

Population reducer

Otto Hannuksela

June 4, 2015

1 Brief description of the code

The population reducer is meant for separating different populations within velocity distributions from each other. The aim of the code is to provide a way to analyze different populations in a robust way. For example, it might be important for the user to be able to separate backstreaming populations from core maxwellian populations.

2 Algorithm employed

2.1 Short description

The algorithm relies on locating *local maximas* within the velocity distributions. The idea is that we assume each velocity population has a local maximum somewhere, and that different populations are connected by a local minimum:

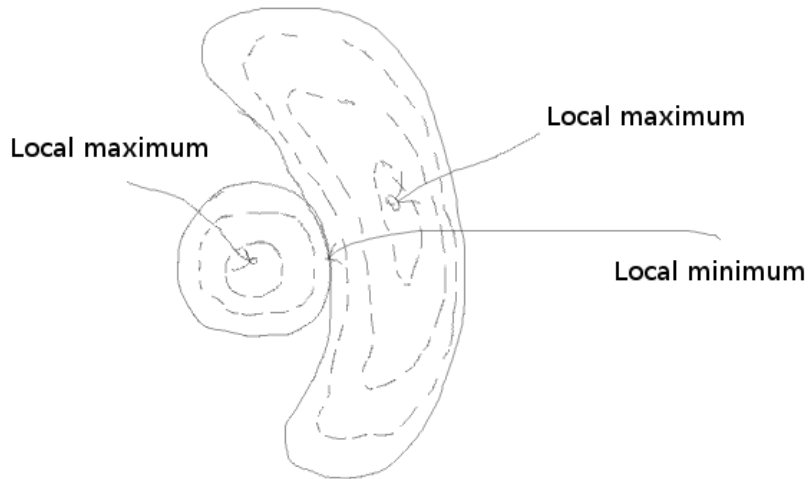


Figure 1: Two populations within the velocity space

In this case we can assume that by iterating through velocity space starting from the largest value and moving to the next-largest value on every iteration,

we finally hit the local minimum shown in Figure 1, and get two separate populations.

2.2 Pseudocode

Algorithm 1 Population algorithm

```

1: function POPULATION_REDUCER(cell, tolerance)
2:   vcells  $\leftarrow$  get_velocity_cells(cell)
3:   sort(vcells) based on their value
4:   Let populations be all populations
5:   for all velocity cells vcell in vcells do
6:     if vcell neighbor of any population in populations then
7:       if vcell neighbor of one population then
8:         population  $\leftarrow$  vcell
9:       else if vcell neighbor of more than one population in populations
10:      then
11:        Let the populations be population1, population2
12:        if population1 or population2 small then
13:          Merge(population1, population2)
14:        else
15:          population1  $\leftarrow$  vcell
16:        end if
17:      end if
18:    else
19:      population  $\leftarrow$  new_population()
20:      population  $\leftarrow$  vcell
21:      populations  $\leftarrow$  population
22:    end if
23:  end for
24: end function

```
