This program creates a TPD simulation using a previously created KMC\_model. It updates the temperature using how much time has passed and beta. However, the time is determined by kmc\_time. There are certain checks that have to be met before the program exports, such as temperature increment, coverage amount, and tof comparison. If the values don't pass the checks, then the model gets reset, changed, and runs the simulation again with the new values. After the checks are passed, the program exports information, such as site coverage, temperature, time, etc.

I**nitilization (setting variables):**

**User Set Parameters (1 of 2):**

The starting and final temperatures(Ti and Tf), and the max and min steps are initialized using user input. Beta, spec(species), ncovs(number of different coverages), and nelems(number of different random seeds) are set.

**Random seed:**

The random seed is generated using the random class by a loop if there were multiple seeds created.

**Calling the simulate\_tpd function:**

This function is passed the starting and intial temperatures, beta, the coverage, the species, and the random seed.

**Double for loop that goes through the different coverages and seeds:**

**The for loop creates a seperate simulation for each coverage and seed that is stored in the covs and ensemble arrays.**

**\*Note: Another loop could be added to have different activation energies**

**simulate\_tpd function:**

**Initiating and opening file streams:**

The files to be exported to are created and opened. Then, some of the initial conditions are exported into the corresponding files(info.txt, representation.txt). Also, the headers are exported to the necessary files.

KMC\_model is initialized. The kmc\_settings.random\_seed is set to a random seed that was produced. Then, the initial temperature is set for the model.

The number of sites are put into a variable(nsites) using the command model.lattice.spuck()

The volume(total sites) is recorded using model.base.get\_volume()

**Initial configuration:**

The initial coverage (determined by the variable set in the initialization phase) is produced by using a loop that determines the number of sites needing to have a molecule adsorbed to it. Then, choosing a random site to adsorb a molecule to using random.choice(), then using model.\_put() to add the species to the chosen site. Then, the model configuration is retrieved and stored in the variable config.

**Starting the simulation:**

**User Set Parameters(2 of 2):**

The initial temperature, min and max steps are set. The initial temperature is set according the the number passed into the simulate\_tpd function. The min and max steps are set according to user input. The maxDeltaTpersnapshot is set, which determines the max temperature increase that can occur between each snapshots. The max\_covChange and min\_covChange are set, which determines the max and min coverage change that can occur between snapshots. The cov\_final is set, which determines the lowest amount of coverage that will happen before ending the loop. This breaking out of the loop is done so that the simulation doesn't go too far and crash. The previous configuration, temperature, and time is stored in case the simulation needs to be reset back a snapshot. The current and previous coverage are set so that the coverage change can be determined.

**Loop for increasing temperature and running steps:**

Next, a loop is entered to continue the simulation as long as the temperature is less than the final temperature and the current coverage isn't less than the final coverage allowed. During the loop, the temperature parameter is set, so that the rate constant can reflect that temperature. The model goes through some steps, determined by a series of checks. Then, the time is retrieved and used to increment the temperature.

**If the temperature increment is over maxTpersnapshot:**

Then the model configuration is set to the previous model configuration, the time is set to the previous time, and the steps are decreased by half. The **boolean for exporting(successfulresolution) is set to false**, and the simulation is ran again with the new set values.

**Else if the coverage change is over the max coverage change:**

**Then, the model configuration is set to the previous model configuration, the time is set to the previous time, and the steps are decreased by half. The boolean for exporting(successfulresolution) is set to false. and the simulation is ran again with the new set values.**

**Else if the coverage change is under the min coverage change:**

**Then, the steps are increased by double and the temperature is incremented. Then, the boolean for exporting(successfulresolution) is set to true.**

**Else:**

**Then, the temperature is incremented and the boolean for exporting(successfulresolution) is set to true.**

**If the current coverage is smaller than the set amount:**

**Change min steps to one and decrease the amount of steps by half. This is to prevent the simulation from stepping after all the molecules have desorbed, which causes it to crash.**

**If the steps are smaller than the min steps per snapshot allowed:**

**Then, the steps are set to the min steps.**

**Else if the steps are greater than the max steps per snapshot allowed:**

**Then, the steps are set to the max steps.**

**If a successful resolution occured**:

**If the coverage change isn't 0:**

**Set the desorbed boolean to true, and update the total desorbed variable.**

**If a molecule desorbed:**

Then, data about the current model is exported into two files(cov1(site\_coverages) and cov2(total\_coverages)). The temperature that is exported is the average of the previous temperature and the current temperature. This is to have a more accurate representation of what temperature the simulation would actually be using. Another way of making the temperature export more accurate, is by using the tofs to determine which temperature should be used. If the tof is increasing, then the temperature to be exported would be the previous temperature. If the tof is decreasing, then the temperature to be exported would be the current temperature.

**Finalize:**

After the loop has been executed(either the final temperature has been reached or the molecules have been desorbed), the model is deallocated and the files are closed.