Some useful general remarks for GAMS:

(1) Multiple lines per statement, embedded blank lines, and multiple statements per line are allowed.

(2) You should terminate every statement with a semicolon (except lines started with $ symbol).

(3) The GAMS compiler doe not distinguish between upper- and lower-case letters.

(4) The names given to the entitles of the model must start with a letter and can be followed by up to 9 more letters or digits.

(5) The creation of GAMS entitles involves two steps: a declaration and an assignment / definition. “Declaration” means declaring the existence of something and giving it a name. “Assignment” or “definition” means giving something a specific value or dorm. In the case of equations you must make a declaration and definition in separate GAMS statements. For all other GAMS entitles, however, you have the option of making declaration and assignments in the same statement or separately.

(6)The first step in writing a GAMS model is to provide the constant elements (data that are not determined within the model, but they have a fixed value that you have to provide).

(7) Zero is default value for all parameters.

(7) The same parameter can be assigned value more than once. In contrast, the same parameter may not be declared more than once.

(8) Every variable must be assigned a type: free, positive, negative, binary, integer.

(9) The lower and upper bounds of a variable are set automatically according to the variable's type, but these bounds can be overwritten by: UP - to assign an upper bound, LO - to assign a lower bound, FX - to assign a fixed value, L - to assign an initial level

(10) The .LO and .UP fields are entirely under the control of the GAMS user. The .L and .M fields can be initialized by the user but are the controlled by the solver.

(11) The optimand (the variable that serves as the value to be optimized) must be a scalar of the free type (e.g. you cannot set optimand as a positive variable)

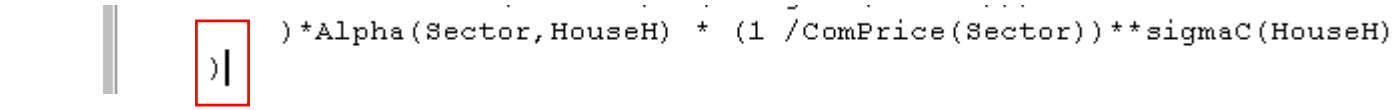
(12) The “=” symbol is used only in direct assignments. (A direct assignment gives a desired value to a parameter before the solver is called.) The “=e=” symbol is used only in equation definitions. (An equation definition also described relationship, but it cannot be satisfied until the solver is called). Equation definitions must contain variables and direct assignments must not.

(13) The equations need not be defined in the same order in which they are declared.

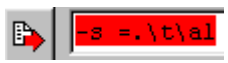
Useful Tools (con't) – Parentheses matching

parentheses match up

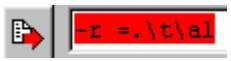
```
CommodMkt(Sector)..
  Production(Sector)
   =G=
  sum(HouseH,
      (HHIncome(HouseH)
        /sum(Sector1,alpha(Sector1,HouseH)
          *ComPrice(Sector1)**(1-SigmaC(HouseH)))
       )*Alpha(Sector,HouseH) * (1 /ComPrice(Sector))**sigmaC(HouseH)
   )
```



```
      )*Alpha(Sector,HouseH) * (1 /ComPrice(Sector))**sigmaC(HouseH)
   )
```

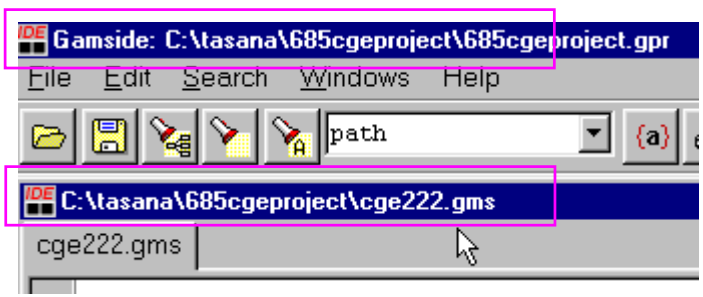


Command line saving parameters once defined in \t\a1



Command line calling/retrieving saved parameters from \t\a1

Caution:Make sure that you are working on files located in the same directory location as the project is located.



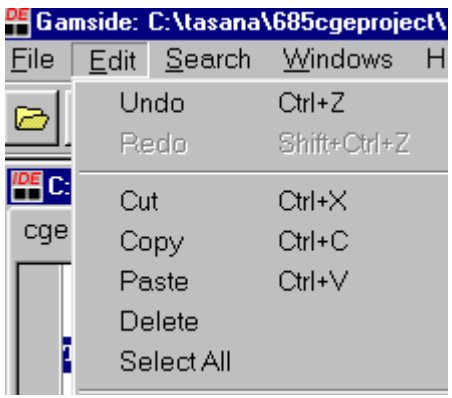
Useful Tools (con't) – Column block

Useful Tools (con't)

ALT+SHIFT moving column blocks of text

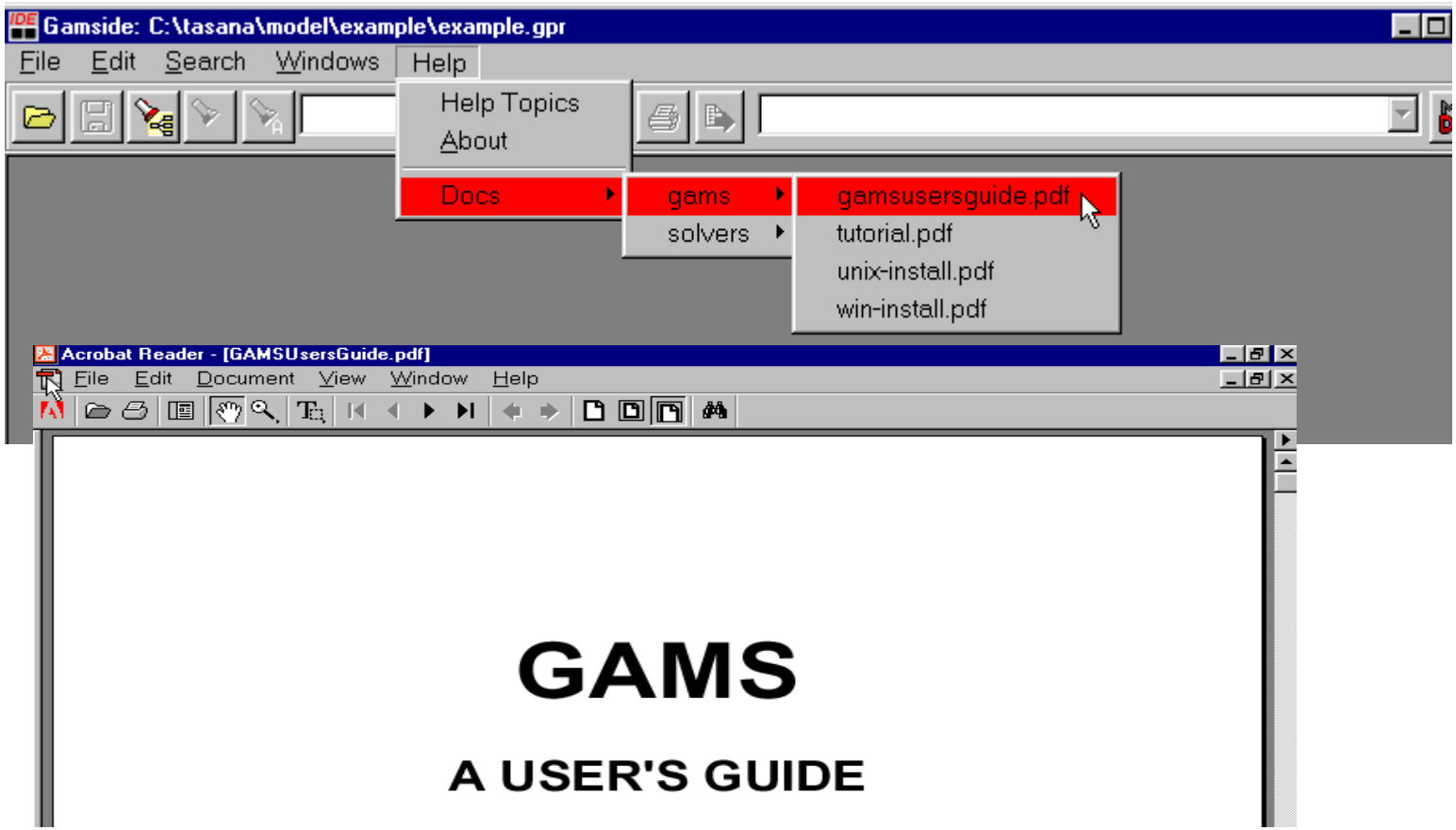
```
TABLE Delta(Factor,Sector)     distribution parameters attached to CES production f

                        Food      NonFood
        Labor           0.6       0.7
        Capital         0.4       0.3 ;
```

The copy, cut, and paste can be done with the Edit menus as in normal windows.



GAMS Documentation – GAMS instruction

- **GAMS** documentation is accessible through the help menu under the choice GAMS



GAMS Documentation – Path