Supplementary Material

NanoParticleFlowReactor(NanoPFR): A tested model for simulating carbon nanoparticle formation in flow reactors

Neil A. Juan[a], Ali Naseri [a], Mohammad Reza Kholghy [b], Murray J. Thomson[a1]

[a]*Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Ontario, M5S 3G8, Canada*

[b]*Department of Mechanical and Aerospace Engineering, Carleton University, 1125 Colonel By Dr, Ottawa, ON, K1S 5B6, Canada*

[1]Corresponding author: murray.thomson@utoronto.ca

S.1. Using NanoPFR

S.1.1. Downloading and Installing NanoPFR

Currently, the NanoPFR code is not available publicly and must be requested from the corresponding author at the University of Toronto. A link will be provided to the private GitHub repository that hosts the code. Upon receipt, the program exists as a package containing detailed documentation on code usage, sample input and mechanism files, and a directory of source files. Additional requirements to run the code includes the GNU Fortran compiler(GFortran) and a Linux shell or similar. The GFortran compiler supports standard Fortran 95 code and legacy Fortran 77 code and can be downloaded from. The program functions in any operating system given access to a Linux type terminal.

S.1.2. Code Structure and Instructions

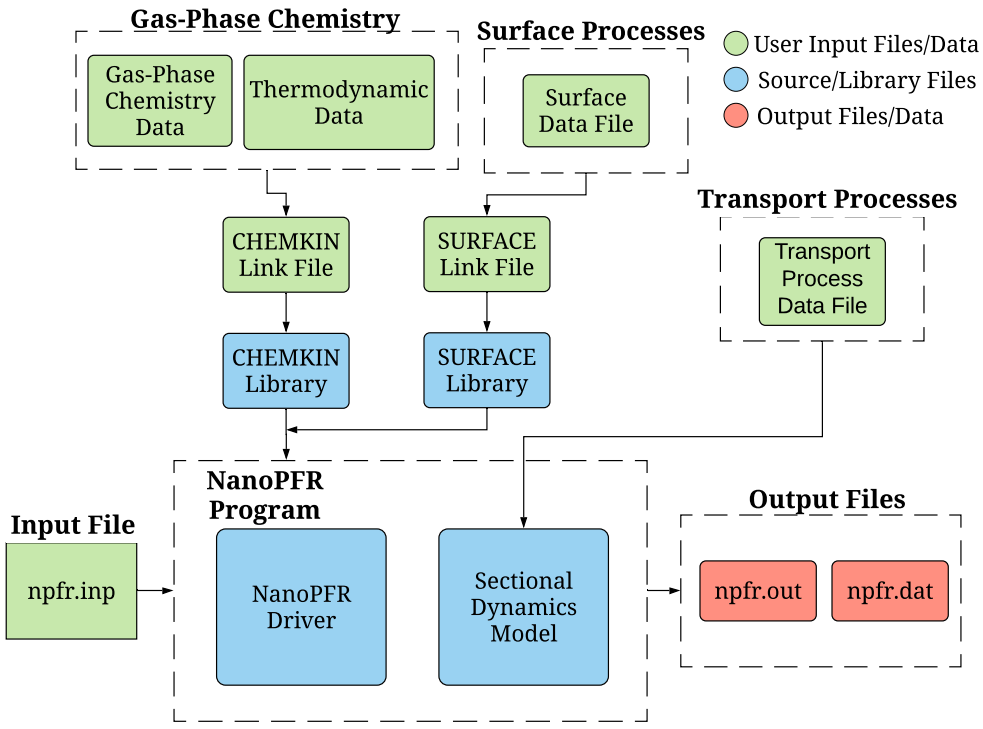


FIGURE S1. The required files and layout for the NanoPFR code. The users must provide the input files (green) to the source files/libraries (blue). After the simulation finishes, it will generate the output files (red).

Fig. S1 depicts the overall structure of the program and the required files to run it. The previous sections mention the required mechanism files from the user input. These files contain the gas-phase chemistry/thermodynamic data and the transport data. The first file

must be named `chem.asc` and arranged in CHEMKIN format, which will define the linking file. This linking file provides the species and thermodynamic information directly to the CHEMKIN library subroutines, which is accessed by the NanoPFR driver code. Similarly, the transport and surface linking files use the CHEMKIN format and are named `tran.asc` and `surf.asc`, respectively. The transport file provides information on the properties of the nucleating and adsorbing PAH species such as the Lennard-Jones collision diameter in angstroms.

The NanoPFR program consists of the NanoPFR driver program and the aerosol dynamics code. The driver program reads in the file, initializes and updates the main arrays, and carries out the main algorithm by solving the gas-phase chemistry and particle characteristics. The aerosol dynamics code utilizes the fixed pivot sectional method to generate particle source terms for the various carbon nanoparticle formation processes. The driver program calls on the aerosol dynamics code to determine the particle dynamics. An important note is that the main development of this study is the sectional aerosol dynamics model and further development of the driver program which is based on an existing plug flow reactor model [39].

S.1.3. User Defined Files

After downloading all the required programs, the usage of the NanoPFR code is straightforward. The process of performing a simulation includes defining required user-input files and using the terminal to compile and run the program. The following section illustrates this process and covers a sample case that will simulate the results in section 5.2. Using the NanoPFR code starts with unzipping and opening the package through a terminal. There are two directory levels in the NanoPFR package; the top-level has the input/output files and stores the executable; the second is a subdirectory named source that stores the source files along with the sectional model and reactor parameters.

The program requires user input in the top-level and the subdirectory. The files in the top-level include the mechanism files and the input data. The mechanism files were discussed in a previous section and take the form of a text file compatible with CHEMKIN libraries. The input data must be a text file arranged in keyword/value(s) pairs. The keywords specify the reactor dimensions, process parameters for the mixture properties, and solution parameters for the solver. The input file has a specific structure, where each line begins with a $3-4$ letters keyword followed by the corresponding values. The code documentation accompanying this paper provides greater detail and covers the full set of relevant keywords for this input file. The following shows a sample input file for the experimental validation in section 5.2.

npfr.inp

```
1.      XSTR   0.        ! Reactor start, cm
2.      XEND   64.       ! Reactor end, cm
3.      DIAM   1.1       ! Reactor diameter, cm
4.      TFIX
5.      DX     1.E-2     ! Spatial step, cm
6.      PRES   1.E+00    ! Pressure, atm
7.      VEL    49.15     ! Velocity, cm/s
8.      MOLE
9.      GAS C2H4  0.0100
10.     GAS N2   0.9900
11.     ATOL 1.E-8
12.     RTOL 1.E-6
13.     SFAC 1.0D0       ! Soot Factor
14.     SURF 1.0D0       ! Surface Reactivity
15.     COAG 1.0D0       ! Coagulation
16.     PADD 1.0D0       ! PAH Addition
17.     NPAH 165         ! Nucleating species 1
18.     NPAH 188         ! Nucleating species 2
19.     NPAH 191         ! Nucleating species 3
20.     CPAH 165         ! Adsorbing species 1
21.     CPAH 188         ! Adsorbing species 2
22.     CPAH 191         ! Adsorbing species 3
23.     NNEG
24.     END
```

The first 3 lines `XSTR`, `XEND` and `DIAM` specifies the start, end, and diameter of the reactor with units of cm. `TFIX` is the keyword defining the use of a temperature profile in the driver program, and `DX` is the spatial steps for iterating the solution in cm. An equivalent time step `D`τ can be derived using Eq. 5 to provide the time interval between each iteration which is used in the grid independence anaylsis in section 4.1.3. `PRES` defines the pressure in the reactor in atm and `VDOT` defines the volumetric flow rate in $\frac{cc}{s}$. Lines 8 – 10 specify the inlet gas mixture in mole fractions using the keyword `MOLE`. `ATOL`, and `RTOL` are tolerances for errors in the DASSL algorithm. Lines 13 – 16 specify the activation of specific carbon nanoparticle formation processes. These keywords take the value of 0 for inactive or 1 for active but can also act as efficiencies using values in between. `SFAC` defines soot factor, which is the generation of particles, `SURF` activates the surface reactivity given by Eq. 28, and `COAG` activates coagulation given by Eq. 33 and 34, and `PADD` activates PAH addition

given by Eq. 23 and 24. The next 2 lines include the keywords `NPAH` and `CPAH` which identify the indexes of the nucleating species and adsorbing species. The code is capable of using 3 different nucleating and adsorbing species, however, in this case, only pyrene is included. `NNEG` restricts the DASSL to only non-negative solutions, and `END` signifies the end of the input file. Table S1 summarizes the keywords used in `npfr.inp`. Further details on the possible keywords, descriptions, and units for the input file is provided in the documentation accompanying this paper.

TABLE S1. Input file description for `npfr.inp`

Variable Name	Description	Units
XSTR	Inlet axial position `XSTR`	cm
XEND	Outlet axial position	cm
DIAM	Reactor cross section diameter (constant)	cm
TFIX	Flag specifying use of axial temperature profile	N/A
DX	Distance interval	cm
PRES	Inlet pressure	atm
VEL	Inlet velocity	cm/s
MOLE	Flag specifying inlet gas composition is in mole fractions	N/A
GAS	Inlet gas-phase composition	Mole Fraction (mol/mol)
ATOL	Absolute error tolerance for DASSL	N/A
RTOL	Relative error tolerance for DASSL	N/A
SFAC	Soot factor specifies particle generation	N/A
SURF	Surface reactivity specifies surface growth and oxidation	N/A
COAG	Coagulation specifies particle agglomeration	N/A
PADD	PAH addition specifies surface growth through PAH addition	N/A
NPAH	Nucleating PAH species	N/A
CPAH	Adsorbing PAH species	N/A
NNEG	Flag instructing DASSL to constrain all components of solution vector to be non-negative	N/A
END	Required keyword specifying end of input file	N/A

The file in the subdirectory source that requires modification is the sectional model header file. This file provides the aerosol dynamics model with the necessary parameters for the gas phase and particle formation. The user input to this file consists of modifying the values for adjustable variables regarding the specific particle and gas properties. The following shows

a sample header file for the experimental validation in section 5.2.

secstrt.h

```
        Double precision particle constants
    PARAMETER(
                              .
                              .
                              .
1.        FS = 1.35D0,
2.        DFRCT = 1.8D0,
3.        AKF = 1.37D0,
4.        C_aggl = 3.0D0,
5.        c_min_mono = 90000D0,
6.        FVOL = 1.43D0,
7.        DensityP = 1.9D0,
8.        CHI = 2.3D+15,
9.        NumCIncep = 48.D0,
10.       MSECTION = 50,
11.       NUMNUC = 3,
12.       NUMDIM=NumNuc*(NumNuc+1)/2)
                             )
```

This example demonstrates a header file containing the important parameters for the sectional model in the experimental validation. The key parameters for the accuracy of the sectional model include the number of sections `MSECTION`, the spacing factor `FS` used in Eq. 7, the number of carbon atoms considered in the first mass section `NumCIncep`, the particle density `DensityP` in $\frac{g}{cm^3}$ used in Eq. 39, `NUMNUC` which specifies the number of nucleating and adsorbing species, and the number of dimers in the simulation `NUMDIM`. The number of sections and the spacing factor must be chosen carefully to incorporate the full range of particle sizes while keeping the computational efforts at a minimum. This study uses 50 sections for the experimental validation as it completes the simulation with low run times and good accuracy.

S.1.4. Running the Code

After defining the input files and simulation parameters, it is necessary to compile the code to create an executable. The package uses a Makefile with rules to create the NanoPFR executable and to clean the directory afterwards. The program uses the GFortran Compiler by default, although the Intel Fortran Compiler may be used by specifying the correct

compiler in the include directive. The following steps and commands will compile and run the program:

1. Compile the code using the command `make npfr.exe` while in the source directory. This process creates the executable `npfr.exe` in the top-level directory.
2. Run the code at the top-level directory while specifying the input and output files using the command `./npfr.exe < npfr.inp > npfr.out`.

S.1.5. Post Processing the Data

After finishing the simulation, The NanoPFR code produces 2 output data files as plain text with a specific format. The first output file is named `npfr.dat`, which stores axial information of the reactor. This file contains the temperature, residence time, mean particle characteristics, and PAH concentrations at each spatial step in the simulation. Table S2 summarizes the keywords and the units of the output. The next file is `npfr.out`, which stores information on the gas-phase species and detailed particle characteristics. This file is optional and will be displayed on the terminal if not specified in the run command. It provides the residence time, temperature, pressure, density, gas-phase mole fractions, agglomerates, and primary particles at each axial location from the reactor inlet. Additionally, the file presents the PSD profile and the representative mobility diameters for the agglomerates at the end of the reactor. Table S3 summarizes the keywords and the units of the output. Visualizing the results can be achieved by importing the desired data into a post-processing software such as OriginLab or Excel. Plotting and interpreting the ending PSD profile is as simple as copying and pasting the data into Excel, and generating the figure.

TABLE S2. Output file description for `npfr.dat`

Variable Name	Description	Units
X	Axial Distance from `XSTR`	cm
T	Temperature	K
Tau	Residence time	sec
A	A followed by a number specifies the PAH species	Mass Fraction (g/g)
Fv	Carbon Nanoparticle Volume Fraction	ppm
#/cm3	Number of agglomerates	#/cc
#/g	Number of agglomerates	#/g
Kn	Knudsen Number	dimensionless (l/l)
Mean_dm	The mean mobility diameter	nm
Mean_dp	The mean primary particle diameter	nm

TABLE S3. Output file description for `npfr.out`

Variable Name	Description	Units
X	Axial Distance from `XSTR`	cm
RESIDENCE TIME	Residence time	sec
TEMPERATURE	Temperature	K
GAS DENSITY	Gas density at axial location	g/cc
PRESSURE	Pressure at axial location	TORR
VELOCITY	Flow velocity at axial location	cm/s
GAS-PHASE MOLE FRACTIONS	In this section, it will show all the mole fractions of the involved gas phase species. Species considered are obtained from mechanism files.	Mole Fractions (mol/mol)
AG #	This stores the number of agglomerate particles in the # mass bin. For instance AG 1 stores the number of agglomerates in the first mass section	#/g
PP #	This stores the number of primary particles in the # mass bin. For instance PP 1 stores the number of primary particles in the first mass section	#/g
DIM #	This stores the number of PAH particles of Dimer with an index of # from the mechanism file	#/g
Dv	This stores the representative collision diameter for each mass bin. (only shown at the end PSD)	nm
Dm	This stores the representative mobility diameter for each mass bin. (only shown at the end PSD)	nm
DN/DLogDp	This stores the number density of agglomerates in each mass bin. (only shown at the end PSD).	1/cc

S.2. Grid Independence Analysis for Mean Primary Particle Diameter

Fig. S2 demonstrates the effect of grid spacings on the results of the surface growth validation using the arithmetic mean primary particle diameter. Similar to the analysis involving the volume fraction, all validation runs produce similar results, only differing slightly in the initial slope of the surface growth from 0 ms to 20 ms. As the grid gets finer (i.e. $d\tau = 6.0E-1$ ms and smaller), the results appear identical, and the solution is now independent of the grid.

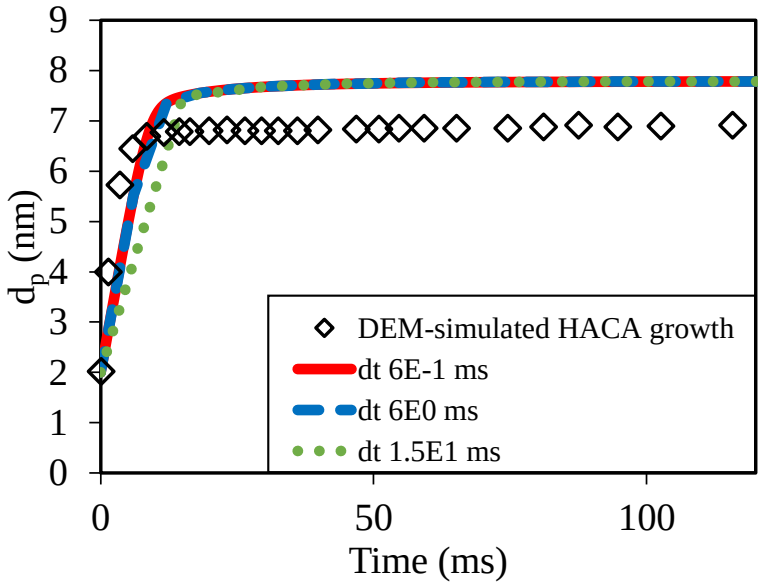


FIGURE S2. Grid independence of the NanoPFR program using the surface growth (arithmetic mean diameter) test case under the following initial conditions: $N_{a,0} = 4.5E12\ \frac{\#}{cc}$, $d_{P,0} = 2\ nm$, $F_V = 0.1885\ ppb$ and $T = 1830\ K$.

S.3. Detailed Analysis of Primary Particle Geometric Standard Deviation

This section examines the evolution of the primary particle characteristics during the test case given in section 4.1.3. Fig. S3a shows the trend from the test case that compares the geometric standard deviation of primary particles from the NanoPFR code against the DEM results. The code output demonstrates an increasing $\sigma_{g,p}$ until surface growth ends, then follows a continuous decrease while agglomeration occurs. This result contradicts the DEM results that plateau after an initial increase. Fig. S3b shows the total surface area over time. The total surface area reaches an asymptotic value after 20 ms, which is where surface growth finishes, and the volume fraction becomes constant. This demonstrates that the code can correctly conserve surface area during the surface growth and agglomeration processes.

Fig. S3c depicts the calculated PPSD in various residence times. This figure demonstrates the artificial narrowing of the PPSD. As surface growth ends and agglomeration becomes the main process, the primary particles begin to converge to an average size. At 5 ms, the PPSD profile is smooth and covers a wide range of primary particle sizes. At 20 and 120 ms, the PPSD exhibits a sharp profile that covers a small range of primary particle sizes. This issue arises from the limitation of the implemented sectional model. The modeling of the agglomeration process occurs on a mass basis. In this model, the collisions of agglomerates produce a larger resulting mass, which is transferred into a higher section. The

code calculates the average primary particle size in each section using the total agglomerate mass and n_p, which can be seen in Fig. S3d. Consequently, this assumption prevents the tracking of the history of primary particle sizes.

The agglomeration process in this test case highlights the drawback of this method. Each mass section in the PSD can contain agglomerates of different structures (they can consist of many small primary particles or a few large primary particles). However, the code only stores the average primary particle size in each section, which prevents the tracking of the history of different primary particle sizes. The code calculates the average based on the number of primary particles of a specific size in each agglomerate. Consequently, the large primary particles with a lower number ($d_p > 10$ nm) will eventually get lost from averaging during agglomeration. For instance, a section may contain a large number of agglomerates consisting of many small primary particles, but only a few consisting of large primary particles. When the code determines the average primary particle size for the section, it will heavily favor the many small primary particles and disregard the larger sizes. As agglomeration continues, this averaging process occurs for all the collisions and ultimately leads to a significant narrowing of the primary particle sizes. This issue can be resolved through the use of a fully 2D sectional model, which will track the PPSD in each section rather than relying on the average value of all the contributing primary particles. However, these models can be quite computationally expensive.

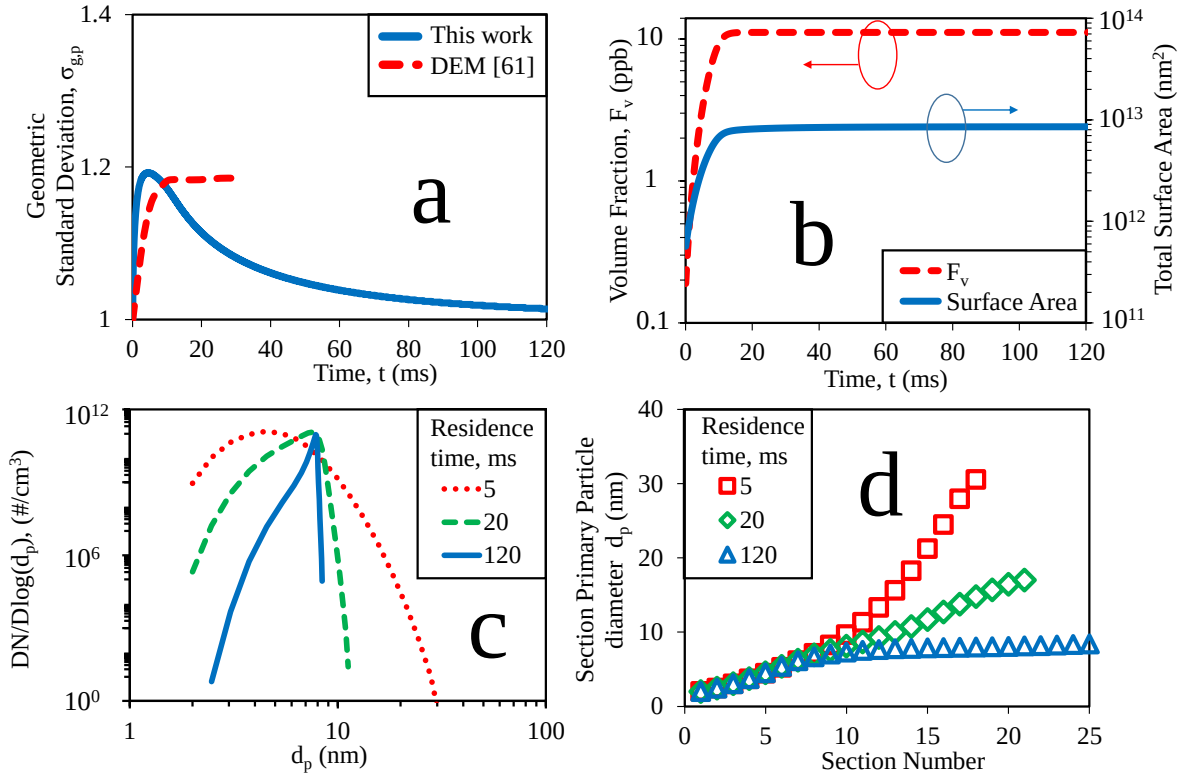
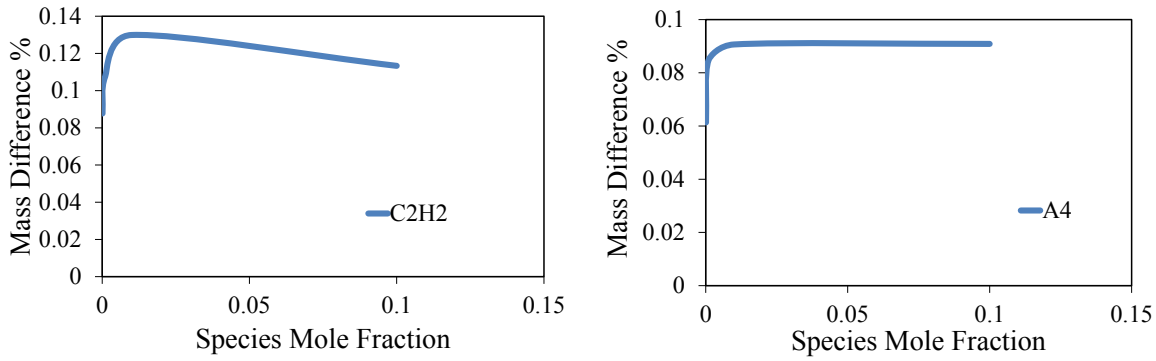


FIGURE S3. The evolution of primary particle geometric standard deviation $\sigma_{g,p}$ (a), total surface area (b), primary particle size distribution (c), and the section primary particle sizes (d) as a function of time. The simulations for both the DEM and the model in this work used the following initial conditions: $N_{a,0} = 2.6 \times 10^{12}\ \frac{\#}{cm^3}$, $d_{p,0} = 2\ nm$, and $T = 1830\ K$ [61].

S.4. Detailed Analysis of Numerical Mass Diffusion

Fig. S4 presents the investigation of numerical diffusion within the NanoPFR code. The analysis examines four trials for both A_4 and C_2H_2, under the concentrations of 0.0001, 0.001, 0.01, and 0.1 each. The inlet initial conditions include $F_{V,intial} = 1.1 \times 10^{-2}\ ppm$, $N_{a,initial} = 1.63 \times 10^{16}\ \frac{\#}{g}$, particle density of 1.9 $\frac{g}{cc}$ and a reactor temperature of $T = 1830\ K$. The parameters for the sectional model consists of 25 mass sections and a spacing factor of 1.9 for consistency with the previous sections. Manual mass balance calculations between the gaseous species and the initial particle mass fraction determine the theoretical results. The simulation is performed without agglomeration, and with surface growth operating via PAH addition and HACA. Fig. S4 compares the final mass fraction of the theoretical results against the final mass fraction output from the model. The lower the mass difference represents a lower overall numerical diffusion in the model. In the figure, C_2H_2 depicts the error from surface growth via HACA. As the concentration of C_2H_2 increases, the error increases due to a higher amount of mass converting from the gas phase to the particle. A_4 shows the error from surface growth via PAH addition. Similar to HACA, a higher

concentration of reacting species yields a larger error. However, the error remains unchanged after a certain concentration, even after increasing it an order of magnitude. Both cases yield an overall mass difference of less than 1 percent, which is an acceptable amount of numerical diffusion.



(A) Numerical Mass Diffusion via HACA

(B) Numerical Mass Diffusion via PAH addition

FIGURE S4. Investigation of numerical diffusion within the NanoPFR code for the following inlet concentrations: 0.0001, 0.001, 0.01, 0.1 for both A_4 and C_2H_2 respectively. The initial conditions for both cases are:$F_{V,initial} = 1.1 \times 10^{-2}$ ppm, $N_{a,initial} = 1.63 \times 10^{16}$ $\frac{\#}{g}$, and $T = 1830$ K.

S.5. Analysis for the Particle Density on Agglomeration model

Fig. S5 depicts the effect of particle density on the agglomeration model through N_a (a) and n_p (b). The analysis examines the simulation results for carbon nanoparticles with particle density varying from 1.7 g/cc to 2.1 g/cc. The results show a moderate difference between the simulations across the range. Frame a) shows that the N_a at 1.7 g/cc is 14 % lower than N_a at 2.1 g/cc. Moreover, Frame b) shows that the n_p at 1.7 g/cc is approximately 30 % greater than the n_p at 2.1 g/cc. Lower particle densities lead to a higher n_p and lower N_a.This is expected as lower particle densities result in a higher collision frequency and thus fewer overall agglomerates. Consequently, more collisions lead to larger agglomerates with more primary particles per agglomerate.

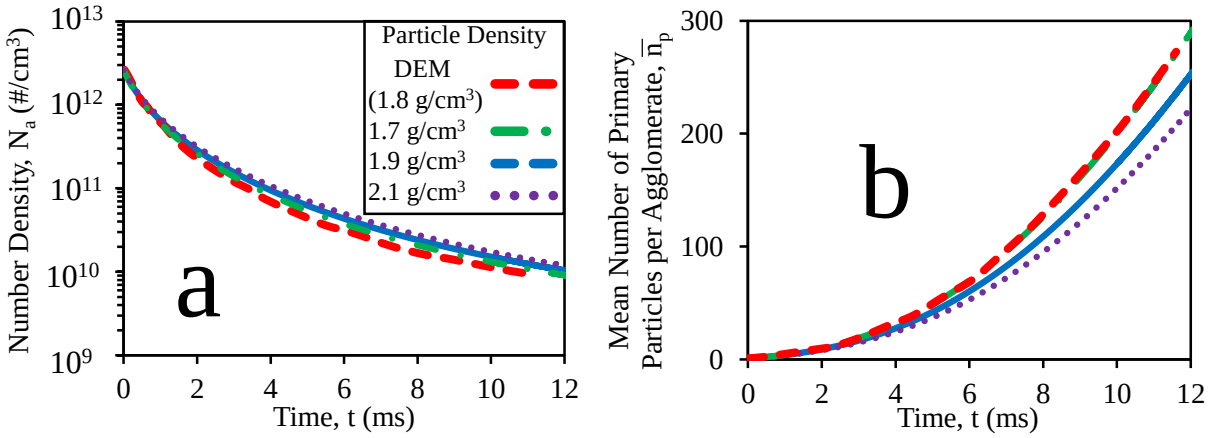


FIGURE S5. The effect of particle density on agglomerate number density N_a (a) and number of primary particles per agglomerate n_p (b) under the following initial conditions: $N_{a,0} = 2.6 \times 10^{12}\ \frac{\#}{cm^3}$, $d_{P,0} = 2\ nm$, and $T = 1830\ K$.

S.6. Additional Experimental Comparison with Ethylene Pyrolysis

In this section, the particle characteristics of the model are compared against experimental data obtained at the reactor outlet by Mei et al. [8]. The experiment involves the pyrolysis of ethylene flowing at 8 L/min in a laminar flow reactor at a constant high-temperature region and atmospheric pressure. The laminar flow reactor has a length of 1400 mm and a diameter of 1.6 cm, with a constant high-temperature region of 1473 K, 1573 K, and 1673 K over three experiments. Fig. S6 depicts the predicted evolution of the mean primary particle diameter and mean mobility diameter in comparison to the experimental results obtained at the reactor exit. The model overpredicts $\overline{d_p}$ and underpredicts $\overline{d_m}$ with maximum error of 43 % and 67 %, respectively.

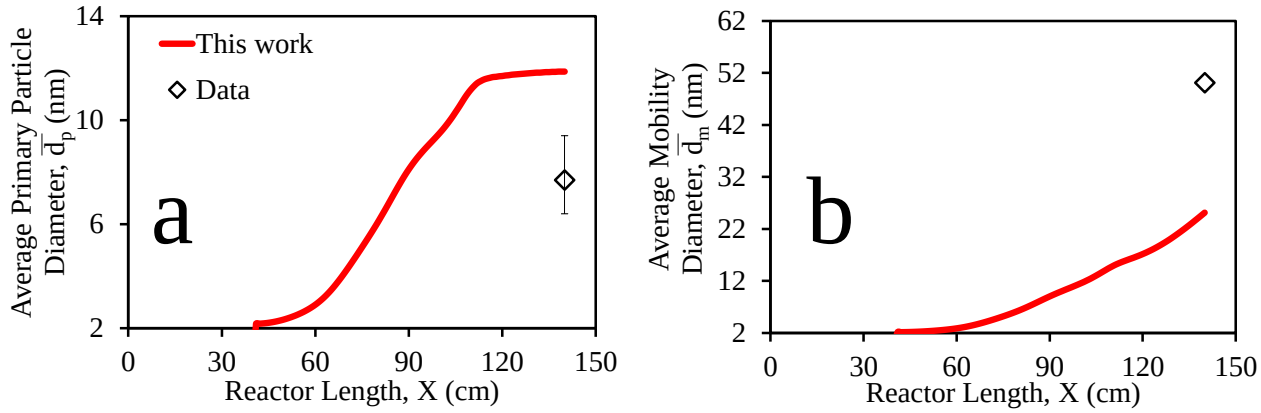


FIGURE S6. Comparison of the mean mobility $\overline{d_m}$ (a) and mean primary particle $\overline{d_p}$ (b) between the model in this work (solid line) and experimental measurements (diamonds) by Mei et al. [8]. The operating conditions include: an inlet flow rate of $8.0 L/min$, a mixture composition of 0.6% C_2H_4 and 99.4% N_2, and a nominal peak temperature of 1673 K.

Fig. S7 compares the PSD profile results from the model against the experimental measurements at the reactor outlet at a maximum reactor temperature of 1473 K, 1573 K, and 1673 K. Frame a) shows that at low temperatures the model overpredicts the number of small agglomerates and fails to capture larger agglomerates. Frame b) and c) demonstrates that the model estimates significantly more larger-sized agglomerates over the experiments at higher temperatures. Overall, while the trend is similar between the model and experiments, further improvements in the model parameters are required.

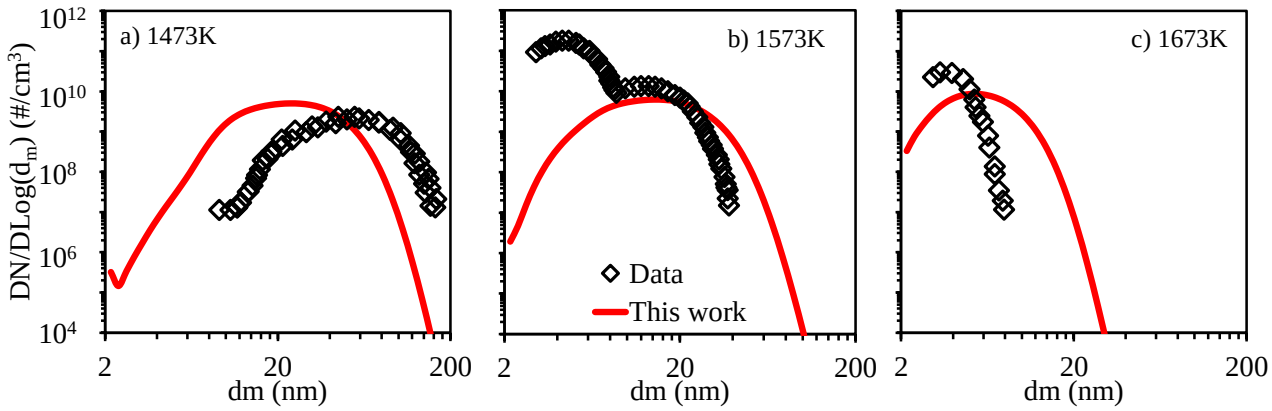


FIGURE S7. Comparison of the simulated PSD profile results from the NanoPFR model (lines) with experimental results (diamonds) [8] at a nominal peak temperature of 1473 K (a), 1573 K (b), and 1673 K (c). The operating conditions include: an inlet flow rate of $8.0 L/min$, a mixture composition of 0.6% C_2H_4 and 99.4% N_2.