rate ac-in per acre

/cot_flood     3.1200
 cot_drip     10.7212   /

parameter  water_p(I)   observed total water in production used to calibrate
;
water_p(I)          = land_p(I) * I_p(I);

* pmp calibration for quadratic yield function theoretically correct ET
parameter

s1_p   (I)    max yield minus dryland yield
n1_p   (I)    intermediate
d1_p   (I)    intermediate
n2_p   (I)    intermediate
d2_p   (I)    intermediate
n3_p   (I)    intermediate
d3_p   (I)    intermediate

num1_p (I)    intermediate
den1_p (I)    intermediate
num2_p (I)    intermediate
den2_p (I)    intermediate

a1_p   (I)    calibrated (indirectly observed) reciprocal of irrigation efficiency based on prof max behavior (p = vmp)
a2_p   (I)    1 - calibrated reciprocal of irr eff

blan_p (I)    intermediate
bland_p(I)    calibrated (indirectly observed) effect of increased land on crop yield (lower quality land with more land in prodn)

b_irr_p(I  )  calculated irrigation efficiency from optimized behavior
```

```
ETd_p  (I,p)  inferred ET dryland - a linear a function of dryland yield
ETm_p  (I,p)  inferred ET max     - a linear function of max yield
;

* calculates irrigation efficiency and related
s1_p   (I) = Ym_p    (I      ) - Yd_p(I      )                                               ;
n1_p   (I) = Land_p  (I      ) * Im_p(I      ) * s1_p(I) - 2 * water_p(I      )*s1_p(I)      ;
d1_p   (I) = Land_p  (I      ) * Im_p(I      ) ** 2                                          ;
n2_p   (I) = pwater_p(  'old')                                                               ;
d2_p   (I) = price_p (I,'old')                                                               ;
n3_p   (I) = 2 * water_p(I      ) * s1_p(I)                                                  ;
d3_p   (I) = Land_p  (I      ) * Im_p(I      )**2;                                           ;

a1_p   (I) = {[n2_p(I) / d2_p(I)] - [n3_p(I) / d3_p(I)]} / [n1_p(I)/d1_p(I)]                 ;  // inferred reciprocal of irr eff
a2_p   (I) = 1 - a1_p(I)                                                                     ;
b_irr_p(I) = 1/a1_p(I)                                                                       ;  // calc irrigation efficiency

ETd_p(I,p) =  b_rain_p(I,p) * rain_p(I,p)                                                    ;  // ET dryland (without irrigation)
ETm_p(I,p) =  b_rain_p(I,p) * rain_p(I,p) + b_irr_p(I) * Im_p(I  )                           ;  // ET under max irrigation

* calculates marginal impact of land on yield
num1_p (I) = a1_p   (I      ) * water_p (I      )                                            ;
den1_p (I) = land_p (I      ) * Im_p    (I      )                                            ;
num2_p (I) = 2*a2_p (I      ) * water_p (I      )**2                                         ;
den2_p (I) = (land_p(I      )**2) * Im_p(I      )**2                                         ;

blan_p (I) = pland_p('old') - price_p(I,'old') * yield_p(I      )
           + price_p(I,'old') * (s1_p(I) * [(num1_p(I)/den1_p(I)) +  (num2_p(I)/den2_p(I))]) ;

bland_p(I) = blan_p(I) / (price_p(I,'old')  * land_p(I      ))                               ;   // inferred impact of more land on yield

display ETd_p, ETm_p, a1_p, a2_p, b_irr_p, bland_p;

*variables
positive variables

divert_v  (  p) stream diversions (total water diverted)
land_v    (I,p) land in production (total land)

VARIABLES

*technical
Y_v       (I,p) yield under actual (deficit) irrigation
Yu_v      (I,p) yield unstressed
Yls_v     (I,p) yield loss from stress

I_v       (I,p) actual irrigation in inches per acre
ET_v      (I,p) actual ET
Ir_v      (I,p) relative irrigation
F1a_v     (I,p) theoretical yield response function
F1_v      (I,p) yield response ratio function (0 to 1) upper bound of 1
ET_v      (I,p) actual ET in inches per acre

*economic
rev_a_v   (I,p) revenue per acre and policy
cost_a_v  (I,p) cost per acre    and policy
income_a_v(I,p) income per acre  and policy

rev_v     (I,p) total revenue by crop and policy
cost_v    (I,p) total cost    by crop and policy
income_v  (I,p) total income  by crop and policy

trev_v    (  p) total revenue by policy
tcost_v   (  p) total cost    by policy
```

```
tincome_v (  p) total income  by policy

income_old_v    total income old policy
income_new_v    total income new policy
;

EQUATIONS

*technical
Ir_e      (I,p) relative irrigation
ET_e      (I,p) actual ET
F1a_e     (I,p) theoretical yield response ratio
F1_e      (I,p) 0 to 1 yield response ratio with upper bound of 1
Y_e       (I,p) actual yield

divert_e  (  p) total stream diversion
div_bd_e  (  p) upper boundon stream diversion

Land_e    (  p) land supply
income_e        income old and new policy

*economic
rev_a_e   (I,p) revenue per acre
rev_e     (I,p) total revenue
trev_e    (  p) total revenue over crops
cost_a_e  (I,p) cost per acre
cost_e    (I,p) total cost
tcost_e   (  p) total cost over crops
income_a_e(I,p) income per acre
income_e  (I,p) total income
tincome_e (  p) total income by policy

income_old_e    income obj old policy
income_new_e    income obj new policy
;

*model variables and equations

Ir_e      (I,p)..  Ir_v (I,p) =e= I_v  (I,p)  /  Im_p(I  )                                ;  // relative irrigation per ac (0-1)
ET_e      (I,p)..  ET_v (I,p) =e= ETd_p(I,p) + (ETm_p(I,p) - ETd_p(I,p)) * F1_v(I,p)  ;  // absolute ET per acre as a fn of yld resp ratio fn

Y_e       (I,p)..  Y_v  (I,p) =e= Yd_p(I  )
                            +   ((Ym_p(I  )   - Yd_p(I  )) * F1_v(I,p))
                            +  bland_p(I  ) * Land_v(I,p)                             ;  // yield per acre based on F1 and acreage

F1a_e     (I,p)..  F1a_v(I,p)  =e= a1_p(I) * Ir_v(I,p) + (1-a1_p(I)) * (Ir_v(I,p) ** 2);  //quadratic yield response ratio fn (0-1) scales ET and Yield
F1_e      (I,p)..  F1_v (I,p)  =e=          F1a_v(I,p)                                 ;  // upper bound on F is 1, at which point it becomes horizontal;

divert_e  (  p)..  sum(I,   I_v(I,p) * land_v(I,p))   =e= divert_v    (p)          ;  // water diverted over acres
div_bd_e  (  p)..      divert_v(  p)                  =l= divert_rhs_p(p)          ;  // water diverted < water supply
Land_e    (  p).. sum(I, Land_v(I,p)) =l= land_rhs_p(p)                            ;  // land use < land supply

rev_a_e   (I,p)..  rev_a_v     (I,p)  =e=  price_p (I,p) * Y_v(I,p)                ;  // revenue per acre
cost_a_e  (I,p)..  cost_a_v    (I,p)  =e=  pwater_p(  p) * I_v(I,p) + pland_p(p)   ;  // cost per acre
income_a_e(I,p)..  income_a_v  (I,p)  =e=  rev_a_v (I,p) - cost_a_v(I,p)           ;  // profit per acre

rev_e     (I,p)..       rev_v  (I,p)  =e=  rev_a_v (I,p)  * land_v(I,p)            ;  // total revenue by crop and policy
cost_e    (I,p)..      cost_v  (I,p)  =e=  cost_a_v(I,p) * land_v (I,p)             ;  // total cost by crop and policy
income_e  (I,p)..    income_v  (I,p)  =e=  rev_v   (I,p) - cost_v (I,p)               ;  // total income by crop and policy

trev_e    (  p)..        trev_v(  p)  =e=  sum(I,    rev_v(I,p))                   ;  // total revenue by policy
tcost_e   (  p)..       tcost_v(  p)  =e=  sum(I,   cost_v(I,p))                   ;  // total cost by policy
tincome_e (  p)..     tincome_v(  p)  =e=  sum(I, income_v(I,p))                   ;  // total income by policy
```

```
income_old_e..     income_old_v       =e=   tincome_v('old')                     ;  // total income old policy
income_new_e..     income_new_v       =e=   tincome_v('new')                     ;  // total income new policy

model pmp_quad /all/;
options nlp = conopt;

Ir_v.lo    (I,p)  = 0.00;  // zero lower bound on relative irrigation
Ir_v.up    (I,p)  = 1.00;  // unity upper bound on relative irrigation: yield is discontinous at 1
Land_v.lo  (I,p)  = 0.01;  // excludes 0 from denominator

solve pmp_quad using nlp maximizing income_old_v;

F1_v.fx(i,'old') = F1_v.l(I,'old');

solve pmp_quad using nlp maximizing income_new_v;

* post optimiality calcs based on optimal solution above
parameter

Yu_p       (I,p)  unstressed (full water) yield
Yls_p      (I,p)  yield loss from water stress
sig1_p     (I,p)  max yield minus dryland yield

numL1_p    (I,p)  int
denL1_p    (I,p)  int
numL2_p    (I,p)  int
denL2_p    (I,p)  int
mpp_land_p (I,p)  mpp of land
vmp_land_p (I,p)  vmp of land

numw1_p    (I,p)  int
denw1_p    (I,p)  int
numw2_p    (I,p)  int
denw2_p    (I,p)  int
mpp_water_p(I,p)  mpp water (guard: only accurate at less than full irrigation)
vmp_water_p(I,p)  vmp water (guard: only accurate at less than full irrigation)
;

Yu_p       (I,p)  =   Yd_p(I)
                  + ((Ym_p(I)  - Yd_p(I)))
                  + bland_p(I)   * Land_v.l(I,p)                    ;  // unstressed yield

Yls_p      (I,p)  = Yu_p(I,p) - Y_v.l(I,p)                          ;  // yield loss from crop water stress

sig1_p     (I,p)  =  Ym_p(I) - Yd_p(I)                              ;
numL1_p    (I,p)  = -a1_p(I) * I_v.l(I,p)                           ;
denL1_p    (I,p)  =  Im_p(I)                                        ;
numL2_p    (I,p)  = -2*(1-a1_p(I))  * I_v.l(I,p)**2                 ;
denL2_p    (I,p)  =  Im_p(I)**2                                     ;

numw1_p    (I,p)  =  a1_p(I)                                        ;
denw1_p    (I,p)  =  Im_p(I)                                        ;
numw2_p    (I,p)  =  (1-a1_p(I)) * 2 * (I_v.l(I,p) * land_v.l(I,p));
denw2_p    (I,p)  =  Land_v.l(I,p) * (Im_p(I  ) **2)                ;

mpp_land_p (I,p)  =  {sig1_p(I,p) * [(numL1_p(I,p)/denL1_p(I,p)) + (numL2_p(I,p)/denL2_p(I,p))] + land_v.l(I,p)*bland_p(I)} + Y_v.l(I,p);
vmp_land_p (I,p)  =  price_p(I,p)   * mpp_land_p(I,p);

mpp_water_p(I,p)  =  {sig1_p(I,p) * [(numw1_p(I,p)/denw1_p(I,p)) + (numw2_p(I,p)/denw2_p(I,p))]};
vmp_water_p(I,p)  =  price_p(I,p)  * mpp_water_p(I,p);
;
display Yu_p, Yls_p, vmp_land_p, vmp_water_p;
```