Supporting Information for "Generalized Temporal Acceleration Scheme for Kinetic Monte Carlo Simulations of Surface Catalytic Processes by Scaling the Rates of Fast Reactions"

Eric C. Dybeck[†], Craig P. Plaisance[†], Matthew Neurock[†‡]

†Department of Chemical Engineering, University of Virginia, Charlottesville, VA

‡Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN

Additional Details for the Fischer-Tropsch Synthesis Reaction

The rate of individual surface reaction events, such as CO* inserting into CH₃*, are calculated using the familiar equation from Transition State Theory (TST):

(S1)

$$k = \frac{k_B T}{h} e^{\left[\frac{\Delta S^{\neq}}{R}\right]} e^{\left[-\frac{\Delta E^{\neq}}{RT}\right]}$$

where k is the rate, k_B is the Boltzmann constant, T is the temperature, h is Plank's constant, $\Delta S^{\neq}$ and $\Delta E^{\neq}$ are the entropy change and energy change between the reactant state and transition state, and R is the ideal gas constant. Molecular adsorption events, such as gas phase CO adsorbing on to a metal site, are assumed to be non-activated processes and the rate constants are calculated from collision theory using:

(S2)

$$k = \frac{P_i A S_i}{\sqrt{2\pi M_i RT}}$$

Where P_i is the gas phase pressure of reactant i, A is the cross-sectional area of the reactive site, S_i is the sticking coefficient of reactant i (assumed to be unity), and M_i is the molecular weight of reactant i. To ensure that microscopic reversibility is preserved, the rate constant of the reverse reactions for all processes are calculated using:

(S3)

$$k_{rev} = k_{for} K_{eq}^{-1} = k_{for} e^{\left[-\frac{\Delta S}{R}\right]} e^{\left[\frac{\Delta E}{RT}\right]}$$

where k_{rev} is the rate constant of the reverse reaction, k_{for} is the forward reaction rate constant, K_{eq} is the equilibrium constant of the process, and ΔS and ΔE are the differences in entropy and energy between the reactant and product state of the forward reaction.

The Kinetic Monte Carlo simulation algorithm was carried out using a code recently developed to model catalytic reactions occurring on metal nanoclusters. The details of this model will be elaborated in a subsequent manuscript. Briefly, the catalyst is modeled as a 4.2 nm cubo-octahedral particle built out of an infinite ruthenium lattice (shown in Figure S1). The surface reactions are allowed to occur on the exposed (111) terraces of the nanoparticle, consistent with theoretical and experimental observations [46,52]. This particle consists of 2406 Ruthenium atoms, with 752 atoms on the surface of the particle and 488 of them residing on the exposed 111 terraces. The surface bound intermediates CO*, CH_x* (x=1, 2, 3), and C_2H_y* (y=0, 1, 2, 3) occupy a single site. Larger intermediates such as *R-CO* and *R-COH* (R=H, CH_3) sit di-sigma and block 2 adjacent surface atop sites. Theoretical investigations have shown that surface hydrogen species (H*) prefer to sit in the 3-fold fcc sites in between the ruthenium surface atoms and interact very weakly with all nearby intermediates. This is consistent with experimental infrared spectroscopy studies that suggest the presence of chemisorbed surface hydrogen does not inhibit carbon monoxide uptake [52]. In addition, hydrogen represents a minority species on the surface under all conditions of practical interest. We have therefore chosen to model all hydrogen addition and elimination reaction as referenced back to the gas phase molecular hydrogen in place of modeling H* explicitly. The rate constants for hydrogen addition reactions are calculated using:

$$(S4)$$

$$k = P_{H_2} \frac{k_b T}{h} e^{\left[-\frac{S_{H_2}}{2R}\right]} e^{\left[-\frac{\Delta E_{H_2}}{2RT}\right]} e^{\left[\frac{\Delta S^{\neq}}{R}\right]} e^{\left[-\frac{\Delta E^{\neq}}{RT}\right]}$$

where P_{H_2} is the partial pressure of hydrogen, ΔE_{H_2} is the adsorption energy of H_2, and S_{H_2} is the entropy of gas phase H_2.
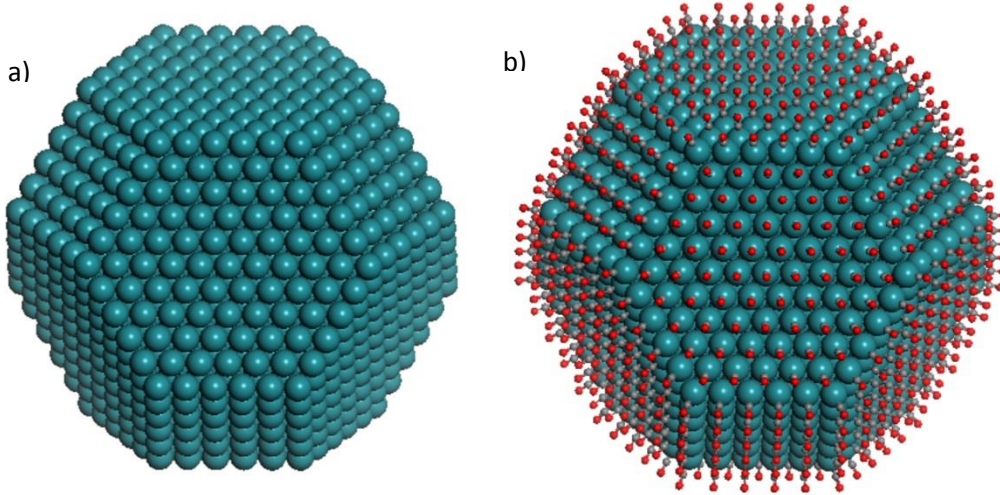
Figure S1. Model 4.2 nm metal particle consisting of 2406 Ruthenium atoms, shown both (a) bare and (b) at 1ML coverage of carbon monoxide. CO adsorption and desorption events occur at all surface metal atoms. The Fischer-Tropsch synthesis reaction steps proceeds on the exposed (111) terraces.

The kinetic inputs to the simulation have been loosely adopted from the DFT energetics calculated in ref 47. The activation barriers have been modified to more closely match experimental values; however the overall mechanism and rate-limiting step remain the same. A summary of the energetic and entropic barriers for each elementary reaction is shown in Table 1. For simplicity, the effects of lateral interactions on the activation barriers are not included, however this coarse-graining algorithm could easily be extended to systems where repulsive interactions influence kinetics. We also briefly note that the intrinsic forward and reverse rate constants for each reaction include a factor describing the number of nearest-neighbor sites that can facilitate the reaction. This factor is specific to the lattice being used to model the reactions and is necessary to match the KMC simulation data with the mean-field analytical model. Atop sites on (111) terraces have 6 nearest neighbor atop sites (included for additions between atop-bound intermediates) and 3 nearest neighbor 3-fold sites (used for additions and removal of hydrogen).

Table S1. Elementary reactions along with energetic and entropic changes between the reactant state, transition state, and product state.

	Reaction	$\Delta E^{\neq}$	$\Delta S^{\neq}$	ΔE	ΔS
1	$CO_{(g)} + * \rightleftharpoons CO^*$	0.0	0.0	-80.0	-163.5
2	$CO^* + * + \frac{1}{2}H_{2(g)} \rightleftharpoons {}^*HCO^*$	37.2	-118.7	25.2	-118.7
3	${}^*HCO^* + \frac{1}{2}H_{2(g)} \rightleftharpoons {}^*HCOH^*$	38.0	-71.7	5.6	-71.7
4	${}^*HCOH^* \rightleftharpoons CH^* + OH^*$	17.2	0.0	-43.8	0.0
5	$CH^* + \frac{1}{2}H_{2(g)} \rightleftharpoons CH_2^*$	25.0	-71.7	-10.0	-71.7
6	$CH_2^* + \frac{1}{2}H_{2(g)} \rightleftharpoons CH_3^*$	25.0	-71.7	-10.0	-71.7
7	$CH_3^* + \frac{1}{2}H_{2(g)} \rightarrow CH_{4(g)} + *$	25.0	-71.7	-	-
8	$CO^* + CH_3^* \rightarrow {}^*CH_3CO^*$	30.0	-47.0	-38.0	-47.0
9	${}^*CH_3CO^* + \frac{1}{2}H_{2(g)} \rightarrow {}^*CH_3COH^*$	20.0	-71.7	-40.0	-71.7
10	${}^*CH_3COH^* \rightarrow CH3C^* + OH^*$	20.0	0.0	-40.0	0.0
11	$CH_3C^* + \frac{1}{2}H_{2(g)} \rightleftharpoons CH_3CH^*$	25.0	-71.7	-10.0	-71.7
12	$CH_3CH^* + \frac{1}{2}H_{2(g)} \rightleftharpoons CH_3CH_2^*$	25.0	-71.7	-10.0	-71.7
13	$CH_3CH_2^* + \frac{1}{2}H_{2(g)} \rightarrow CH_3CH_{3(g)} + *$	25.0	-71.7	-	-
14	$OH^* + \frac{1}{2}H_{2(g)} \rightarrow H_2O_{(g)} + *$	20.0	-71.7	-	-

Mean-Field Kinetic Model of the Fischer-Tropsch Synthesis Process

The overall scheme for the Ficher-Tropsch Synthesis process is shown in Figure S2 and is reproduced from the main text Figure 5. Initially, gas phase CO adsorbs on to the surface. Assuming that CO* is the most abundant surface intermediate, and that CO* is quasi-equilibrated with the gas phase, the surface concentration of CO* and free sites * can be represented as:

(S5)

$$[CO^*] = \frac{K_{CO}P_{CO}}{1 + K_{CO}P_{CO}}$$

(S6)

$$[*] = \frac{1}{1 + K_{CO}P_{CO}}$$

Hydrogen adsorbs noncompetitively from the gas phase and participates in multiple subsequent hydrogen addition steps. Because hydrogen is equilibrated with the surface at all conditions examined in this study and the concentration is low, the surface hydrogens are references back to the gas phase reactants by:

(S7)

$$[H^*] \approx \sqrt{K_{H_2}P_{H_2}}$$

The rate of the unequilibrated hydrogen addition to the *HCO* species is related to the surface concentration of CO* and hydrogen pressure using the equilibrium of the first two mechanistic steps.

(S8)

$$r_3 = k_3\sqrt{P_{H_2}}[\,^*HCO^*] = k_3K_2K_1P_{H_2}[CO^*][*]$$

Substituting the results from eqs. (S5) and (S6) into eq. (S6) leads to the rate of CO-activation presented in eq. (14) of the main text:

(S9)

$$r_3 = \frac{k_3K_2K_1P_{H_2}P_{CO}}{(1 + K_1P_{CO})^2}$$

The selectivity to methane is the ratio of the methane production to methane and ethane production in the FTS reaction. Both the hydrogen addition to CH_3 and the hydrogen addition to CH_3CH_2 irreversibly produce methane and ethane, respectively. However, we observe that the CO insertion into CH_3 is

unequilibrated and essentially always leads to the production of ethane. Therefore, the rate of ethane production is equal to the rate of the CO insertion into surface CH_3. The selectivity to methane is therefore:

(S10)

$$X_{CH_4} = \frac{r_7}{r_7 + r_8} = \frac{k_7\sqrt{P_{H_2}}[CH_3^*]}{k_7\sqrt{P_{H_2}}[CH_3^*] + k_8[CH_3^*][CO^*]} = \frac{k_7\sqrt{P_{H_2}}}{k_7\sqrt{P_{H_2}} + k_8[CO^*]}$$

Substituting the results from eqs. (S5) and (S6) into eq. (S10) leads to the selectivity to methane presented in eq. (15) of the main text:

(S11)

$$X_{CH_4} = \frac{r_7}{r_7 + r_8} = \frac{k_7\sqrt{P_{H_2}}}{k_7\sqrt{P_{H_2}} + \dfrac{k_8 K_1 P_{CO}}{1 + K_1 P_{CO}}}$$
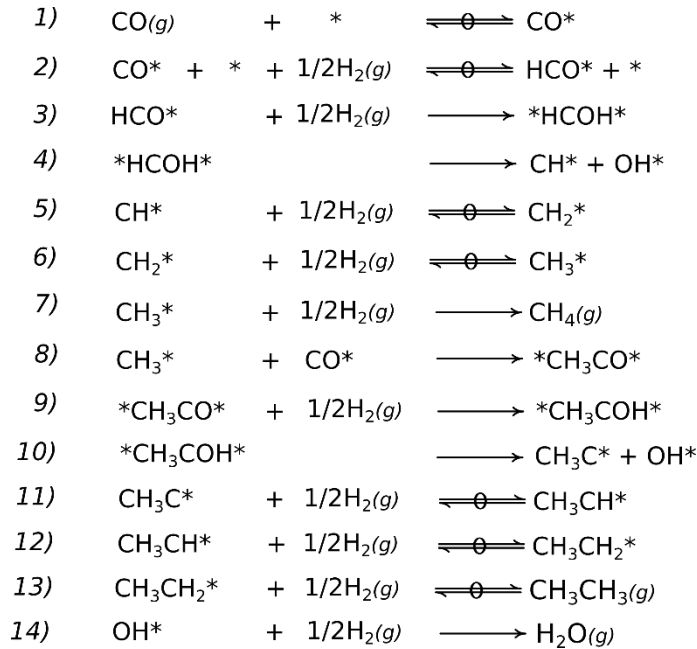
1) $CO_{(g)}$ + * $\rightleftharpoons$ CO*

2) CO* + * + $1/2H_{2(g)}$ $\rightleftharpoons$ HCO* + *

3) HCO* + $1/2H_{2(g)}$ $\longrightarrow$ *HCOH*

4) *HCOH* $\longrightarrow$ CH* + OH*

5) CH* + $1/2H_{2(g)}$ $\rightleftharpoons$ CH_2*

6) CH_2* + $1/2H_{2(g)}$ $\rightleftharpoons$ CH_3*

7) CH_3* + $1/2H_{2(g)}$ $\longrightarrow$ $CH_{4(g)}$

8) CH_3* + CO* $\longrightarrow$ *CH_3CO*

9) *CH_3CO* + $1/2H_{2(g)}$ $\longrightarrow$ *CH_3COH*

10) *CH_3COH* $\longrightarrow$ CH_3C* + OH*

11) CH_3C* + $1/2H_{2(g)}$ $\rightleftharpoons$ CH_3CH*

12) CH_3CH* + $1/2H_{2(g)}$ $\rightleftharpoons$ CH_3CH_2*

13) CH_3CH_2* + $1/2H_{2(g)}$ $\rightleftharpoons$ $CH_3CH_{3(g)}$

14) OH* + $1/2H_{2(g)}$ $\longrightarrow$ $H_2O_{(g)}$

Figure S2. Reaction network for Fischer-Tropsch synthesis.

Optional Subroutines to Improve Efficiency

The procedure described in the main text of unscaling all reactions and re-equilibrating the quasi-equilibrated reactions after an unequilibrated reaction event rigorously ensures simulation accuracy. However, this also significantly reduces the simulation efficiency during the initial approach to equilibrium where many fast processes are still considered unequilibrated. To improve the efficiency, one can optionally bypass the reaction unscaling if the unequilibrated process belongs to the same reaction channel as the previous unequilibrated process executed. If the unequilibrated process just executed belongs to the same reaction channel as the last unequilibrated reaction executed before it, then it is likely that the rates of the reaction channels in the new superbasin are similar to those in the previous superbasin such that the previous scaling factors are still acceptable. This is due to the fact that the stiffness that necessitates rescaling and re-equilibration arises primarily when a fast unequilibrated reaction immediately follows a slow unequilibrated reaction (i.e., two *different* unequilibrated reactions).

If the reactions are not unscaled, due to the above scenario, the scaling factors cannot be updated until the reaction rates have been sampled with sufficient accuracy. We therefore use the scaling factors from the previous superbasin until the system has reached equilibration in the current superbasin (all of the quasi-equilibrated reactions with non-zero rates have executed n_e times in the superbasin). At this point, we have adequately sampled the new superbasin and the scaling factors can be updated accurately using eq. **Error! Reference source not found.**. If the system exits the superbasin before equilibration has occurred and the unequilibrated process by which it exits is from the same reaction channel as the previous unequilibrated step, then the scaling factors for those reactions that have been sufficiently executed in the current superbasin are updated just prior to exiting for use in the new superbasin. For those quasi-equilibrated reactions that have not yet been sufficiently executed, the scaling factors are reset to unity. The entire unscaling procedure with scaling factor freezing is depicted in Figures S3 and S4.
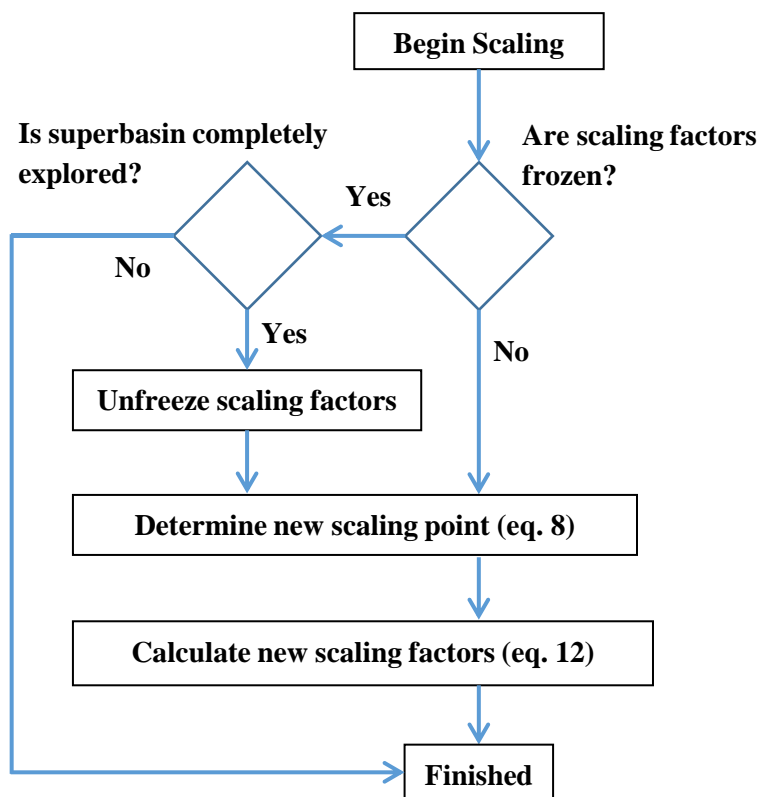
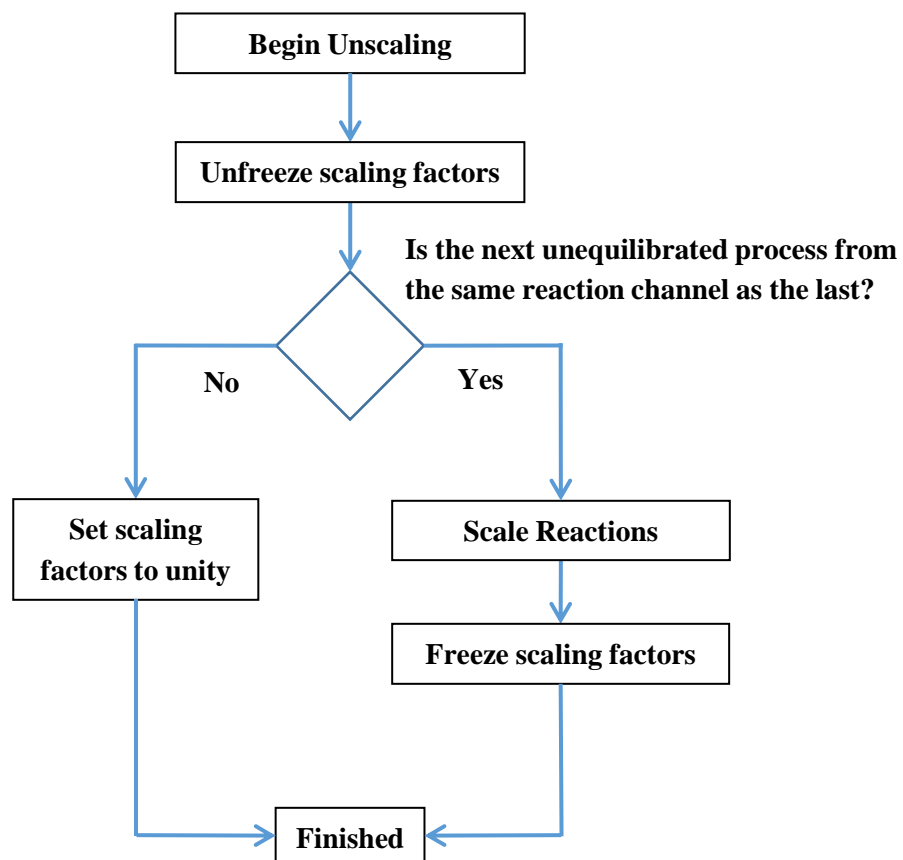**Figure S1.** Flow diagram depicting the reaction scaling procedure

Figure S2. Flow diagram depicting the unscaling procedure carried out after execution of an unequilibrated process

Pseudocode for the Proposed Reaction Scaling Algorithm

 # Initialize step, time, scaling factors, and reaction counters

$i = 0$ # simulation step

$t = 0$ # simulation time

For each reaction m in *all_reactions*:

    $\alpha_m = 0$ # scaling factor

    $N_m = 0$ # n_e element array containing execution history

    $p_m = 0$ # current position in array N_m

    $n_{e,m} = 0$ # number of executions in current superbasin

    $r_m = 0$ # rate of forward reaction

    $r_{-m} = 0$ # rate of reverse reaction

    Flag reaction m as not *sufficiently_executed*

    Flag reaction m as *unequilibrated*

Next reaction

 # Main loop

Do main_loop:

    # Populate the Event Table

    For each site s in *all_sites*:

        For each reaction m in *all_reactions*:

            If reaction m is possible for site s and this configuration in the forward/reverse direction:

                Add a new event e to list *all_events* for reaction m in the forward/reverse direction with rate constant k_e and scaling factor $\alpha_e = \alpha_m$

        Next reaction

Next site

 # calculate the rate constants of all reaction channels from current configuration

For each reaction *m* in *all_reactions*:

$$k_{m,i} = 0$$

$$k_{-m,i} = 0$$

    For each event *e* in *all_events*:

        If event *e* is a reaction of type *m* in forward direction:

$$k_{m,i} = k_{m,i} + k_e$$

        If event *e* is a reaction of type *m* in reverse direction:

$$k_{-m,i} = k_{-m,i} + k_e$$

    Next event

Next reaction

 # pick an event to occur at this step

$RN = \mathrm{rand}(0,1)$ # uniform random number between 0 and 1

$k_{sum} = 0$

$counter = 0$

For each event *e* in *all_events*:

$$k_{sum} = k_{sum} + \eta_e k_e$$

Next event

For each event *e* in *all_events*:

$$counter = counter + \frac{\alpha_e k_e}{k_{sum}}$$

    If $(counter > RN \times k_{sum})$:

        selected_event = e

Next event

 # unscale reactions when an unequilibrated reaction is executed

If *selected_event* is an *unequilibrated* reaction:

$$N_{\text{superbasin}} = i$$

Call unscale_reactions()

Update configuration based on *selected_event*

 # increment the reaction counters and determine reversibility/equilibration of selected reaction channel

Set *m* to the reaction channel of *selected_event*

If *selected_event* occurs in forward direction of reaction channel *m*:

$$N_m(p_m) = 1$$

Else:

$$N_m(p_m) = -1$$

$$p_m = \text{mod}(p_m + 1, n_e)$$

$$n_{e,m} = n_{e,m} + 1$$

If $\sum |N_m| = n_e$ and $|\sum N_m| < \delta \, n_e$:

Flag reaction *m* as *quasi-equilibrated*

If $n_{e,m} \geq n_e$:

Flag reaction *m* as *sufficiently_executed*

Else:

Flag reaction *m* as *unequilibrated*

Flag reaction *m* as not *sufficiently_executed*

 # increment simulation time

$RN = \text{rand}(0,1)$ # uniform random number between 0 and 1

$$\Delta t_i = -\frac{\ln(RN)}{k_{sum}}$$

$$t = t + \Delta t_i$$

 # update reaction rates

For each reaction m in *all_reactions*:

$$r_m = r_m + k_{m,i}\Delta t_i$$

$$r_{-m} = r_{-m} + k_{-m,i}\Delta t_i$$

Next reaction

 # rescale reactions every N_s steps

If $(\mathrm{mod}(i, N_s) = 0)$ **And** $(i - N_{superbasin}) > N_s$:

Call scale_reactions()

 # increment simulation step

$$i = i + 1$$

Next main_loop

#Subroutine to unscale all reactions

Subroutine unscale_reactions():

 # reset reaction rates and counters

For each reaction m in *all_reactions*:

$$n_{e,m} = 0$$

$$r_m = 0$$

$$r_{-m} = 0$$

Flag reaction m as not *sufficiently_executed*

Next reaction

unfreeze scaling factors in case they were frozen

rescaling = true

reset or update scaling factors

different behavior depending on whether next unequilibrated
reaction is identical to the last unequilibrated reaction

If next unequilibrated reaction is different from last
unequilibrated reaction:

>   # unscale all reactions

>   **For** each reaction *m* in *all_reactions*:

>> $\alpha_m = 1$

>   **Next** reaction

Else:

>   # update and freeze scaling factors

>   **Call** scale_reactions()

>   *rescaling = false*

End Subroutine

#Subroutine to scale the reactions

Subroutine scale_reactions()

if scaling factors are frozen, start updating them again when
all quasi-equilibrated reactions have been sufficiently executed

If *rescaling* is *false*:

>   **If** all quasi-equilibrated reactions are
    sufficiently_executed:

>>   *rescaling = true*

>   **Else:**

Exit Subroutine

 # calculate the rate of escape from the current superbasin

$r_\tau = 0$

For each reaction m in *all_reactions*:

    If reaction m is not *sufficiently_executed*:

        $r_\tau = r_\tau + r_m + r_{-m}$

Next reaction

 # calculate scaling factors

For each reaction m in *all_reactions*:

    If reaction m is *sufficiently_executed*:

$$\alpha_m = \min\left(N_f \frac{2\, r_\tau}{r_m + r_{-m}}, 1\right)$$

    Else:

$$\alpha_m = 1$$

Next reaction

End Subroutine