```
$TITLE: M4-1.GMS: Cournot and Bertrand with continuous strategies

$ONTEXT
begin with Cournot doupoly
single unified market, constant marginal costs
goods XH and XF are imperfect substitutes
inverse demand functions PH = ALPHA - BETA*XH - GAMMA*XF   BETA > GAMMA
maximizing profits gives FOC (implicity reaction functions)
PROFIT = PH*XH - CH*XH = (ALPHA - BETA*XH - GAMMA*XF)*XH - CH*XH
first order condition:  ALPHA - 2*BETA*XH - GAMMA*XF - CH = 0
$OFFTEXT

PARAMETERS
 ALPHA   intercept of demand curve
 BETA    slope of inverse demand curve wrt own quantity
 GAMMA   slope of inverse demand curve wrt rival's quantity
 CH      marginal cost of home firm
 CF      marginal cost of foreign firm
 WELHC0  welfare in country h before policy under Cournot
 WELHB0  welfare in country h before policy under Bertrand;

ALPHA = 12;
BETA = 2;
GAMMA = 1;
CH = 2;
CF = 2;
```

```
NONNEGATIVE VARIABLES
 PH          price of XH
 PF          price of XF
 XH          quantity of XH
 XF          quantity of XF
 PROFH       profit of firm h
 PROFF       profit of firm f;

EQUATIONS
 PRICEH      inverse demand curve facing firm h
 PRICEF      inverse demand curve facing firm f
 HCOURNOT    cournot FOC for firm h (reaction function)
 FCOURNOT    cournot FOC for firm f (reaction function)
 PROFITH     profit of firm h
 PROFITF     profit of firm f;

PRICEH..    PH =E= ALPHA - BETA*XH - GAMMA*XF;

PRICEF..    PF =E= ALPHA - BETA*XF - GAMMA*XH;

HCOURNOT..  CH =G= ALPHA - 2*BETA*XH - GAMMA*XF;

FCOURNOT..  CF =G= ALPHA - 2*BETA*XF - GAMMA*XH;

PROFITH..   PROFH =E= PH*XH - CH*XH;
```

```
PROFITF..   PROFF =E= PF*XF - CF*XF;

MODEL COURNOT  /PRICEH.PH, PRICEF.PF, HCOURNOT.XH, FCOURNOT.XF,
                 PROFITH.PROFH, PROFITF.PROFF/;

SOLVE COURNOT USING MCP;

WELHC0 = PROFH.L;


$ONTEXT
now assume Bertrand price competition
requires you to invert the inverse demand functions
XH = INTERB - SLOPEB1*PH + SLOPEB2*PF
$OFFTEXT

PARAMETERS

 INTERB    intercept of the (direct) demand function
 SLOPEB1   slope of the demand function wrt own price
 SLOPEB2   slope of the demand function wrt rival's price;

INTERB =  (ALPHA*BETA - ALPHA*GAMMA)/(BETA**2 - GAMMA**2);
SLOPEB1 = BETA/(BETA**2 - GAMMA**2);
SLOPEB2 = GAMMA/(BETA**2 - GAMMA**2);
```

```
EQUATIONS
 XBERTH     demand for XH
 XBERTF     demand for XF
 HBERTRAND  bertrand FOC for PH
 FBERTRAND  bertrand FOC for PF;

XBERTH..    XH =E= INTERB - SLOPEB1*PH + SLOPEB2*PF;

XBERTF..    XF =E= INTERB - SLOPEB1*PF + SLOPEB2*PH;

HBERTRAND.. -SLOPEB1*CH =E= INTERB - 2*SLOPEB1*PH + SLOPEB2*PF;

FBERTRAND.. -SLOPEB1*CF =E= INTERB - 2*SLOPEB1*PF + SLOPEB2*PH;

MODEL BERTRAND  /XBERTH.XH, XBERTF.XF, HBERTRAND.PH, FBERTRAND.PF,
                 PROFITH.PROFH, PROFITF.PROFF/;

SOLVE BERTRAND USING MCP;

WELHB0 = PROFH.L;


*now analyze a production subsidy by h (strategic trade policy)
```

```
PARAMETER
 S  subsidy on H's output
 WELFAREHC  country h's welfare under Cournot
 WELFAREHB  country h's welfare under Bertrand;

S = 0.4;
CH = CH - S;

SOLVE COURNOT USING MCP;
WELFAREHC = PROFH.L - S*XH.L;
DISPLAY WELHC0, WELFAREHC;


SOLVE BERTRAND USING MCP;
WELFAREHB = PROFH.L - S*XH.L;
DISPLAY WELHC0, WELFAREHC,WELHB0, WELFAREHB;


$ONTEXT
now let's use nlp to find the OPTIMAL subsidies under Cournot and
Bertrand keep in mind that the optimal subsidy may be NEGATIVE, meaning
a tax let's play the goofy Brander-Spencer gams that all output is sold
to a third country.  Then welfare = profits minus subsidy payments
or plus tax payments.  PROFF will give the welfare of country f
$OFFTEXT
```

```
CH = 2;
S = 0;

VARIABLES
 WELFJ     joint welfare
 SUBH      subsidy on XH is now a (free) variable: it can be negative
 WELHS     welfare of country h: objective to maximize
 SUBF      subsidy on XF is now a (free) variable: it can be negative
 WELFS     welfare of country F: objective to maximize;

EQUATIONS
 WELJ         joint welfare - Cobb-Douglas
 WELH         welfare of country h is WELH = PROFH - SUBH*XH
 PROFITHS     new equation for profits of firm h - replaces PROFITH
 WELF         welfare of country f is WELF = PROFF - SUBF*XF
 PROFITFS     new equation for profits of firm f - replaces PROFITF
 HCOURNOTS    new Cournot reaction function firm h - replaces HCOURNOT
 HBERTRANDS   new Bertrand reaction function firm h - replaces HBERTRAND
 FCOURNOTS    new Cournot reaction function firm f - replaces FCOURNOT
 fBERTRANDS   new Bertrand reaction function firm f - replaces fBERTRAND;

WELJ..        WELFJ =E= WELHS**0.5*WELFS**0.5;

WELH..        WELHS =E= PROFH - SUBH*XH;

PROFITHS..    PROFH =E= PH*XH - (CH - SUBH)*XH;
```

```
HCOURNOTS..  (CH - SUBH) =E= ALPHA - 2*BETA*XH - GAMMA*XF;

HBERTRANDS.. -SLOPEB1*(CH-SUBH) =E= INTERB - 2*SLOPEB1*PH + SLOPEB2*PF;

WELF..       WELFS =E= PROFF - SUBF*XF;

PROFITFS..   PROFF =E= PF*XF - (CF - SUBF)*XF;

FCOURNOTS..  (CF - SUBF) =E= ALPHA - 2*BETA*XF - GAMMA*XH;

FBERTRANDS.. -SLOPEB1*(CF-SUBF) =E= INTERB - 2*SLOPEB1*PF + SLOPEB2*PH;

SUBH.L = 0.4;
WELHS.L = 8;



* first, a unilateral action by the government of country h

SUBF.FX = 0;

MODEL COURNOTS  /WELH, HCOURNOTS, FCOURNOT, PRICEH, PRICEF,
                 PROFITHS, PROFITF/;
SOLVE COURNOTS  USING NLP MAXIMIZING WELHS;
```

```
MODEL BERTRANDS /WELH, HBERTRANDS, FBERTRAND, XBERTH, XBERTF,
                 PROFITHS, PROFITF/;
SOLVE BERTRANDS USING NLP MAXIMIZING WELHS;

SUBF.UP = +INF;
SUBF.LO = -INF;

* compute cooperative and non-cooperative outcomes between governments

SETS I /I1*I10/
     J /COOP, NONCOOP/;

PARAMETER
 RESULTSC(*, J);


* compute a cooperative Nash eq between the governments

MODEL WELFJOINT /WELJ, WELH, WELF, HCOURNOTS, FCOURNOTS, PRICEH, PRICEF,
                 PROFITHS, PROFITFS/;
SOLVE WELFJOINT USING NLP MAXIMIZING WELFJ;


RESULTSC("WELJ", "COOP") = WELFJ.L;
RESULTSC("WELH", "COOP") = WELHS.L;
RESULTSC("WELF", "COOP") = WELFS.L;
```

```
RESULTSC("PROFITH", "COOP") = PROFH.L;
RESULTSC("PROFITF", "COOP") = PROFF.L;
RESULTSC("SUBH", "COOP") = SUBH.L;
RESULTSC("SUBF", "COOP") = SUBF.L;

DISPLAY RESULTSC;


* compute a non-cooperative outcome in subsidy rates
* iterative procedure:
*  max WELHS subject to SUBF fixed
*  hold SUBH at it's solution level and free up SUBF
*  max WELFS solve model for fixed SUBH
*  repeat 10 time

SUBH.L = 0;
SUBF.L = 0;

LOOP(I,

SUBH.LO = -INF;
SUBH.UP = +INF;
SUBF.FX = SUBF.L;

SOLVE WELFJOINT USING NLP MAXIMIZING WELHS;
```

```
SUBF.LO = -INF;
SUBF.UP = +INF;
SUBH.FX = SUBH.L;

SOLVE WELFJOINT USING NLP MAXIMIZING WELFS;

);

RESULTSC("WELJ", "NONCOOP") = WELFJ.L;
RESULTSC("WELH", "NONCOOP") = WELHS.L;
RESULTSC("WELF", "NONCOOP") = WELFS.L;
RESULTSC("PROFITH", "NONCOOP") = PROFH.L;
RESULTSC("PROFITF", "NONCOOP") = PROFF.L;
RESULTSC("SUBH", "NONCOOP") = SUBH.L;
RESULTSC("SUBF", "NONCOOP") = SUBF.L;

DISPLAY RESULTSC;
```