Habfuzz

Calculating the instream hydraulic habitat suitability based on fuzzy logic using FORTRAN

A quick user's guide

by
Christos Theodoropoulos

September 2016

Contact Information

Hellenic Centre for Marine Research
Institute of Marine Biological Resources and Inland Waters
46.7 km Athens-Sounio ave.
19013
Anavyssos
Greece
Tel. +30 22910 76335
Fax. +30 22910 76419
Email. ctheodor@hcmr.gr
URL. http://www.hcmr.gr/en/

National Technical University of Athens
Department of Water Resources and Environmental Engineering
Iroon Polytechniou 5
15780
Athens
Greece
Tel. +30 210 7722809
Fax. +30 210 7722814
Email. stamou@central.ntua.gr
URL. https://www.hydro.ntua.gr/?set_language=en

To report errors, bugs, possible amendments or anything else you would like to indicate, please contact Mr. Christos Theodoropoulos at ctheodor@hcmr.gr

TABLE OF CONTENTS

Habfuzz is a FORTRAN 95 code, which implements the Mamdani - Assilian fuzzy inference process (Mamdani and Assilian, 1975). It is specifically structured to quickly calculate the fuzzy-logic-based instream habitat suitability for fish or freshwater macroinvertebrates along a 2D hydraulically simulated river reach. However, if appropriately modified, it can be applied to wider research topics requiring fuzzy logic to be addressed.

Why Habfuzz?

Instead of Habfuzz, you can use the non-free MATLAB Fuzzy Logic Toolbox (http://www.mathworks.com/products/fuzzy-logic/) or the free CASiMiR 2D software upon request (http://www.casimir-software.de/ENG/habitate_eng.html). However, Habfuzz has been designed to be a one-click tool, for those researchers with no or very minor programming knowledge, in need of an easy-to-use software to calculate the habitat suitability along a hydrodynamically simulated river reach, based on fuzzy logic. For those researchers who can't afford to purchase the MATLAB (because it does everything, but they only need a small amount of its capabilities). And for those self-studying researchers who need a very comprehensive, step-by-step, yet short-legthed tutorial to enable them quickly run a tool for a specific part of their project.

1. Overview of the fuzzy inference process

As initially proposed by Zadeh (1965) and described in detail by Ross (2010), the process of deriving the fuzzy-based habitat suitability, given the flow velocity, water depth and substrate type, can be summarized in four steps (Fig. 1):

a. Fuzzification of the input variables
   In this step, the user defines categories (membership functions) for each input variable and the input values of flow velocity, water depth and substrate type are assigned to one or more membership functions. By this procedure, crisp numerical values of each input variable are converted to a fuzzy ‘degree of membership’, ranging from 0 to 1 for each membership function. For example, a depth value of 14 cm may derive a membership degree of 0.7 for the ‘shallow’ membership function and 0.28 for the ‘very shallow’ membership function.

b. Application of a fuzzy operator (AND or OR) in the antecedent (IF-THEN rules)
According to the reference data for the target aquatic community, the AND (min) or OR (max) operator is applied to each combination of variables (membership functions since step 1) and the derived value is assigned to the membership function of the output variable (defined in step 1), in this case the habitat suitability. For example, if the user defines five membership functions for habitat suitability (bad, poor, moderate, good, high), then the application of the fuzzy operator would result in,

$$f_4(hs) = min\ \left(f_2(D), f_3(V)\right)$$
$$f_3(hs) = min\ \left(f_3(D), f_2(V)\right)$$
$$f_4(hs) = min\ \left(f_2(D), f_4(V)\right)$$

where,
f_i denotes for the membership function of each input and output variable
V is the flow velocity
D is the water depth
hs is the habitat suitability

etc., until all possible combinations of fuzzy inputs are assigned to an output membership function, based on the rationale that, for example,
IF flow velocity is f_3 (moderate) AND water depth is f_2 (shallow) THEN habitat suitability is f_4 (good).

c. Aggregation of outputs
In this step, the derived habitat suitability membership functions from each rule are combined into one fuzzy set. Usually, the OR (max) operator is applied to aggregate the same output fuzzy sets of the previous step. For example, the *f_4 (hs)* is derived in the previous example two times by the IF-THEN rules. The final fuzzy set representing each habitat suitability class F_j would be

$$F_j = \max(f_i^1(hs), f_i^2(hs), \dots, f_i^v(hs))$$

d. Defuzzification
This final step is applied to derive one single habitat suitability value, by combining the membership degrees of all fuzzy habitat suitability classes.

Among the various defuzzification methods, the 'centroid', 'maximum membership', 'weighted average' and 'mean-max membership methods' are described below.

i. Centroid defuzzification:

Usually called the 'center of gravity' or 'center of area'. It can be defined by the algebraic expression

$$hs = \frac{\int xf(x)dx}{\int f(x)dx}$$

which can be numerically approximated by

$$hs = \frac{\sum_{i=1}^{n} x_i(f(x_i))}{\sum_{i=1}^{n}(f(x_i))}$$

where,
$f(x_i)$ is the membership degree at value x_i

ii. Maximum membership defuzzification - This is the maximum membership degree observed by the aggregation step:

$$hs = max(f(x))$$

iii. Weighted average - This method can be used only for symmetrical output membership functions and is calculated by weighting each output membership function by its largest membership degree:

$$hs = \frac{\sum_{i=1}^{n} \overline{x_i}(f(\overline{x_i}))}{\sum_{i=1}^{n}(f(\overline{x_i}))}$$

where,
$f(\overline{x_i})$ is the membership degree at the average value $\overline{x_i}$ *of each membership function*

iv. Mean of maximum - This method resembles the 'maximum membership' method. However, the maximum membership degree may not be unique but a range of degrees, from which the mean value is derived:

$$hs = \frac{x_a + x_b}{2}$$

where,

x_a is the first value reaching the highest membership degree of the class with the highest membership and

x_b is the last value with the highest membership degree of the class with the highest membership

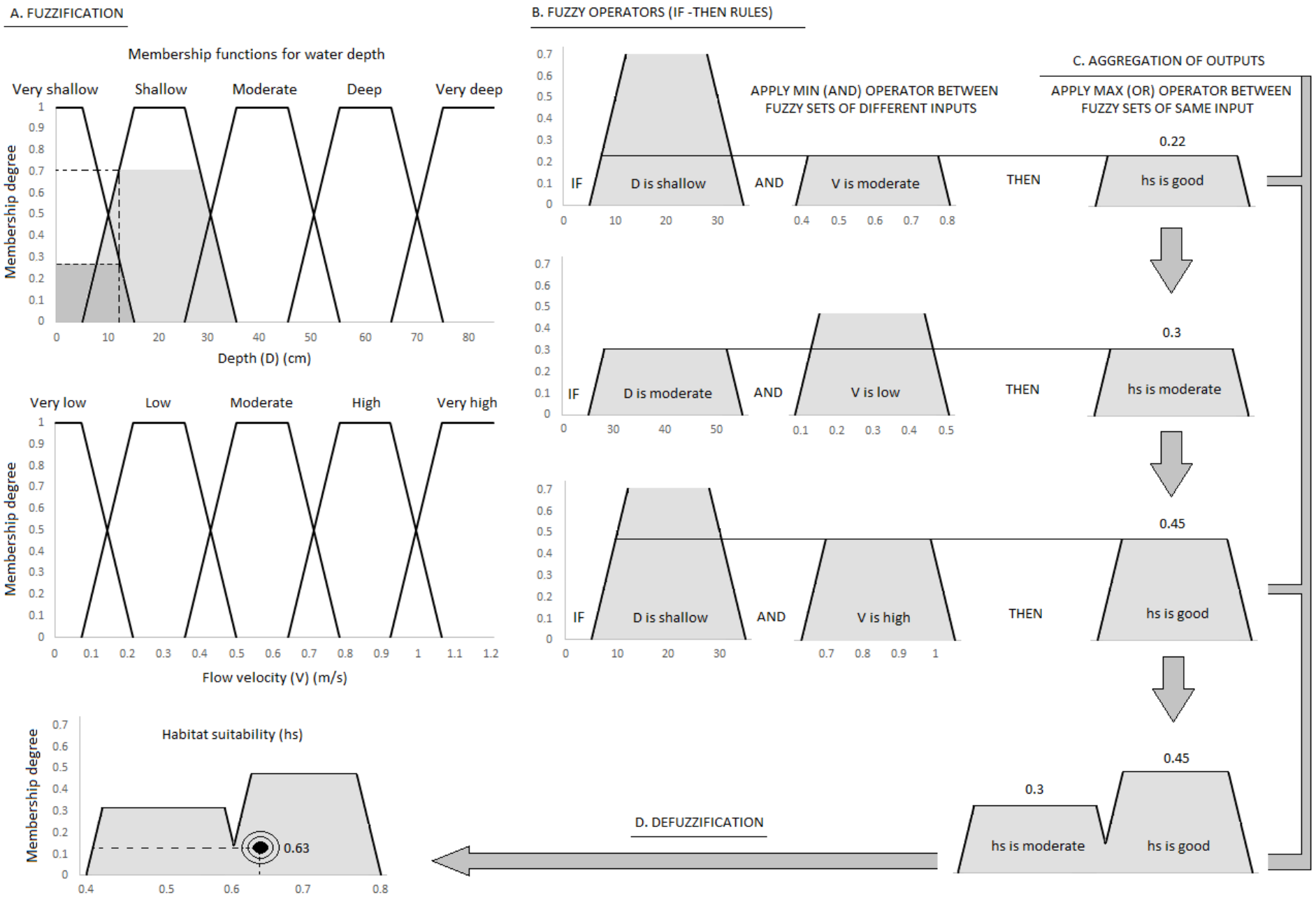

Figure 1: Steps of the Mamdani - Assilian fuzzy inference process

2. Dependencies

All the necessary files to run Habfuzz are included in the program's folder. However, to modify the code, users will need to have a FORTRAN text editor and a FORTRAN compiler installed.

Habfuzz was developed using:

- the Geany text editor (download at www.geany.org)
- the GFortran compiler (download at https://gcc.gnu.org/wiki/GFortranBinaries)

Therefore, it is advised to install the specific additional software to ensure that Habfuzz is working properly.

3. Installing

Habfuzz has been developed to run on 32-bit and 64-bit Windows operating systems, including Windows XP, Vista, 7/8 and 10. No specific installation of the software is required. Just copy the Habfuzz folder to your hard disk and double-click *Habfuzz.exe* to run the program. The Habfuzz folder includes:

1. The 'Habfuzz' subfolder, which contains all the code files of Habfuzz, which are
   a. The *fdeclarations.f95* module containing the number of input arrays and all the variables and parameters necessary to run the program.
   b. The *fuzzifier.f95* subroutine containing the code to apply the fuzzification process (see the 'usage' section)
   c. The *smod.f95, swors.f95* and *sopt.f95* subroutines containing the IF-THEN rules according to the management scenario to be implemented (see the 'usage' section)
   d. The *centroid.f95, meanmax.f95, waver.f95 and maxmem.f95* subroutines containing the code to apply the 'centroid', 'mean-max membership', 'weighted average' and 'maximum membership' defuzzification processes.
2. The 'bin' subfolder, which includes three necessary .dll files to run the software

If the user needs to change the code in one of these files, compilation is necessary prior to running the program. With the gfortran compiler installed, the user can either type the necessary commands

```
gfortran -c fdeclarations.f95
gfortran -c Habfuzz.f95 fdeclarations.f95 fuzzifier.f95 smod.f95 swors.f95 sopt.f95 centroid.f95 maxmem.f95 waver.f95 meanmax.f95
gfortran -o Habfuzz Habfuzz.f95 fdeclarations.f95 fuzzifier.f95 smod.f95 swors.f95 sopt.f95 centroid.f95 maxmem.f95 waver.f95 meanmax.f95
```

or just run the *compile.bat* file (included in the 'Habfuzz' subfolder) to compile (in this case, don't change the names of files!!!). The compiler will then create the new *Habfuzz.exe* and some *.o* files, which can be discarded by the user.

4. Usage

4.1. Input and output data

To run Habfuzz, three different ASCII files are required as input, containing the flow velocity values in m/s, the depths in m and substrate types, according to the Manning's n as depicted in table 1 and detailed in the appendix. These files should be located in the 'Habfuzz' subfolder and named *velocities.txt, depths.txt* and *substrates.txt* accordingly. Normally, such files (after proper manipulation) are the outputs of a hydrodynamic (hydraulic) simulation, where a specific river reach is numerically simulated through a computational grid. The simulation assigns a value for flow velocity, depth and substrate type at each node of the grid. However, the user can still create his/her own files to use in Habfuzz. All values at each file should be arranged in a single column, where the first row denotes the number of elements in the row and the rest of the values being the actual data. An example of the three input files containing 10 values (nodes) is shown in Figure 2.

velocities.txt	depths.txt	substrates.txt
10	10	10
0.28	0.34	0.022
0.36	0.05	0.040
0.45	0.67	0.050
0.67	1.45	0.070
0.89	0.23	0.070
0.05	0.19	0.022
0.45	0.23	0.040
0.32	0.67	0.050
0.76	0.37	0.070
0.56	0.51	0.050

Figure 2: The Habfuzz input file format

The output of Habfuzz is a .txt file named *suitability.exe* containing a single column with all the habitat suitabilities (ranging from 0 - unsuitable to 1 - suitable) calculated for each input element (node) in the same order as with the input files. This file is placed by the program in the 'Habfuzz' subfolder.

The habitat suitability is initially a combination of fuzzy membership functions (five classes of suitability - bad, poor, moderate, good and high) and through the defuzzification process it is converted into a crisp output ranging from 0 to 1. The inputs and the output of Habfuzz are depicted in Fig. 1.

Table 1: Manning's n for various substrate types applied in Habfuzz

Bed material	Size (diameter)	Manning's n
Boulders	>25 cm	0.070
Large stones	12-25 cm	0.050
Small stones	6-12 cm	0.040
Large gravel	2-6 cm	0.030
Medium gravel	0.6-2 cm	0.026
Fine gravel	0.2-0.6 cm	0.024
Sand	<0.2 cm	0.022
Silt	-	0.020

4.2. Running Habfuzz

After having the input files ready, double click *Habfuzz.exe.* The command prompt opens and the software asks for the management scenario to be implemented (Fig. 3).

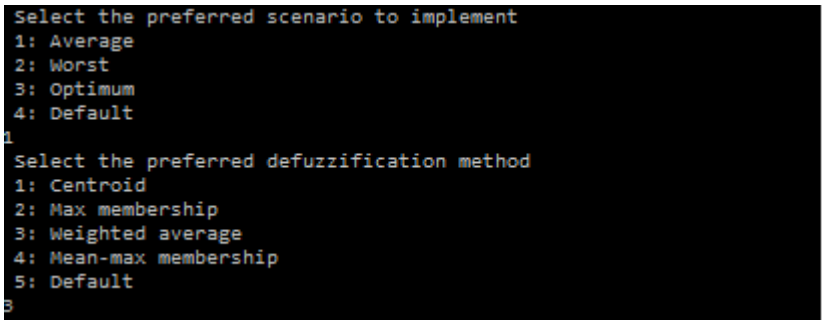

Figure 3: The user defined inputs of the program

There are three available scenarios based on the method used for deriving the outcome of each IF-THEN rule from the reference conditions of the program, (i) the moderate scenario, where the different suitability values for the same combinations of flow velocity, water depth and substrate type are averaged to derive the final suitability, (ii) the worst scenario, where the final suitability is derived from the minimum observed suitability and (iii) the optimum scenario where the final suitability is derived by the maximum observed suitability. A default scenario is also present (the moderate scenario). Note that if a specific combination in the observed data does not match a combination in the reference data, the program returns a value of '-1' for the habitat suitability.

After selecting the desired scenario, the user is asked to select the defuzzification method (see section 1). A default method (centroid) is also available. After selecting the defuzzification method, Habfuzz calls the relevant FORTRAN

subroutines to perform the tasks selected. The program informs the user when the process is completed and indicates the *suitability.txt* file created where the suitability values are stored.
The file is located in the 'Habfuzz' subfolder.

Figure 4: Calling subroutines and creating the results file

4.3. Modifying the code according to the user preferences

While the software is developed to quickly apply the fuzzy inference process for the calculation of the habitat suitability, the user can change the code according to his/her requirements in order to apply the fuzzy inference process for other topics requiring the implementation of fuzzy logic. Possible changes can be applied at specific Habfuzz processes mentioned below:

1. INPUT ARRAY SIZE - The current limit of input array size (the number of rows at each file and therefore the number of nodes in the computational grid) is set at 3000. However, the user can change this value by changing accordingly the 'rsize' parameter in the *fdeclarations.f95* file.
2. FUZZIFICATION - The fuzzification process of Habfuzz is included in the subroutine *fuzzifier.f95.* It converts the crisp inputs of flow velocity and water depth to fuzzy sets (membership functions). The program creates five trapezoidal-shaped flow velocity functions (five classes of flow velocity), (a) very low, (b) low, (c) moderate, (d) high and (e) very high and five classes of water depth (a) very shallow, (b) shallow, (c) moderate, (d) deep and (e) very deep. The substrate type is treated as a crisp input throughout the process since the types of substrate are well defined and there is no need to be fuzzified. The user can either change the values at each membership function included in the *fdeclarations.f95* file (marked as A. PARAMETERS OF THE FUZZIFICATION PROCESS), which results in changing the trapezoidal vertices of each membership function (Fig. 5) or change the whole membership function.

```
!-------------------------------------------------------------------------------
! A. PARAMETERS OF THE FUZZIFICATION PROCESS:
!-------------------------------------------------------------------------------

!INPUT 1: Flow velocity (u) - 5 classes, very low, low, moderate, high, very high
real :: uvlow, ulow, umoderate, uhigh, uvhigh
real, dimension(rsize) :: cuvlow, culow, cumoderate, cuhigh, cuvhigh
real, parameter :: uvla = 0.10, uvlb = 0.15 !VERY LOW u class
real, parameter :: ula = 0.10, ulb = 0.15, ulc = 0.27, uld = 0.32 !LOW u class
real, parameter :: uma = 0.27, umb = 0.32, umc = 0.55, umd = 0.60 !MODERATE u class
real, parameter :: uha = 0.55, uhb = 0.60, uhc = 0.95, uhd = 1.00 !HIGH u class
real, parameter :: uvha = 0.95, uvhb = 1.00 !VERY HIGH u class

!INPUT 2: Depth (d) - 5 classes, very shallow, shallow, moderate, deep, very deep
real :: dvshallow, dshallow, dmoderate, ddeep, dvdeep
real, dimension(rsize) :: cdvshallow, cdshallow, cdmoderate, cddeep, cdvdeep
real, parameter :: dvla = 0.07, dvlb = 0.10 !The VERY SHALLOW d class
real, parameter :: dla = 0.07, dlb = 0.10, dlc = 0.27, dld = 0.30 !The SHALLOW d class
real, parameter :: dma = 0.27, dmb = 0.30, dmc = 0.57, dmd = 0.60 !The MODERATE d class
real, parameter :: dha = 0.57, dhb = 0.60, dhc = 0.77, dhd = 0.80 !The DEEP d class
real, parameter :: dvha = 0.77, dvhb = 0.80 !The VERY DEEP d class

!-------------------------------------------------------------------------------
```

Figure 5: The fuzzy membership functions for flow velocity and water depth. Green numbers are the vertices of each trapezoidal function

3. IF-THEN RULES - The user can also apply modifications to the IF-THEN rules of the *smod.f95, swors.f95* and *sopt.f95* subroutines. The reference conditions are currently derived using freshwater macroinvertebrates in an extended sampling campaign, which took place at 9 sampling sites in Greece and resulted in a set of 380 reference microhabitats (combinations of flow velocity, water depth and substrate type). If the user needs to apply his/her own reference data, changes should be applied to one of the abovementioned subroutines. However, if for example, the *smod.f95* file is modified, the user should also change the column number of the arrays, which are used by *smod.f95* (in B. INTERNAL PARAMETERS TO FACILITATE THE IMPLICATION AND AGGREGATION STEPS, marked as 'Used by smod.f95'), which are the arrays g, h, m, p, b. The modification should be such that the column number is the same as the times that a combination in the *smod.f95* file results to a specific result. If you see for example, 'g' is the outcome of 26 combinations and so the column number for 'g' in the *fdeclarations.f95* file is also 26.

```
!-------------------------------------------------------------------------
!B. INTERNAL PARAMETERS TO FACILITATE THE IMPLICATION AND AGGREGATION STEPS
!-------------------------------------------------------------------------
integer :: scenario, dfuzz
real, dimension(rsize) :: high, good, moderate, poor, bad, hs

!Used by smod.f95
real, dimension(rsize,4) :: h
real, dimension(rsize,26) :: g
real, dimension(rsize,32) :: m
real, dimension(rsize,11) :: p
real, dimension(rsize,1) :: b

!Used by swors.f95
real, dimension(rsize,3) :: h2
real, dimension(rsize,5) :: g2
real, dimension(rsize,28) :: m2
real, dimension(rsize,33) :: p2
real, dimension(rsize,4) :: b2
```

Figure 6: The parameters of the *fdeclarations.f95* file requiring to be changed if the IF-THEN rules are changed

It is not advised to make any changes in the 'defuzzification' subroutines as they do not depend on the array size of other files and they don't require any changes to work properly even if the abovementioned modifications are applied. Still, an experienced FORTRAN user can modify the defuzzification subroutines according to his/her needs. After each modification in the files of Habfuzz, re-compilation is necessary as described in section 3.

5. References

Mamdani E.H. and Assilian S. 1975. An experiment in linguistic synthesis with a fuzzy logic
Controller. International Journal of Man-Machine Studies 7: 1-13.

Ross T.J. 2010. Fuzzy logic with engineering applications. Third Edition, John Wiley and Sons, UK.

Zadeh L.A. 1965. Fuzzy sets. Information and Control 8: 338–353.

APPENDIX

Manning's n for channels (Chow, 1959)

Type of Channel and Description	Minimum	Normal	Maximum
Natural streams - minor streams (top width at floodstage < 100 ft)			
1. Main Channels			
a. clean, straight, full stage, no rifts or deep pools	0.025	0.030	0.033
b. same as above, but more stones and weeds	0.030	0.035	0.040
c. clean, winding, some pools and shoals	0.033	0.040	0.045
d. same as above, but some weeds and stones	0.035	0.045	0.050
e. same as above, lower stages, more ineffective slopes and sections	0.040	0.048	0.055
f. same as "d" with more stones	0.045	0.050	0.060
g. sluggish reaches, weedy, deep pools	0.050	0.070	0.080
h. very weedy reaches, deep pools, or floodways with heavy stand of timber and underbrush	0.075	0.100	0.150
2. Mountain streams, no vegetation in channel, banks usually steep, trees and brush along banks submerged at high stages			
a. bottom: gravels, cobbles, and few boulders	0.030	0.040	0.050
b. bottom: cobbles with large boulders	0.040	0.050	0.070
3. Floodplains			
a. Pasture, no brush			
1.short grass	0.025	0.030	0.035
2. high grass	0.030	0.035	0.050
b. Cultivated areas			
1. no crop	0.020	0.030	0.040
2. mature row crops	0.025	0.035	0.045
3. mature field crops	0.030	0.040	0.050
c. Brush			
1. scattered brush, heavy weeds	0.035	0.050	0.070
2. light brush and trees, in winter	0.035	0.050	0.060
3. light brush and trees, in summer	0.040	0.060	0.080
4. medium to dense brush, in winter	0.045	0.070	0.110
5. medium to dense brush, in summer	0.070	0.100	0.160
d. Trees			
1. dense willows, summer, straight	0.110	0.150	0.200

2. cleared land with tree stumps, no sprouts	0.030	0.040	0.050
3. same as above, but with heavy growth of sprouts	0.050	0.060	0.080
4. heavy stand of timber, a few down trees, little undergrowth, flood stage below branches	0.080	0.100	0.120
5. same as 4. with flood stage reaching branches	0.100	0.120	0.160
4. Excavated or Dredged Channels			
a. Earth, straight, and uniform			
1. clean, recently completed	0.016	0.018	0.020
2. clean, after weathering	0.018	0.022	0.025
3. gravel, uniform section, clean	0.022	0.025	0.030
4. with short grass, few weeds	0.022	0.027	0.033
b. Earth winding and sluggish			
1. no vegetation	0.023	0.025	0.030
2. grass, some weeds	0.025	0.030	0.033
3. dense weeds or aquatic plants in deep channels	0.030	0.035	0.040
4. earth bottom and rubble sides	0.028	0.030	0.035
5. stony bottom and weedy banks	0.025	0.035	0.040
6. cobble bottom and clean sides	0.030	0.040	0.050
c. Dragline-excavated or dredged			
1. no vegetation	0.025	0.028	0.033
2. light brush on banks	0.035	0.050	0.060
d. Rock cuts			
1. smooth and uniform	0.025	0.035	0.040
2. jagged and irregular	0.035	0.040	0.050
e. Channels not maintained, weeds and brush uncut			
1. dense weeds, high as flow depth	0.050	0.080	0.120
2. clean bottom, brush on sides	0.040	0.050	0.080
3. same as above, highest stage of flow	0.045	0.070	0.110
4. dense brush, high stage	0.080	0.100	0.140
5. Lined or Constructed Channels			
a. Cement			
1. neat surface	0.010	0.011	0.013
2. mortar	0.011	0.013	0.015
b. Wood			
1. planed, untreated	0.010	0.012	0.014
2. planed, creosoted	0.011	0.012	0.015
3. unplanned	0.011	0.013	0.015
4. plank with battens	0.012	0.015	0.018
5. lined with roofing paper	0.010	0.014	0.017

c. Concrete			
1. trowel finish	0.011	0.013	0.015
2. float finish	0.013	0.015	0.016
3. finished, with gravel on bottom	0.015	0.017	0.020
4. unfinished	0.014	0.017	0.020
5. gunite, good section	0.016	0.019	0.023
6. gunite, wavy section	0.018	0.022	0.025
7. on good excavated rock	0.017	0.020	
8. on irregular excavated rock	0.022	0.027	
d. Concrete bottom float finish with sides of:			
1. dressed stone in mortar	0.015	0.017	0.020
2. random stone in mortar	0.017	0.020	0.024
3. cement rubble masonry, plastered	0.016	0.020	0.024
4. cement rubble masonry	0.020	0.025	0.030
5. dry rubble or riprap	0.020	0.030	0.035
e. Gravel bottom with sides of:			
1. formed concrete	0.017	0.020	0.025
2. random stone mortar	0.020	0.023	0.026
3. dry rubble or riprap	0.023	0.033	0.036
f. Brick			
1. glazed	0.011	0.013	0.015
2. in cement mortar	0.012	0.015	0.018
g. Masonry			
1. cemented rubble	0.017	0.025	0.030
2. dry rubble	0.023	0.032	0.035
h. Dressed ashlar/stone paving	0.013	0.015	0.017
i. Asphalt			
1. smooth	0.013	0.013	
2. rough	0.016	0.016	
j. Vegetal lining	0.030		0.500