Overall strengths

- My code achieved the desired output and was overall very well organized, with descriptive comments and function docstrings. It is clear what each function does, and there is little extraneous code. I took care to make the implementation as efficient as possible, avoiding updating particles multiple times after a collision was detected.

Improving the code

- Some functionalities that I make use of a lot, such as updating particle positions to the most current time, could have their separate functions to cut down on duplicate code and save time.
- Calculating the wall collision time includes lots of if statements that could be reduced with more thought and care.

Thoughts

- I found this assignment to be much more challenging than the previous one. Overall, I thought my process of designing, writing, testing, and debugging the code was mostly solid. I first wrote the declarations for all the functions I thought I would need, then filled them in, and then went over them to make sure they were compatible. Then, I wrote the main loop and started fixing errors one by one (e.g., segmentation fault or wall collision times being incorrect). I tried utilizing a debugger, but I could not get it to stop at breakpoints (lldb debugger in macOS), so I resorted to printing statements again, which was a bit of a pain. I need to figure out how to efficiently debug my code. Other than that I would have saved some time if I had planned out how each function was going to interact before writing them.