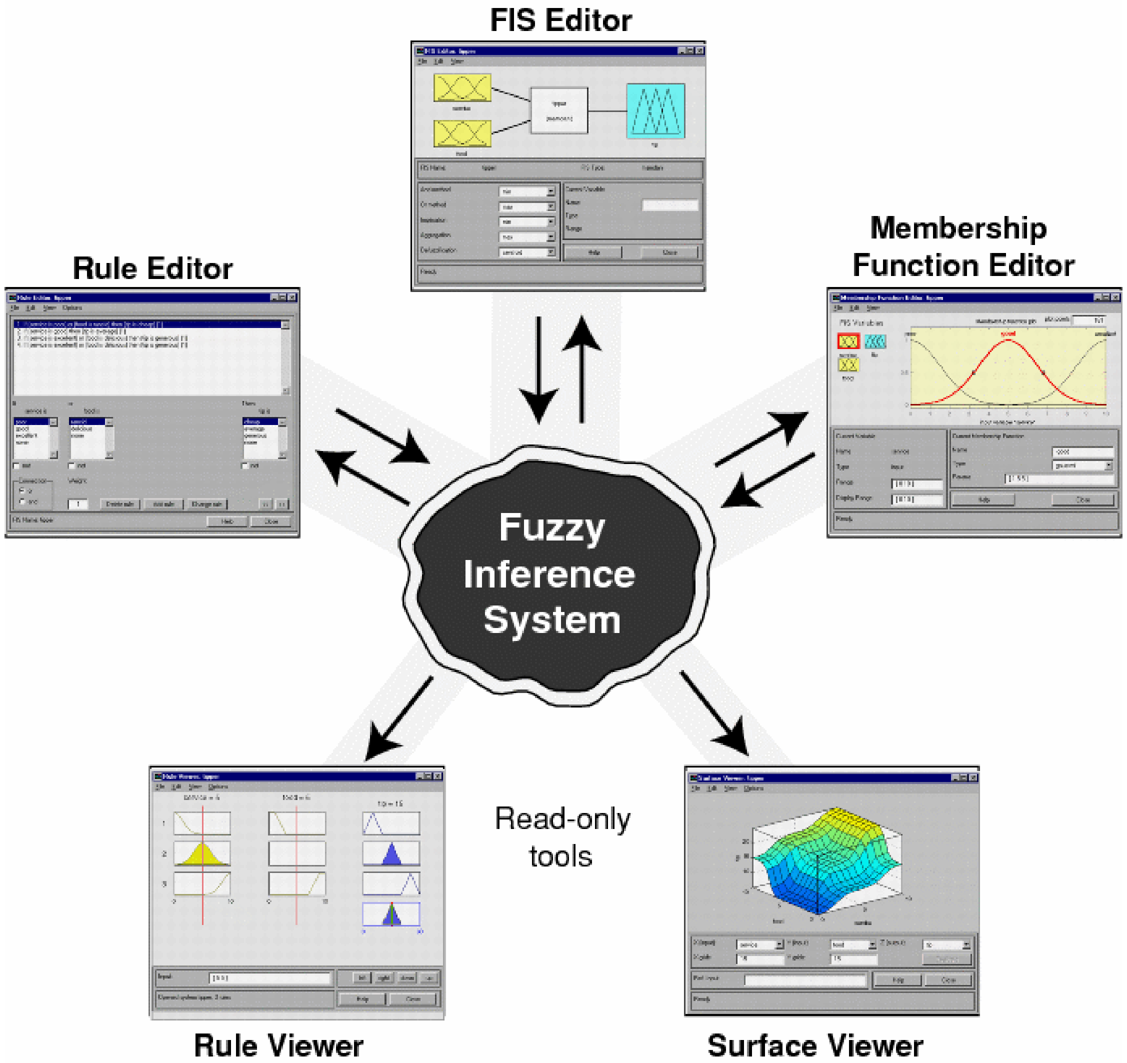
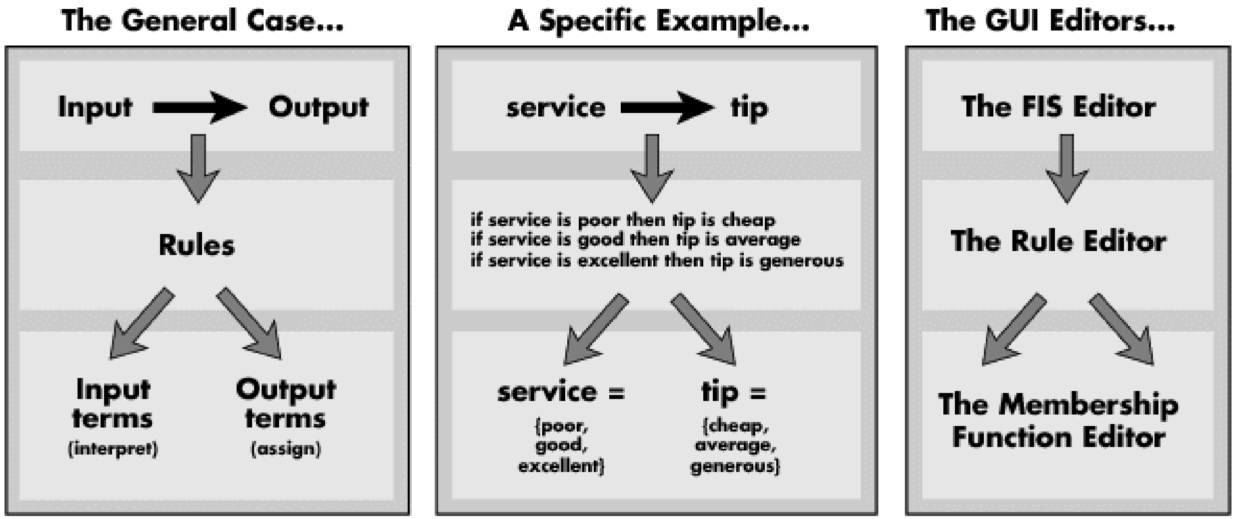
Fuzzy Logic Toolbox Graphical User Interface Tools

- Fuzzy Inference System (FIS) Editor
- Membership Function Editor
- Rule Editor
- Rule Viewer
- Surface Viewer



Getting Started

A basic description of a two-input, one-output tipping problem



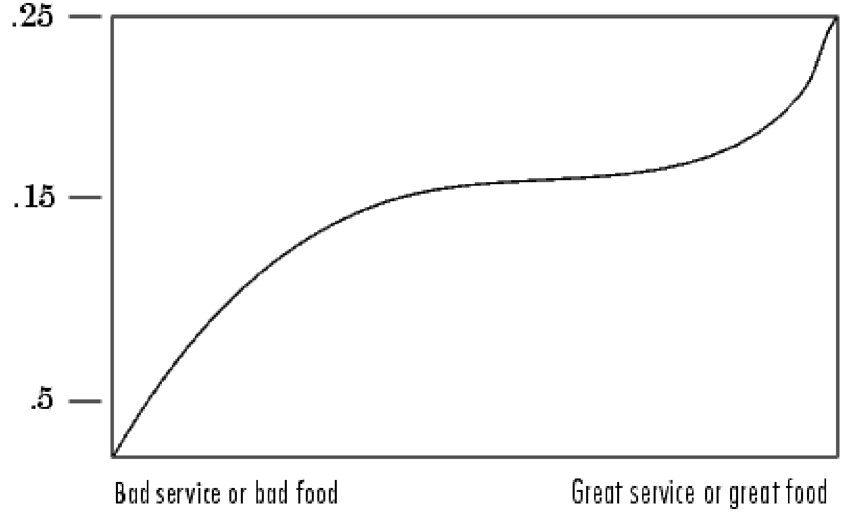
The Basic Tipping Problem

A number [0 … 10] represents the quality of service at a restaurant (10 is excellent), and [0 … 10] represents the quality of the food at that restaurant (10 is excellent), what should the tip be?

Three rules of tipping, based on years of personal experience in restaurants.

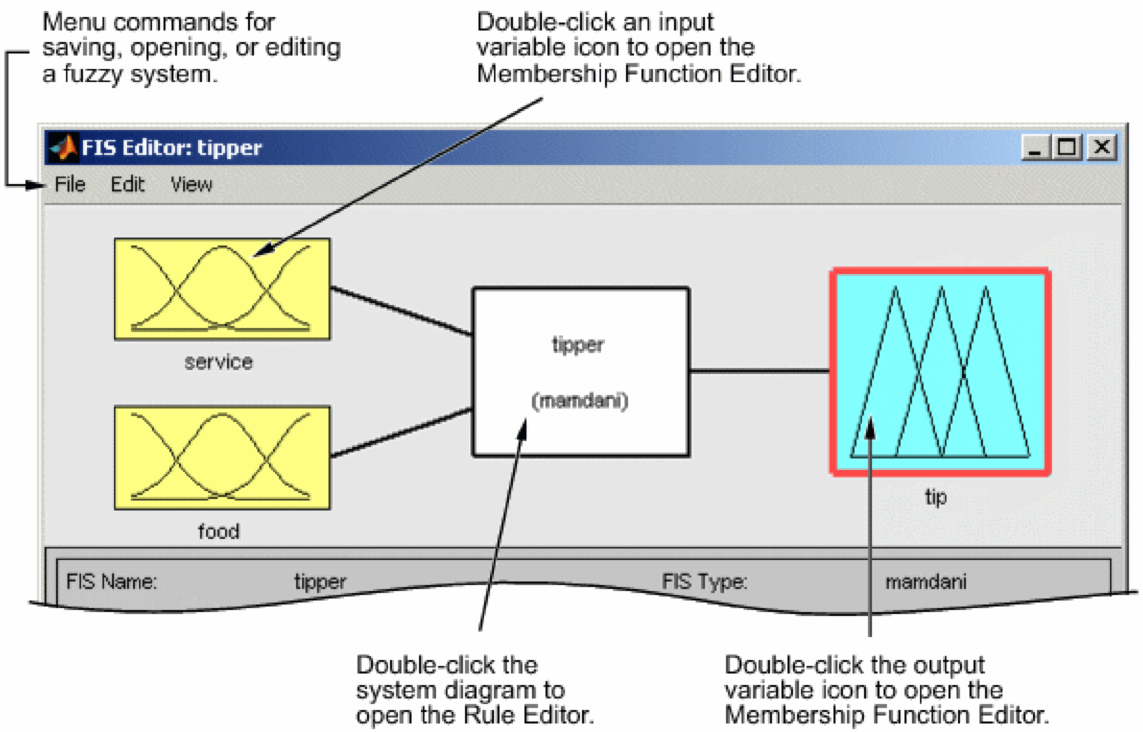
1. If the service is poor or the food is rancid, then tip is cheap.
2. If the service is good, then tip is average.
3. If the service is excellent or the food is delicious, then tip is generous.

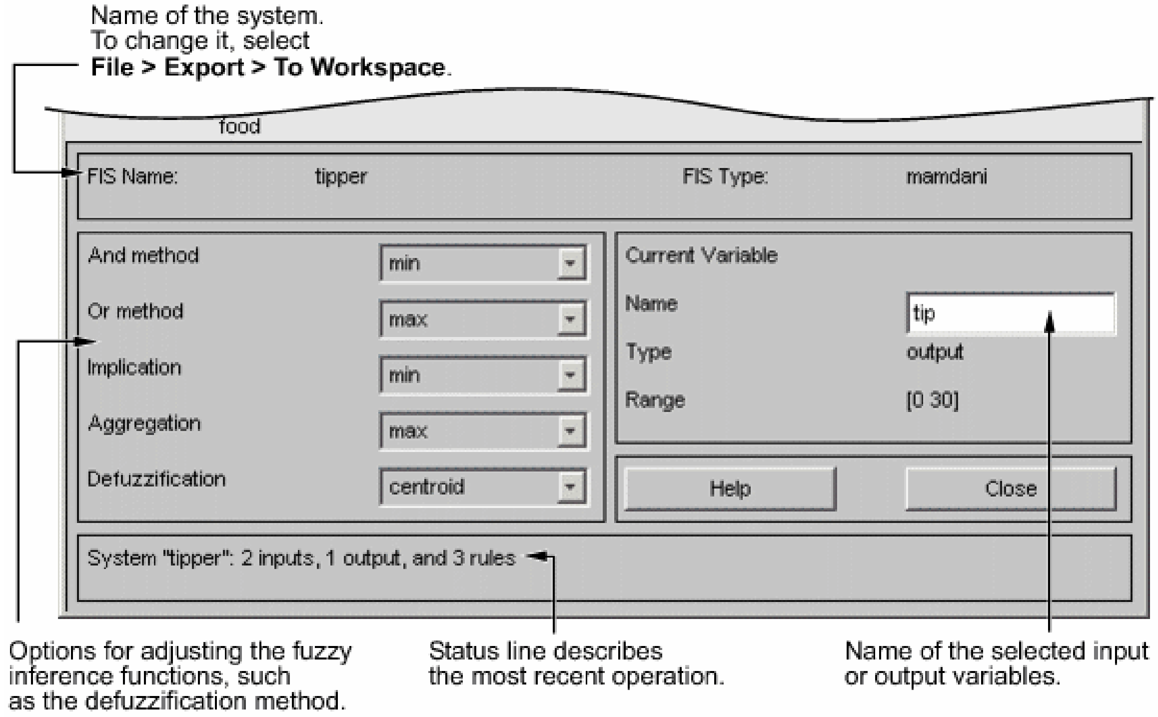
Assume that an average tip is 15%, a generous tip is 25%, and a cheap tip is 5%.



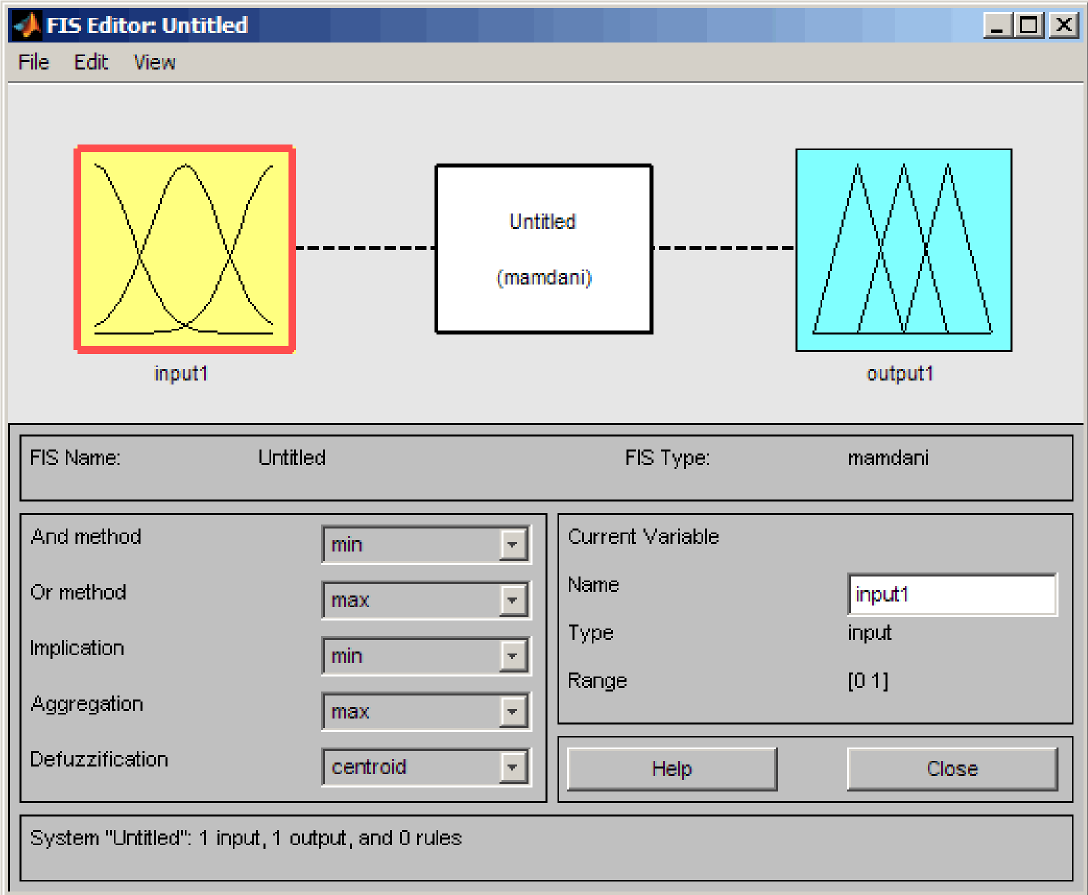
The FIS Editor

>> Fuzzy




Name of the system.
To change it, select
File > Export > To Workspace.
food
FIS Name:
tipper
FIS Type:
mamdani
And method
min
Or method
max
Implication
min
Aggregation
max
Defuzzification
centroid
Current Variable
Name
tip
Type
output
Range
[0 30]
Help
Close
System "tipper": 2 inputs, 1 output, and 3 rules
Options for adjusting the fuzzy inference functions, such as the defuzzification method.
Status line describes the most recent operation.
Name of the selected input or output variables.

The generic untitled FIS Editor opens, with one input labeled **input1**, and one output labeled **output1**.



Construct a two-input, one output system. The two inputs are **service** and **food**. The one output is **tip**.

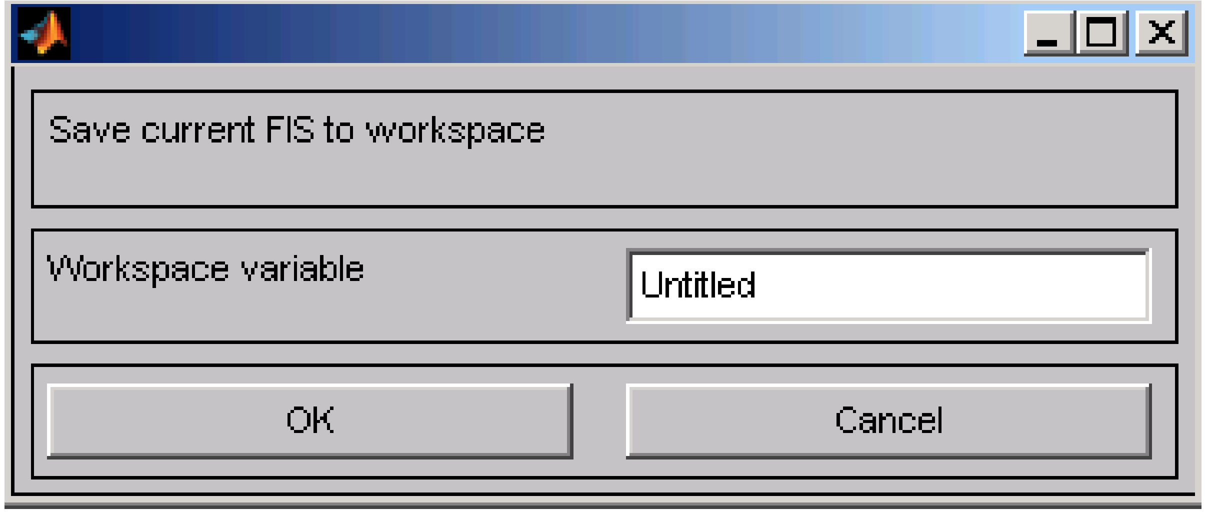
To add a second input variable and change the variable names to reflect these designations:

1. Select **Edit** > **Add variable** > **Input**.

   A second yellow box labeled **input2** appears.

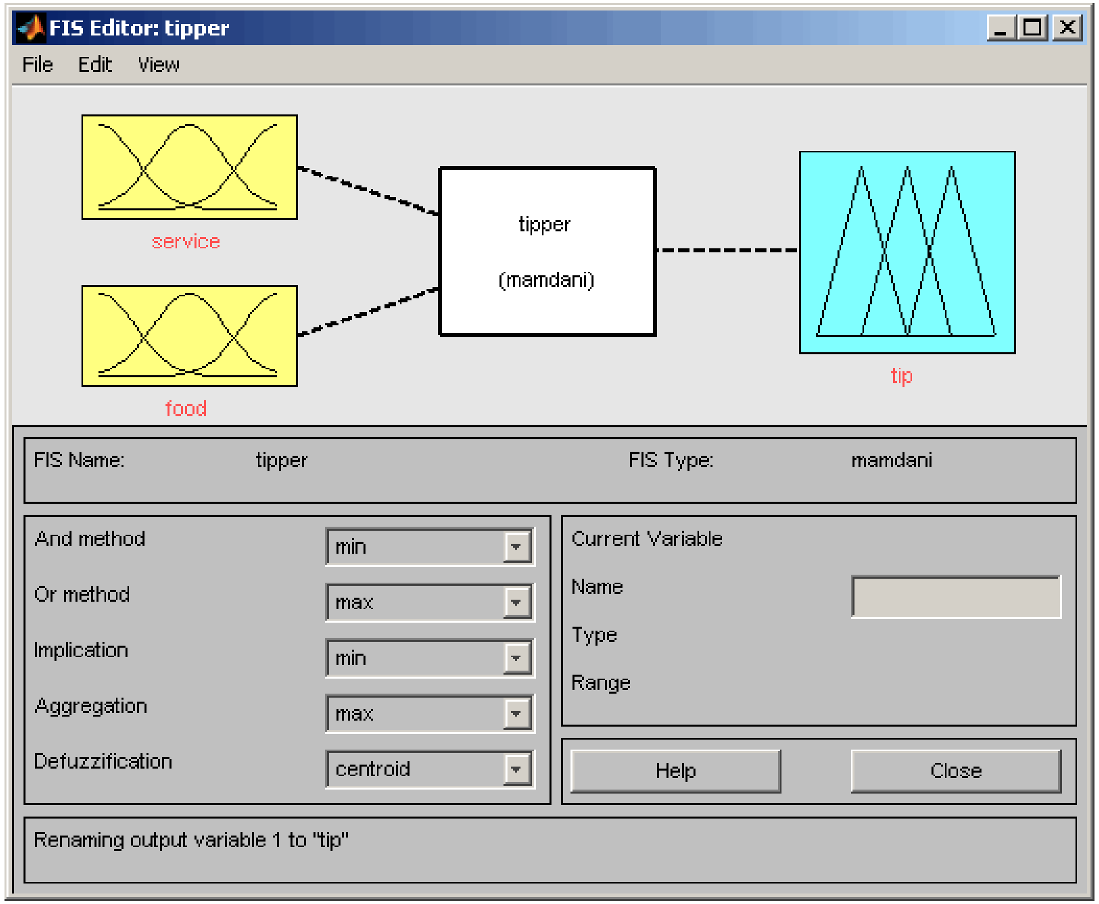
2. Click the yellow box **input1**. This box is highlighted with a red outline.
3. Edit the **Name** field from input1 to “service”, and press **Enter**.
4. Click the yellow box **input2**. This box is highlighted with a red outline.
5. Edit the **Name** field from input2 to “food”, and press **Enter**.
6. Click the blue box **output1**.
7. Edit the **Name** field from output1 to tip, and press **Enter**.

8. Select **File** > **Export** > **To Workspace**.



9. Enter the **Workspace variable** name “tipper”, and click **OK**.

Your window looks something like the following diagram.



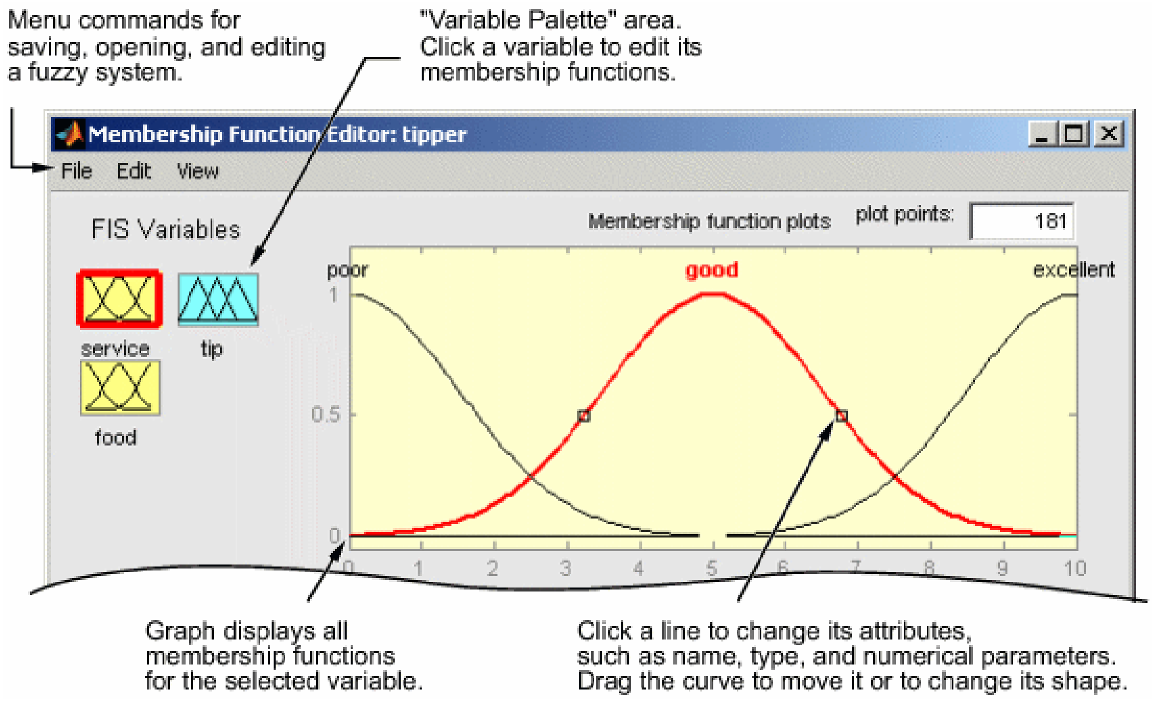
Leave the inference options in the lower left in their default positions for now. You have entered all the information you need for this particular GUI.

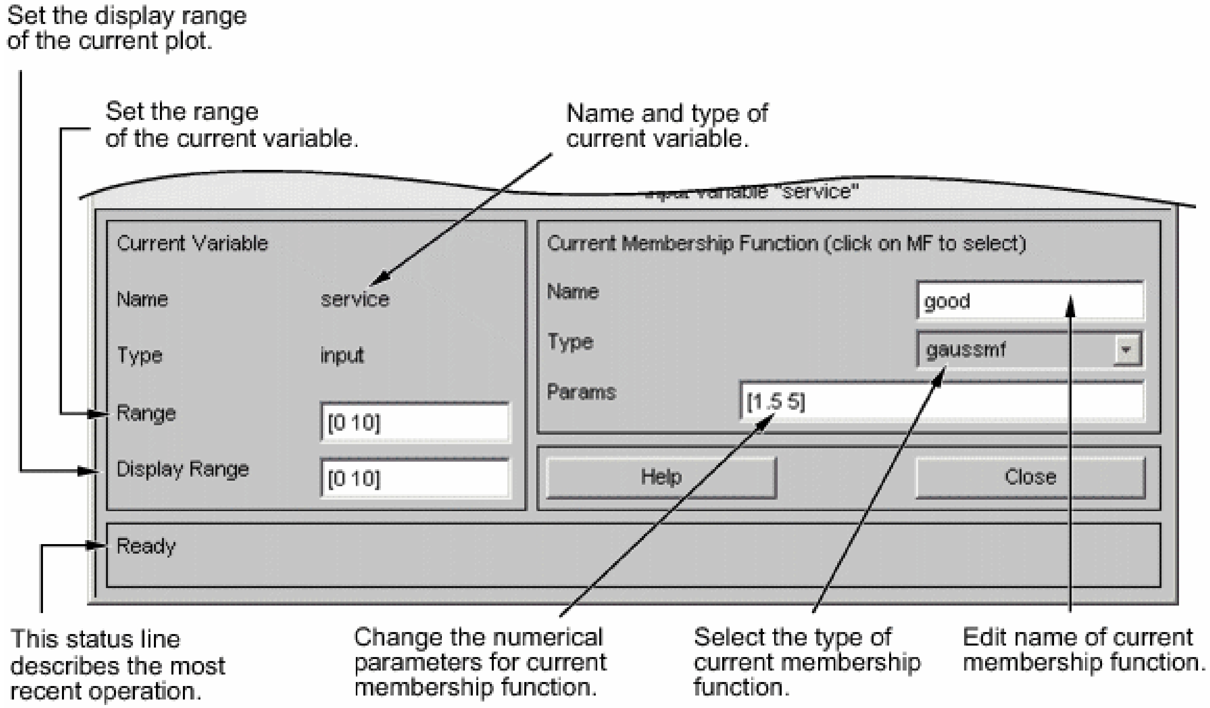
Next, define the membership functions associated with each of the variables. To do this, open the Membership Function Editor.

You can open the Membership Function Editor in one of three ways:

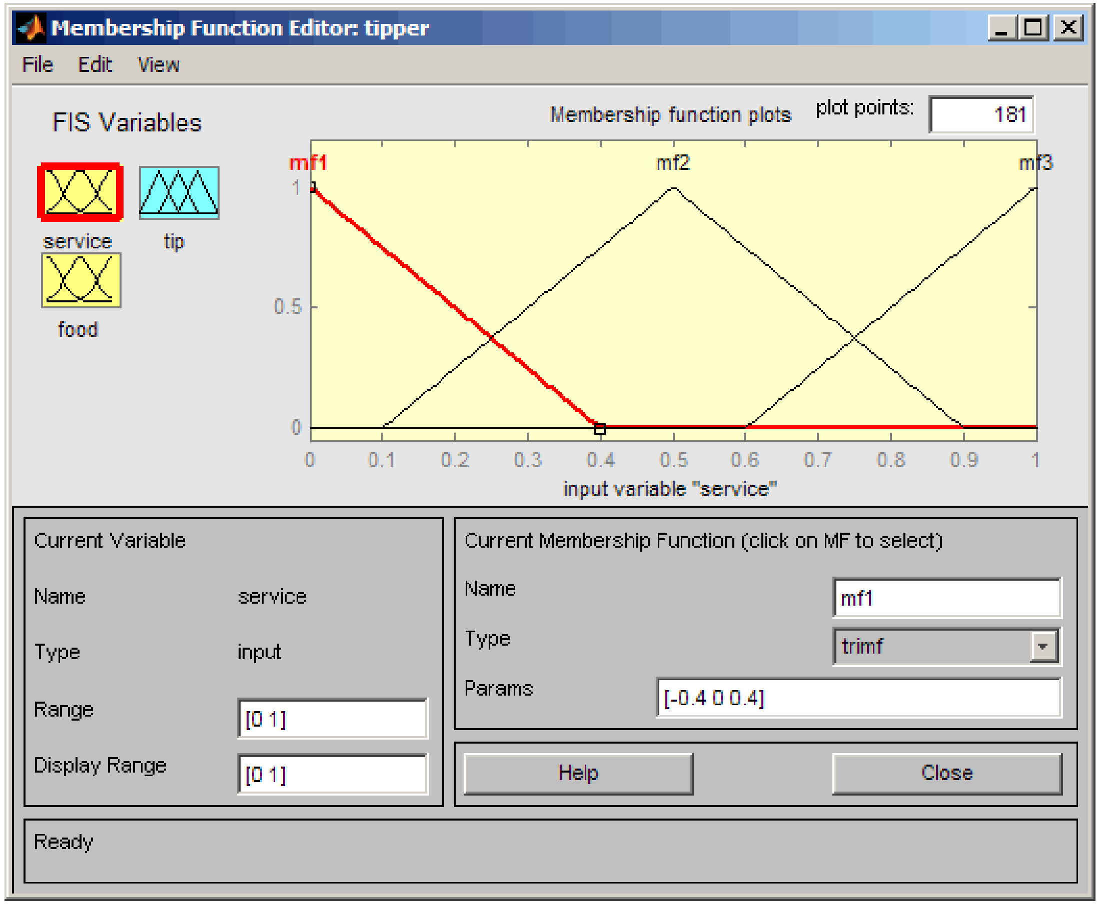
- Within the FIS Editor window, select **Edit** > **Membership Functions.**.
- Within the FIS Editor window, double-click the blue icon called **tip**.
- At the command line, type mfedit.

The Membership Function Editor

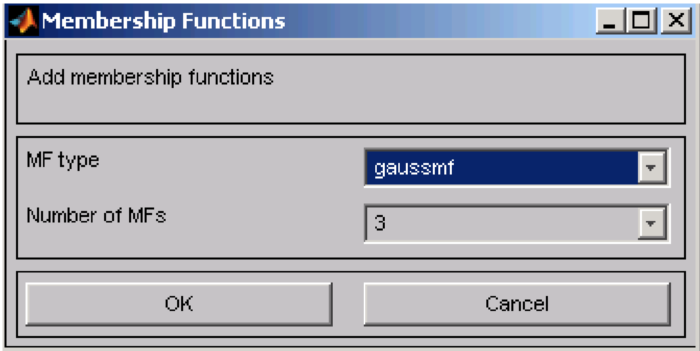



Set the display range
of the current plot.
Set the range
of the current variable.
Name and type of
current variable.
Input variable "service"
Current Variable
Name
service
Type
input
Range
[0 10]
Display Range
[0 10]
Current Membership Function (click on MF to select)
Name
good
Type
gaussmf
Params
[1.5 5]
Help
Close
Ready
This status line
describes the most
recent operation.
Change the numerical
parameters for current
membership function.
Select the type of
current membership
function.
Edit name of current
membership function.

1. Double-click the input variable service to open the Membership Function Editor.



2. In the Membership Function Editor, enter [0 10] in the **Range** and the **Display Range** fields.
3. Create membership functions for the input variable service.
   a. Select **Edit** > **Remove All MFs** to remove the default membership functions for the input variable service.
   b. Select **Edit** > **Add MFs.** to open the Membership Functions dialog box.
   c. In the Membership Functions dialog box, select gaussmf as the **MF Type**.



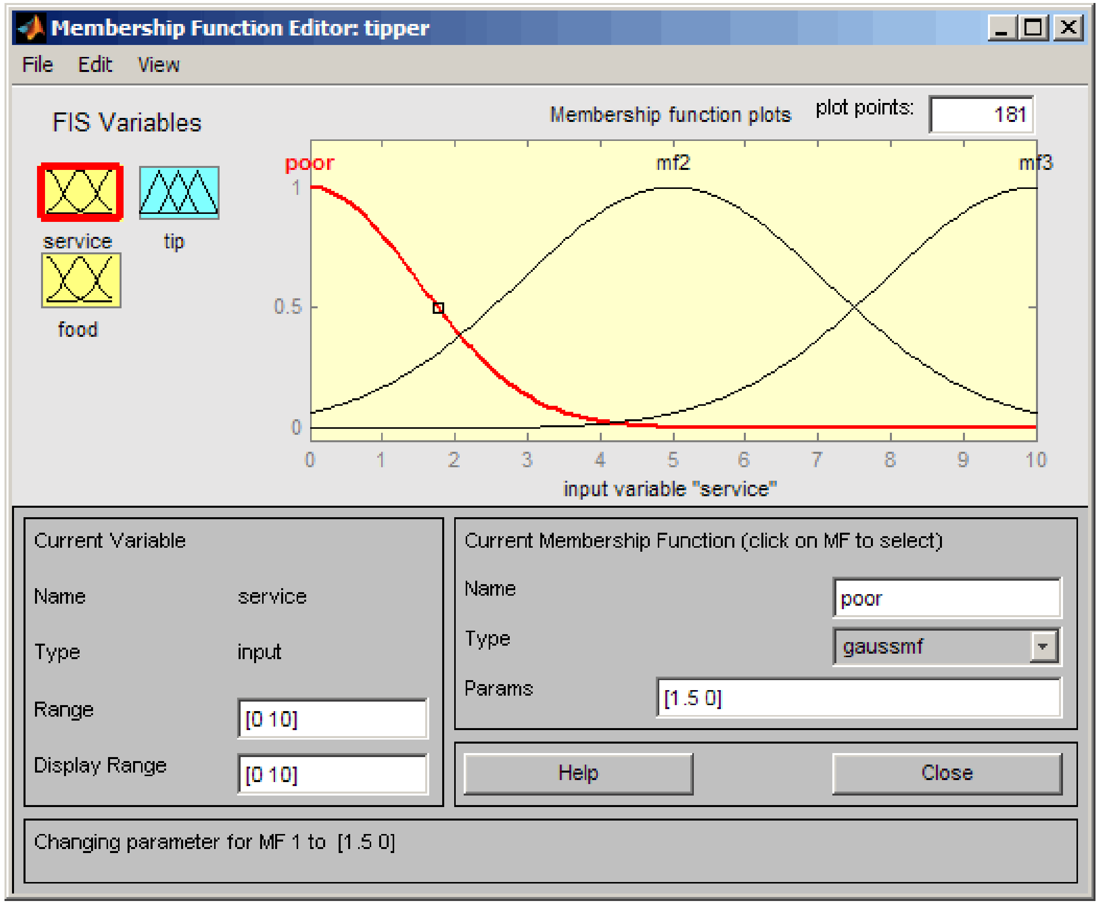
   d. Verify that 3 is selected as the **Number of MFs**.
   e. Click **OK** to add three Gaussian curves to the input variable service.

4. Rename the membership functions for the input variable service, and specify their parameters.
   a. Click on the curve named mf1 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:
      - In the **Name** field, enter poor.
      - In the **Params** field, enter [1.5 0].

      The two inputs of **Params** represent the standard deviation and center for the Gaussian curve.

      > **Tip** To adjust the shape of the membership function, type in a desired parameters or use the mouse, as described previously.

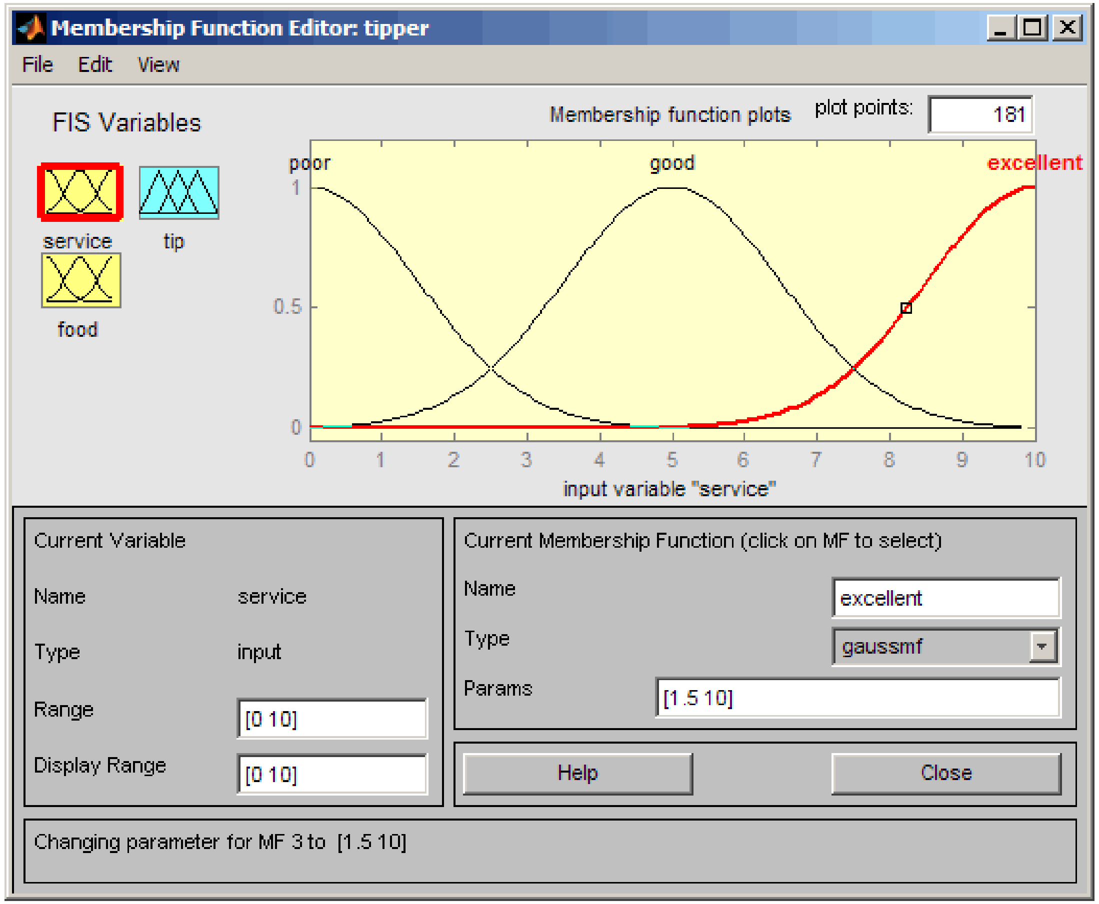
The Membership Function Editor: tipper window looks similar to the following figure.



b. Click on the curve named mf2 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:
   - In the **Name** field, enter good.
   - In the **Params** field, enter [1.5 5].

c. Click on the curve named mf3, and specify the following fields in the **Current Membership Function (click on MF to select)** area:
   - In the **Name** field, enter excellent.
   - In the **Params** field, enter [1.5 10].

The Membership Function Editor window looks similar to the following figure.



5. In the **FIS Variables** area, click the input variable food to select it.
6. Enter [0 10] in the **Range** and the **Display Range** fields.
7. Create the membership functions for the input variable food.

a. Select **Edit** > **Remove All MFs** to remove the default Membership Functions for the input variable food.
b. Select **Edit** > **Add MFs** to open the Membership Functions dialog box.
c. In the Membership Functions dialog box, select trapmf as the **MF Type**.
d. Select 2 in the **Number of MFs** drop-down list.
e. Click **OK** to add two trapezoidal curves to the input variable food.

8. Rename the membership functions for the input variable food, and specify their parameters:
   a. In the **FIS Variables** area, click the input variable food to select it.
   b. Click on the curve named mf1, and specify the following fields in the **Current Membership Function (click on MF to select)** area:
      - In the **Name** field, enter rancid.
      - In the **Params** field, enter [0 0 1 3].
   c. Click on the curve named mf2 to select it, and enter delicious in the **Name** field. Reset the associated parameters if desired.
9. Click on the output variable tip to select it.

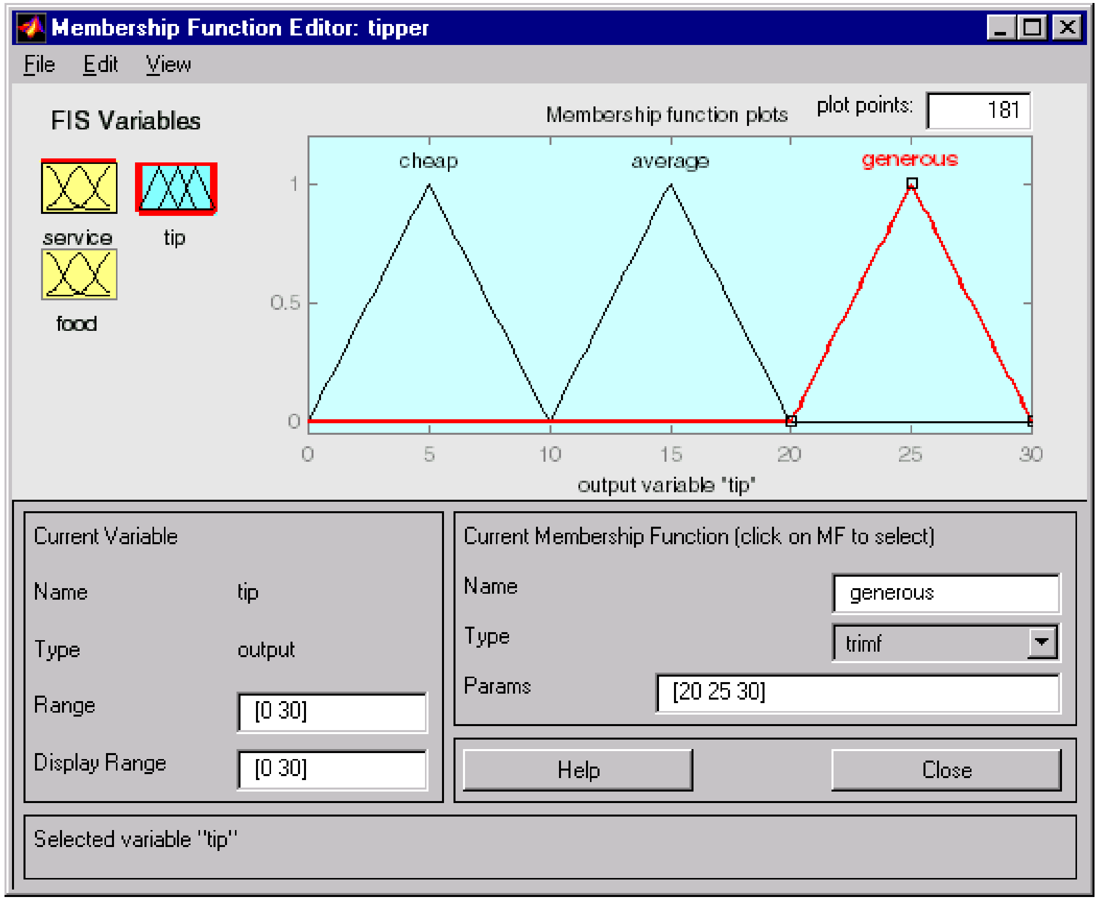
10. Enter [0 30] in the **Range** and the **Display Range** fields to cover the output range.

The inputs ranges from 0 to 10, but the output is a tip between 5% and 25%.

11. Rename the default triangular membership functions for the output variable tip, and specify their parameters.
    a. Click the curve named mf1 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:
        - In the **Name** field, enter cheap.
        - In the **Params** field, enter [0 5 10].
    b. Click the curve named mf2 to select it, and specify the following fields in the **Current Membership Function (click on MF to select)** area:
        - In the **Name** field, enter average.
        - In the **Params** field, enter [10 15 20].
    c. Click the curve named mf3 to select it, and specify the following:

- In the **Name** field, enter generous.
- In the **Params** field, enter [20 25 30].

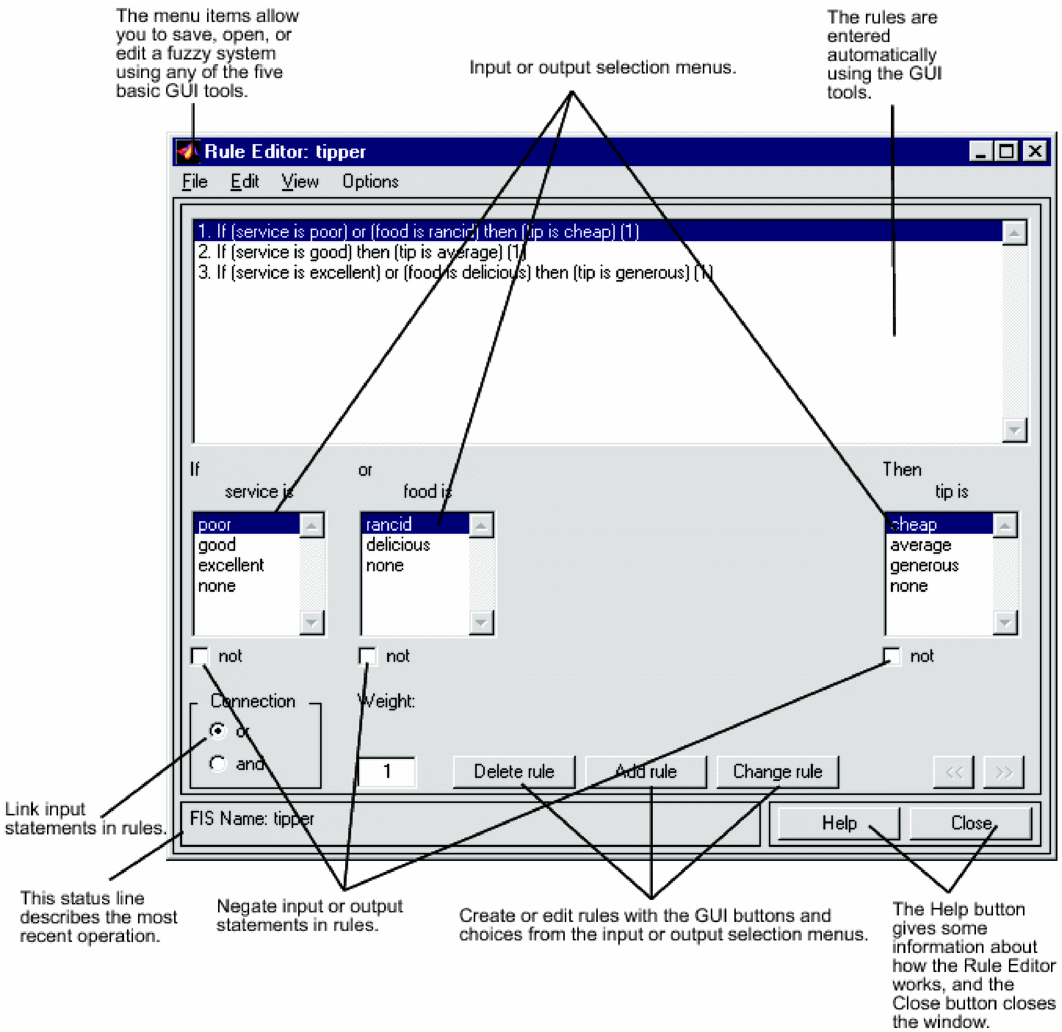
The Membership Function Editor looks similar to the following figure.



Now that the variables have been named and the membership functions have appropriate shapes and names, you can enter the rules.

To call up the Rule Editor, go to the **Edit** menu and select **Rules**, or type ruleedit at the command line.

The Rule Editor



From the GUI, you can:

- Create rules by selecting an item in each input and output variable box, selecting one **Connection** item, and clicking **Add Rule**. You can choose none as one of the variable qualities to exclude that variable from a given rule and choose not under any variable name to negate the associated quality.
- Delete a rule by selecting the rule and clicking **Delete Rule**.
- Edit a rule by changing the selection in the variable box and clicking **Change Rule**.
- Specify weight to a rule by typing in a desired number between 0 and 1 in **Weight**. If you do not specify the weight, it is assumed to be unity (1).

The menu items allow you to open, close, save and edit a fuzzy system using the five basic GUI tools. From the menu, you can also:

- Set the format for the display by selecting **Options** > **Format**.
- Set the language by selecting **Options** > **Language**.

You can access information about the Rule Editor by clicking **Help** and close the GUI using **Close**.

To insert the first rule in the Rule Editor, select the following:

- poor under the variable **service**
- rancid under the variable **food**
- The **or** radio button, in the **Connection** block
- cheap, under the output variable, **tip**.

Then, click **Add rule**.

The resulting rule is

1. If (service is poor) or (food is rancid) then (tip is cheap) (1)

The numbers in the parentheses represent weights.

Follow a similar procedure to insert the second and third rules in the Rule Editor to get

1. *If (service is poor) or (food is rancid) then (tip is cheap) (1)*
2. *If (service is good) then (tip is average) (1)*
3. *If (service is excellent) or (food is delicious) then (tip is generous) (1)*

Tip To change a rule, first click on the rule to be changed. Next make the desired changes to that rule, and then click **Change rule**. For example, to change the first rule to

1. If (service not poor) or (food not rancid) then (tip is not cheap) (1)

Select the **not** check box under each variable, and then click **Change rule**.

The **Format** pop-up menu from the **Options** menu indicates that you are looking at the verbose form of the rules. Try changing it to symbolic. You will see

1. (service==poor) | (food==rancid) => (tip=cheap) (1)

2. (service==good) => (tip=average) (1)

3. (service==excellent) | (food==delicious) => (tip=generous) (1)

There is not much difference in the display really, but it is slightly more language neutral, because it does not depend on terms like *if* and *then*. If you change the format to indexed, you see an extremely compressed version of the rules.

1 1, 1 (1) : 2
2 0, 2 (1) : 1
3 2, 3 (1) : 2

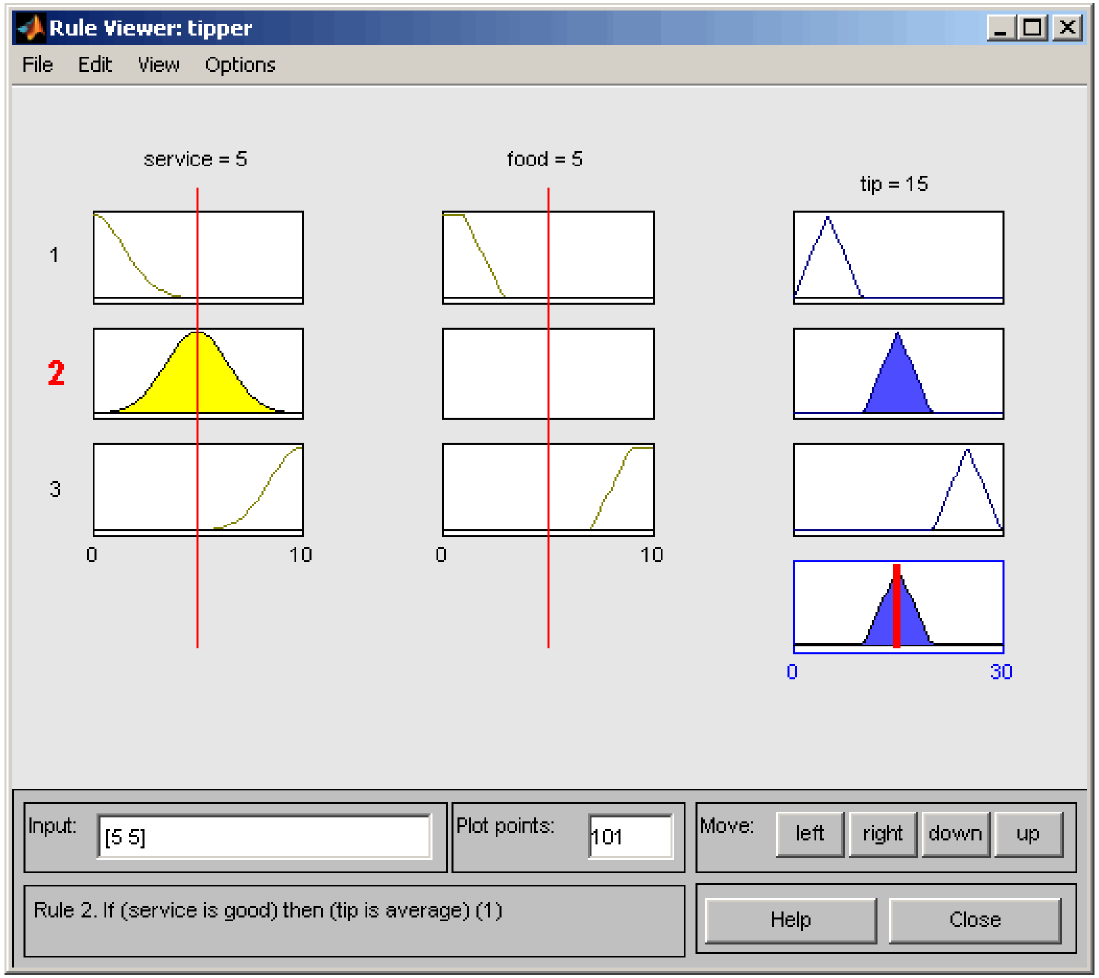
This is the version of the rules that the machine deals with.

- The first column in this structure corresponds to the input variables.
- The second column corresponds to the output variable.
- The third column displays the weight applied to each rule.
- The fourth column is shorthand that indicates whether this is an OR (2) rule or an AND (1) rule.

- The numbers in the first two columns refer to the index number of the membership function.

A literal interpretation of rule 1 is "If input 1 is MF1 (the first membership function associated with input 1) or if input 2 is MF1, then output 1 should be MF1 (the first membership function associated with output 1) with the weight 1.

The Rule Viewer



The Rule Viewer displays a roadmap of the whole fuzzy inference process. You can click on a rule number to view the rule in the status line.

- The first two columns of plots (the six yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule.
- The third column of plots (the three blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule.

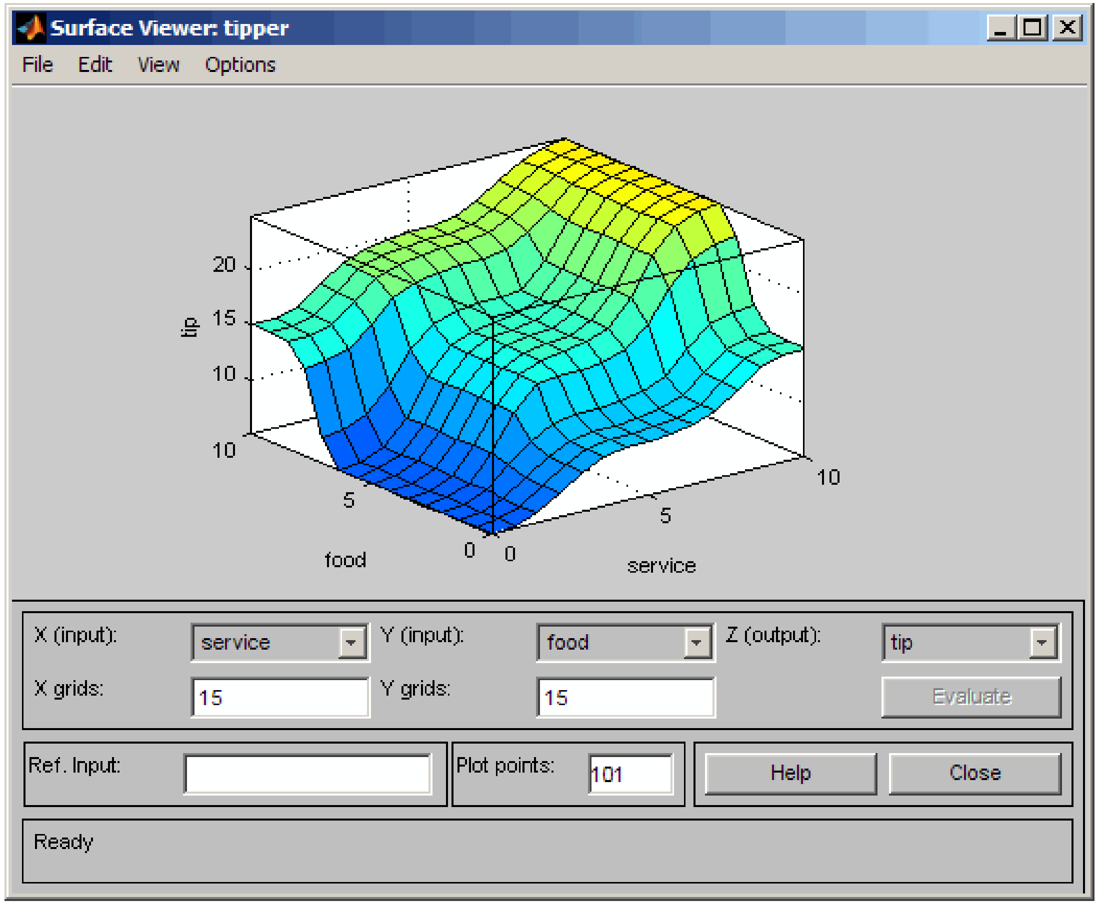
  Notice that under **food**, there is a plot which is blank. This corresponds to the characterization of none for the variable **food** in the second rule.

- The fourth plot in the third column of plots represents the aggregate weighted decision for the given inference system.

  This decision will depend on the input values for the system. The defuzzified output is displayed as a bold vertical line on this plot.

The variables and their current values are displayed on top of the columns. In the lower left, there is a text field **Input** in which you can enter specific input values.

The Surface Viewer



Upon opening the Surface Viewer, you see a three-dimensional curve that represents the mapping from food and service quality to tip amount.

Because this curve represents a two-input one-output case, you can see the entire mapping in one plot.

When we move beyond three dimensions overall, we start to encounter trouble displaying the results.

Accordingly, the Surface Viewer is equipped with drop-down menus **X (input):**, **Y (input):** and **Z (output):** that let you select any two inputs and any one output for plotting.

Below these menus are two input fields **X grids:** and **Y grids:** that let you specify how many x-axis and y-axis grid lines you want to include.

This capability allows you to keep the calculation time reasonable for complex problems.

If you want to create a smoother plot, use the **Plot points** field to specify the number of points on which the membership functions are evaluated in the input or output range. By default, the value of this field is 101.

Clicking **Evaluate** initiates the calculation, and the plot is generated after the calculation is complete. To change the *x*-axis or *y*-axis grid after the surface is in view, change the appropriate input field, and press **Enter**. The surface plot is updated to reflect the new grid settings.

The Surface Viewer has a special capability that is very helpful in cases with two (or more) inputs and one output: you can grab the axes, using the mouse and reposition them to get a different three-dimensional view on the data.

The **Ref. Input** field is used in situations when there are more inputs required by the system than the surface is mapping. You can edit this field to explicitly set inputs not specified in the surface plot.

Suppose you have a four-input one-output system and would like to see the output surface. The Surface Viewer can generate a three-dimensional output surface where any two of the inputs vary, but two of the inputs must be held constant because computer monitors cannot display a five-dimensional shape. In such a case, the input is a four-dimensional vector with NaNs holding the place of the varying inputs while numerical values indicates those values that remain fixed. A NaN is the IEEE® symbol for Not a Number.

The menu items allow you to open, close, save and edit a fuzzy system using the five basic GUI tools. You can access information about the Surface Viewer by clicking **Help** and close the GUI using **Close**.

This concludes the quick walk-through of each of the main GUI tools. For the tipping problem, the output of the fuzzy system matches your original idea of the shape of the fuzzy mapping from service to tip fairly well. In hindsight, you might say, "Why bother? I could have just drawn a quick lookup table and been done an hour ago!" However, if you are interested in solving an entire class of similar decision-making problems, fuzzy logic may provide an appropriate tool for the solution, given its ease with which a system can be quickly modified.

```
>> tipper

tipper =

             name: 'tipper'
             type: 'mamdani'
        andMethod: 'min'
         orMethod: 'max'
     defuzzMethod: 'centroid'
        impMethod: 'min'
        aggMethod: 'max'
            input: [1x2 struct]
           output: [1x1 struct]
             rule: [1x3 struct]
```