Model schematics

Biocellion main loop

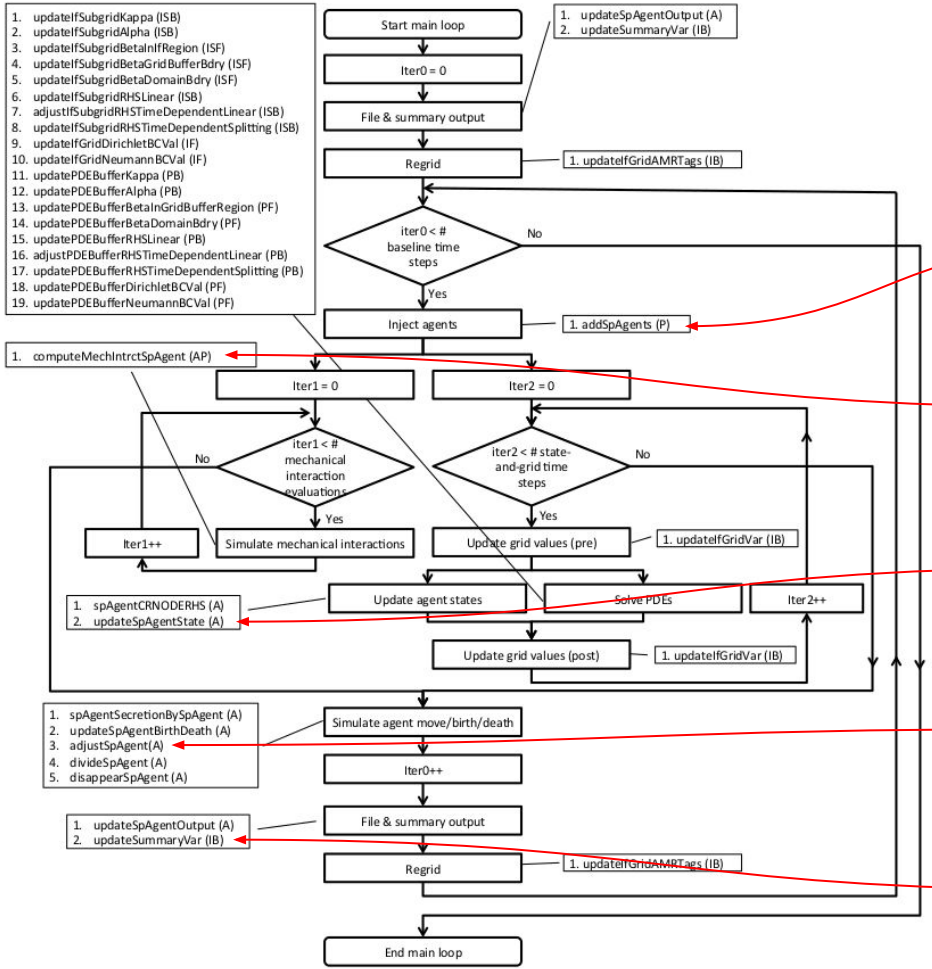


Figure 4.2. Main loop.

1. Cell input

2.1 Collision detection

2.2 Cell state update

3. Cell movement

4. Save LTo states

Setting parameters

- Done in model_define.h file.
- Set length of one time step in seconds (5 seconds in this case).
- Number of steps per minute can be calculated (60 / 5 = 12 in this case).
- Set maximum speed of H cells in micron per minute (8.8).
- Set minimum speed of H cells in micron per minute (3.8).
- Cell radii are set in micron (8 for H cells, 24 for LTo cell).
- Other parameters such as stable bind probability are set.

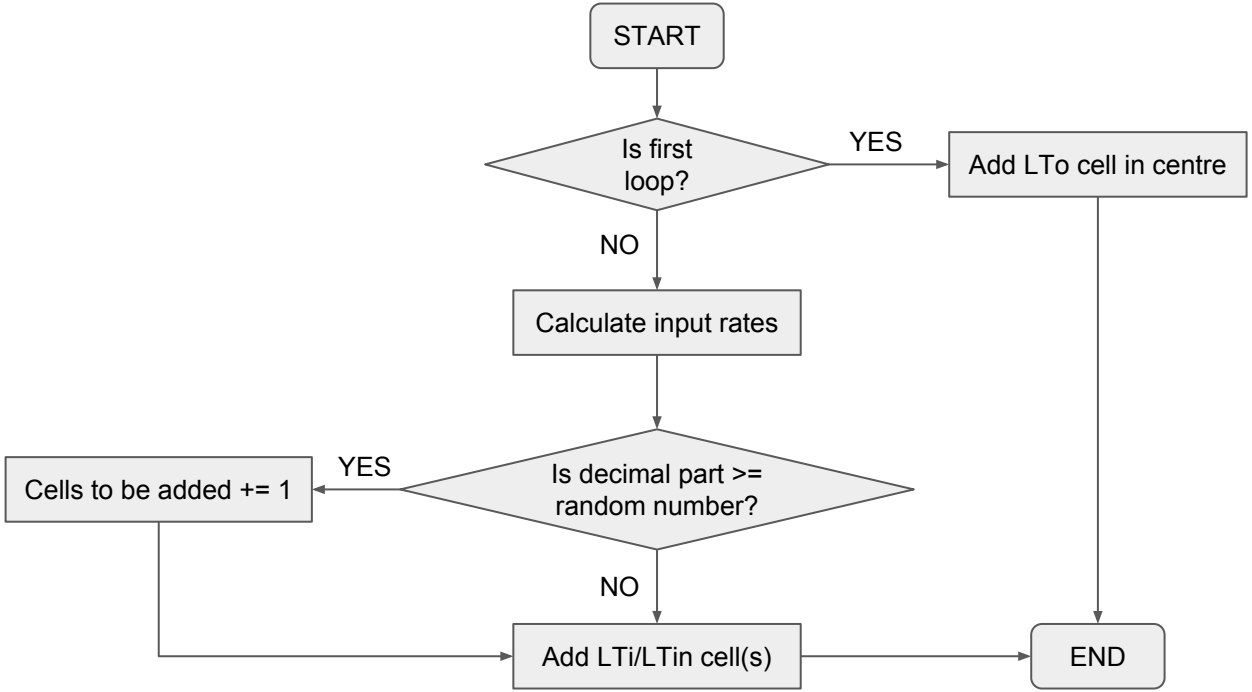
1. Cell input

- Done in addSpAgents() method in model_routine_agent.cpp file.
- 1 LTo cell is added in the centre of grid in the first loop.
- After the first loop, cell input rate is calculated for H cells (LTi and LTin cells) which is in cells per step.
- H cells are added to the grid every step at the calculated rate.
- In PPSim, decimal points in input rate are handled by keeping the sum of decimal points.
- For example, if input rate was 1.7, 1 cell will be added and 0.7 will be added to a field variable on the first loop. On the second loop, the 0.7 will be added again to the variable resulting 1.4. Since 1.4 is greater than 1, additional cell is added and 1 is subtracted from the variable, resulting 0.4.

Cell input continued

- Because of the restrictions in Biocellion, I do not have access to mutable global variables that persists throughout the simulation.
- Therefore, probabilistic model is used for cell addition.
- For example, if the input rate was 1.7, 1 cell will be added and there is probability of 0.7 for 1 additional cell to be added.

Simplified flow chart for cell input



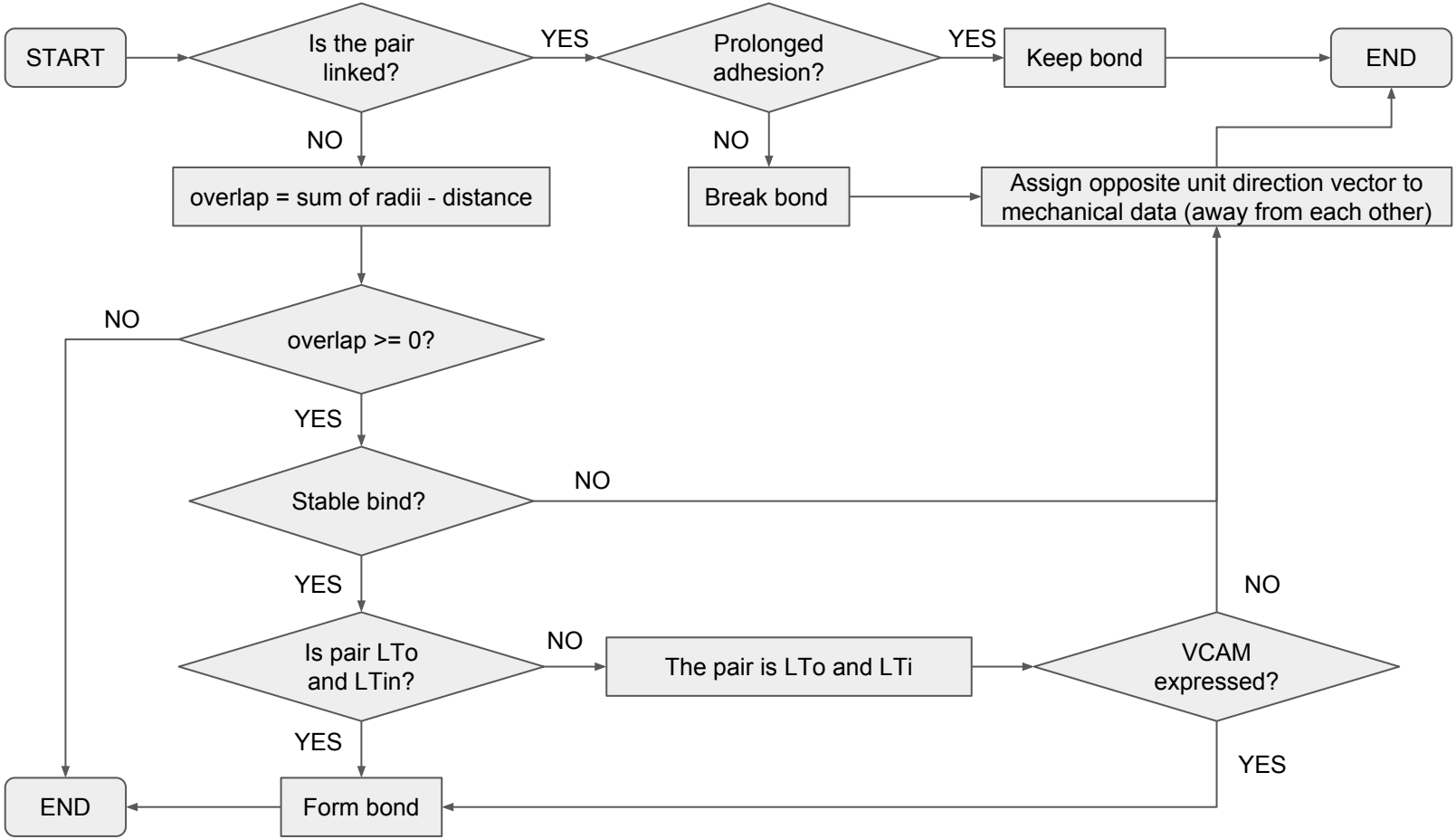
2.1 Collision detection

- Done in computeMechIntrctSpAgent() method in model_routine_mech_intrct.cpp file.
- Biocellion checks mechanical interaction between pairs of cells that are within maximum interaction distance which is set in model_define.h along with other parameters.
- Therefore prevents checking every single pair and saves computation time.

Collision detection continued

- It can save temporary mechanical data(persists only for the current loop) for each cell of the pair which can be used later in adjustSpAgent() method.
- Mechanical data is used for setting direction when collision occurs between two cells or when a stable bond is broken.
- A cell that collided with, or broken away from another will have the opposite movement direction from the corresponding cell.
- Bonds between cells can be formed or removed only in this method so collision detection cannot happen in cell movement stage.

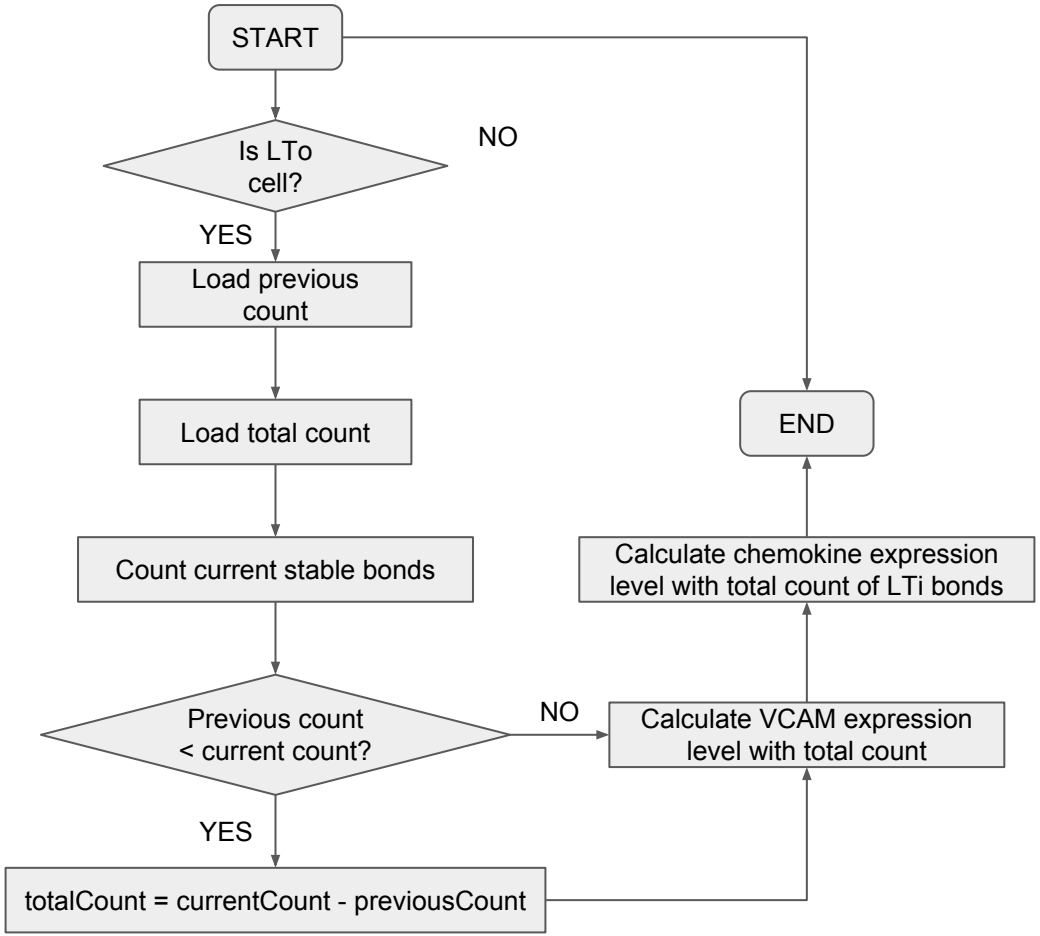
Simplified flowchart for collision detection



2.2 Cell state update

- Done in updateSpAgentState() method in model_routine_agent.cpp file.
- Used to update cell model specific state variables.
- Called for every agent in the simulation.
- In case of LTo cell, it updates the number of stable binds with H cells.
- Using the number of bonds, it also updates VCAM expression level and chemokine expression level.

Simplified flow chart for cell state update



3. Cell movement

- Done in adjustSpAgent() method in model_routine_agent.cpp file.
- Called for every agent in the simulation.
- The only method that allows cell movement.
- It can happen only once per step.
- In case of H cell, it checks the number of movements left for it to make a full minute movement and set new movement direction if it’s 0.
- If the cell is affected by chemokine, change its move direction towards LTo cell.
- It’s impossible to get a reference of another cell in the method, so the summary output function of Biocellion is exploited to retrieve LTo cell states such as location.

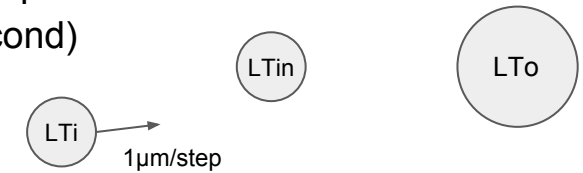
Cell movement continued

- It uses mechanical data passed from collision detection stage to change movement direction in case of collision or bond break.
- Direction change due to collision takes priority over chemokine attraction.
- Every move a cell makes requires movement steps.
- For example, if step length was 5 seconds and cell speed was 6 micron per minute, it should move 12 times (60 / 5 = 12) in a set direction at 0.5 micron per step (6 / 12 = 0.5), provided it makes no contact with another cell.
- If step length was 8 seconds, the number of step per minute would not be a whole number (60 / 8 = 7.5).
- In this case, the cell will move 8 times, but the 8th movement will have 0.5 * normal displacement to compensate for the fraction.

Cell movement example

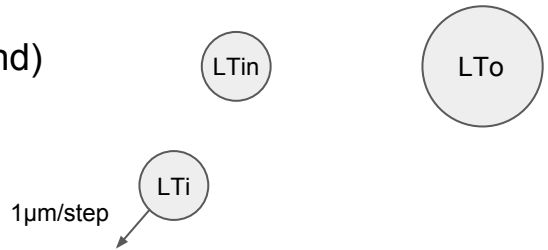
Assumes that
- Time step is 10 seconds
- Cell speed is 6 micron per minute
- LTi cell is always attracted chemokine
- LTin cell is stationary

Step 1
(10 second)

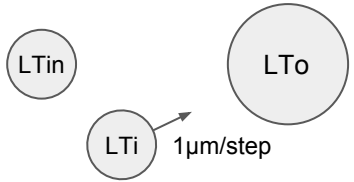


Step 2
(20 second)

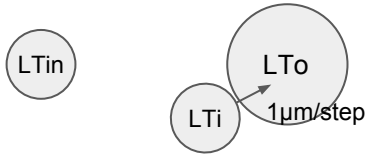
Step 3
(30 second)

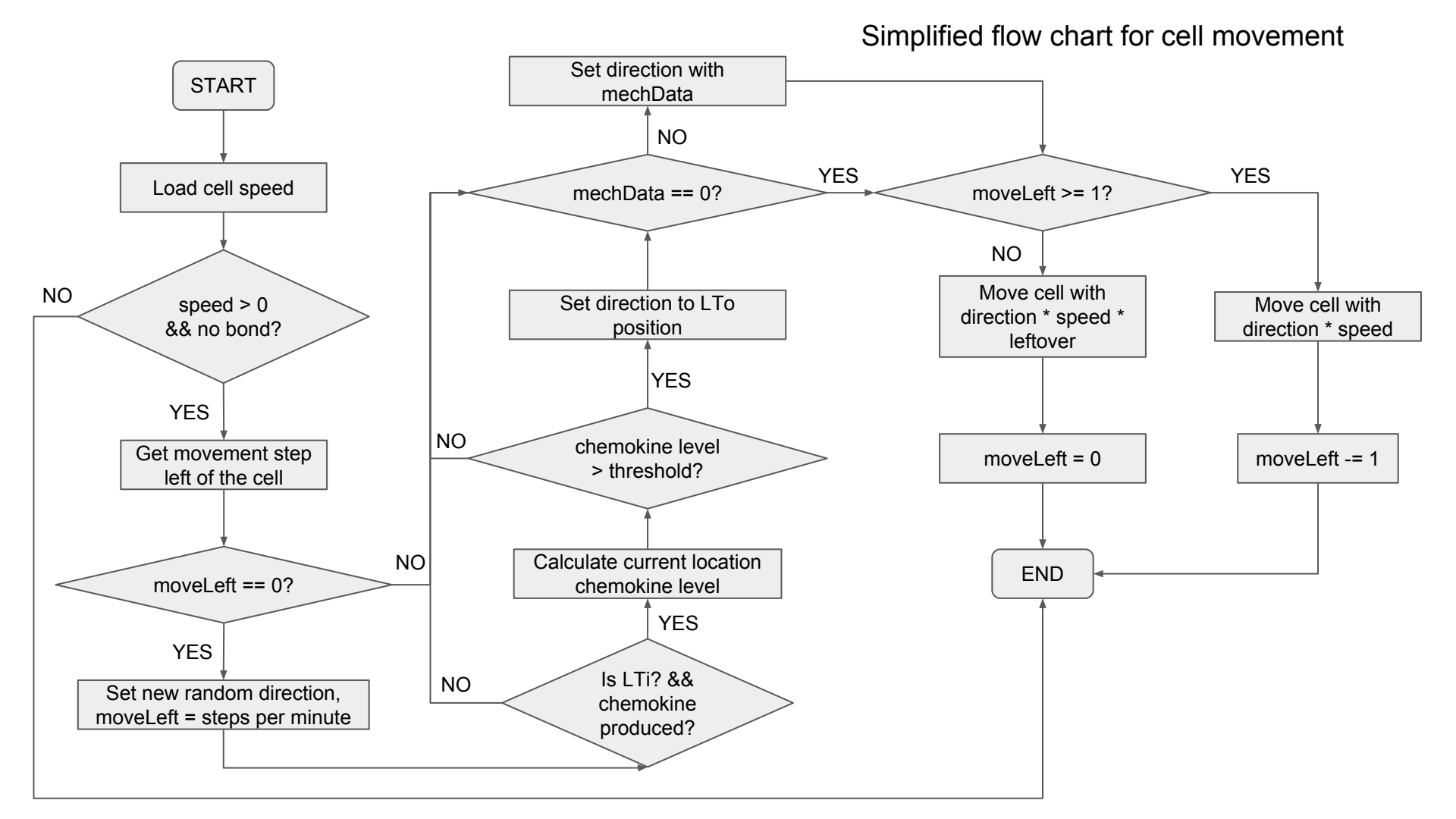


Step 4
(40 second)



Step 5
(50 second)




Simplified flow chart for cell movement
START
Load cell speed
speed > 0 && no bond?
NO
YES
Get movement step left of the cell
moveLeft == 0?
NO
YES
Set new random direction, moveLeft = steps per minute
Set direction with mechData
NO
mechData == 0?
YES
Set direction to LTo position
YES
chemokine level > threshold?
NO
Calculate current location chemokine level
YES
Is LTi? && chemokine produced?
NO
moveLeft >= 1?
YES
NO
Move cell with direction * speed * leftover
Move cell with direction * speed
moveLeft = 0
moveLeft -= 1
END