

PyPE: A new platform for analyzing trade policy with partial equilibrium models

Saad Ahmad, Tyler Daun and Christopher Montgomery

ECONOMICS WORKING PAPER SERIES
Working Paper 2022–05–C

U.S. INTERNATIONAL TRADE COMMISSION
500 E Street SW
Washington, DC 20436

May 2022

Office of Economics working papers are the result of ongoing professional research of USITC Staff and are solely meant to represent the opinions and professional research of individual authors. These papers are not meant to represent in any way the views of the U.S. International Trade Commission or any of its individual Commissioners. Working papers are circulated to promote the active exchange of ideas between USITC Staff and recognized experts outside the USITC and to promote professional development of Office Staff by encouraging outside professional critique of staff research. Please address correspondence to saad.ahmad@usitc.gov or christopher.montgomery@usitc.gov.

PyPE: A new platform for analyzing trade policy with partial equilibrium models
Saad Ahmad, Tyler Daun and Christopher Montgomery
Office of Economics Working Paper 2022–05–C
May 2022

Abstract

We introduce a new Python package PyPE for developing and conducting partial equilibrium (PE) analysis of trade policies. PyPE offers practitioners a flexible framework to select and combine different trade model features to determine the impact of select policies on a particular market or industry. With a user-friendly interface, PyPE allows users to run model simulations and report results with only a few lines of code. Interested users are also encouraged to assist with PyPE development by proposing or constructing new model features that can be incorporated within the package.

Saad Ahmad, Office of Economics
saad.ahmad@usitc.gov

Tyler Daun, Office of Economics
tyler.daun@usitc.gov

Christopher Montgomery, Office of Economics
christopher.montgomery@usitc.gov

1 Introduction

Partial equilibrium (PE) models are often used to analyze the economic effects from changes in trade policy. In contrast to general equilibrium economy-wide models, PE models can be narrowly tailored to examine the impact of the policy changes on specific industries and sectors. With their practical data requirements, PE models represent a convenient option for conducting ex-ante quantification of both tariff and non-tariff policy changes. Some recent policy applications of PE models include the effect of stricter USMCA Rules of Origin on the auto sector (USITC, 2019); the impact of CBERA preferences on U.S. domestic industries (USITC, 2021a); and the impact on the U.S. domestic raspberries processing industry from higher imports of raspberries (USITC, 2021b).¹

In this paper, we introduce a new Python package PyPE that allows for the development and execution of PE models analyzing the impact of trade policy. PyPE offers a flexible framework for model developers to create new types of PE models. Our framework allows model developers to select and combine pre-codified model features that are stored and validated within the PyPE package. Likewise, interested users are encouraged to assist with PyPE development by proposing or constructing new model features to be stored within the package. Existing features and future updates will be designed to be agnostic to the number of firms in a market, allowing completed models to execute given any number of user-specified firms or products.

In addition to tools for model developers, PyPE offers an intuitive, user-friendly interface for running model simulations and reporting results. With PyPE, model users can supply their own market data to a flexible model of their choosing in less than a few simple lines of Python code. PyPE also reports key model results in an intuitive, easy to access manner.

Rest of the paper is organized as follows. Section 2 covers the theoretical foundation of

¹Reports published by the U.S. International Trade Commission are available here: https://www.usitc.gov/commission_publications_library.

the PyPE models, including the key equations for each of the included models of competition. Section 3 describes the structure of the PyPE package. Section 4 shows a demonstration of PyPE using sample inputs and compares results of the different types of competition. Section 5 concludes with a brief discussion of future plans for the PyPE package.

2 Theory

2.1 Demand

The suite of PE models currently available in PyPE is built on the assumption that consumers have a love of variety and prefer to consume a little of each product variety in the market. Product varieties in the market can be differentiated either at the country level, as in Armington (1969), or at the firm level following Krugman (1979) and other more recent trade models. We further assume a Constant Elasticity of Substitution (CES) demand framework so that consumers switch between product varieties at a constant rate of σ , the elasticity of substitution.²

Let i represent a distinct source supplying goods to the domestic market. To simplify the subsequent discussions, we assume there are only 3 sources of goods in the market: domestic suppliers that face no tariffs (d), foreign suppliers whose products are subject to the new tariffs (s), and foreign suppliers whose products are not subject to the new tariffs (n).³ Under CES demand, domestic consumers thus face the following utility maximization problem:

$$U = \left(\sum_i m_i b_i q_i^{\frac{\sigma-1}{\sigma}} \right)^{\frac{\sigma}{\sigma-1}} \quad s.t. \quad \sum_i m_i p_i (1 + \tau_i) q_i = E \quad (1)$$

²This framework can be easily extended to a nested CES demand structure so that consumers in the first stage aggregate over a user-defined subset of varieties in the industry and a composite of the varieties excluded from the subset. In the second stage, consumers aggregate over the subset of varieties that comprise the composite in the first stage.

³The number of sources in PyPE is flexible and user can input as many sources as needed.

For each source $i \in (d, s, n)$, p_i is the producer's price and q_i is the quantity demanded of the good in the domestic market. Consumers pay an ad valorem tariff rate τ_i for all goods coming from source i with $\tau_d = 0$. b_i is an exogenous parameter capturing asymmetries in consumer preferences across sources while m_i is the mass of firms producing the different varieties of the good in each source country. Note that under an Armington framework of trade, we just assume that countries rather than firms are supplying the different product varieties. Lastly, E is the level of aggregate expenditure in the domestic economy for this good.

If we let $\beta_i = b_i^\sigma$, then maximizing utility subject to the budget constraint gives us the following consumer demand from each source country:

$$q_i = \beta_i E p_i^{-\sigma} (1 + \tau_i)^{-\sigma} P^{\sigma-1} \quad (2)$$

$$P = \left(\sum_i m_i \beta_i [p_i (1 + \tau_i)]^{1-\sigma} \right)^{\frac{1}{1-\sigma}} \quad (3)$$

Note P is the CES price index for all the varieties consumed in home country. In order to capture aggregate demand effects in PyPE, we take $\theta < 0$ to be the price elasticity of total demand and k an aggregate demand parameter such that:

$$E = k P^{\theta+1} \quad (4)$$

$$q_i = k \beta_i p_i^{-\sigma} (1 + \tau_i)^{-\sigma} P^{\sigma+\theta} \quad (5)$$

$$V_i = m_i k \beta_i p_i^{1-\sigma} (1 + \tau_i)^{1-\sigma} P^{\sigma+\theta} \quad (6)$$

Here V_i represents the total home sales from source i . We next set initial consumer prices from all source countries to 1 and normalize the number of firms and preferences by their domestic counterparts such that $m_d \beta_d = 1$. Using the initial sales for each source in the

domestic market, we get the following calibration:

$$m_s\beta_s = \frac{V_s}{V_d} \quad (7)$$

$$m_r\beta_r = \frac{V_r}{V_d} \quad (8)$$

$$k = \frac{V_d}{[1 + m_s\beta_s + m_r\beta_r]^{\frac{\theta+\sigma}{1-\sigma}}} \quad (9)$$

Thus, initial revenues in the domestic market along with the elasticity parameters are all that is required to recover the other structural parameters of the CES demand.⁴

2.2 Supply under Monopolistic Competition

To model supply under monopolistic competition, we assume that in each source country $i \in (d, s, n)$, there is a mass m_i of homogeneous firms each producing a unique variety. To simplify the modeling, we do not allow for firm entry/exit so that the number of firms is held fixed in the PE simulations, restricting the analysis to only the short-run effects on market participants from policy changes.

Following Krugman (1979), all firms in source i face constant marginal costs c_i to produce their variety. A continuum of monopolistic competitive firms along with the CES demand framework implies that each firm's optimal price will be given as a constant markup over their marginal costs:

$$p_i = \frac{\sigma}{\sigma - 1} c_i \quad (10)$$

With constant markups, there will be a 100% pass-thru of tariffs with domestic consumers paying $(1 + \tau_i)p_i$ for all varieties produced in country i . A 100% pass-thru means that solving for the new equilibrium from the demand side is sufficient for determining the effects of a

⁴The above calibration allows us to recover only $m_i\beta_i$ for each source i with the number of firms m_i indeterminate. Thus, more information would be needed if we want to use the underlying PE framework to also capture firm entry or exit from the policy shock.

tariff change. In practice, this means solving the system of equations given by demand from each source i along with the CES price index as tariffs go from τ_i^0 to τ_i^1 .

Together with consumer prices and quantities, we can also use the monopolistic competition framework to determine changes in firm profits. Besides variable input costs, we assume that firms incur a fixed cost f_i to sell to the domestic market. Thus a firm's net profits from selling to the home market are computed as:

$$\pi_i = p_i q_i - c_i q_i - f_i$$

It can be shown that a firm's operating profit, revenue minus variable costs, are a fixed proportion of its revenue.⁵

2.3 Supply under Perfect Competition

Supply under perfect competition is characterized by many firms (specifically, an infinite number of infinitely small firms) all selling exactly identical products. Individual firms have no pricing power and take the market price as given when choosing how much to produce. Given these characteristics, it is often convenient to return to an Armington framework of trade so that goods are differentiated by source country, rather than firms.

A number of functional forms are available in PyPE for modeling supply under perfect competition. The canonical treatment in PE analysis is to assume isoelastic supply curves so that the price elasticity of supply ϵ_i is the same at all points on the curve:

$$q_i = \alpha_i p_i^{\epsilon_i} \tag{11}$$

Here, α_i is a calibrated supply parameter that captures cost factors such as wages, inputs

⁵See Ahmad (2019) for additional details on how to compute changes in profits in PE models with monopolistic competition.

and energy costs which remain fixed in the policy simulations. We note that the standard supply curves in PyPE can be easily modified to represent perfectly inelastic supply if the elasticity is 0 or perfectly elastic supply if the elasticity is infinity.

Market equilibrium is then just the set of prices for which the demand for each product from source i is equal to its supply. Focusing on the case of isoelastic supply curves, we obtain a system of non-linear equations, where each source $i \in (d, s, n)$ has the following market-clearing condition:

$$k\beta_i p_i^{-\sigma} (1 + \tau_i)^{-\sigma} P^{\sigma+\theta} = \alpha_i p_i^{\epsilon_i} \quad (12)$$

Parameters k , α_i , β_i can be calibrated using the respective market shares of each source i in the domestic market. These calibrated parameters when combined with information on the respective demand, supply and substitution elasticities are sufficient to determine the effect on prices and subsequently quantities from a change in tariffs.

One practical option often observed in policy applications is to log-linearize the above system of equations, enabling a reduced form solution to be obtained for the respective price changes due to a change in tariffs. However, as noted by Francois and Hall (1997), linearization can often include substantial approximation error, particularly in the case of large policy shocks. Thus, we prefer to utilize a non-linear solver in PyPE when solving the above model for perfect competition.

2.4 Supply under Bertrand Competition

For supply under Bertrand competition, our sources $i \in (d, s, n)$ now represent individual firms rather than countries. Since there is no longer any need to track the number of firms per country under this framework, we drop m_i from the subsequent demand equations as well. Again, the focus is only on the short run, so there is no entry or exit of firms from the domestic market.

Similar to monopolistic competition, each firm incurs a fixed cost of production f_i along with constant marginal costs c_i . The profit for each firm i is then given by:

$$\pi_i = (p_i - c_i)q_i - f_i$$

Firms compete over prices with each firm i choosing the price that maximizes its profits, while taking the prices of other competing firms as given. As the number of firms in the market is small, each firm's own price can also impact the aggregate price index and thus overall demand. The Nash equilibrium under these market conditions is for each firm to set their price using the following first order condition:

$$p_i = (p_i - c_i) \left(\sigma - (\sigma + \theta) \frac{b_i [p_i (1 + \tau_i)]^{(1-\sigma)}}{\sum_k b_k [p_k (1 + \tau_k)]^{(1-\sigma)}} \right)$$

Since information on firm's marginal costs are often unavailable, we follow Riker (2018) and use data on initial market shares h_i and the elasticity parameters to calibrate the ratio of a firm's marginal costs to its prices $\frac{c_i}{p_i}$:

$$\frac{c_i}{p_i} = \left(1 - \frac{1}{\sigma - (\sigma + \theta)h_i} \right)$$

To solve the model for a given change in tariffs, we combine the first order conditions of each firm with their respective demand equations and the CES price index. As was the case for the other supply frameworks, a non-linear solver is used by PyPE to solve this system of equations to determine the effect on prices and quantities for each firm due to a change in tariffs.

3 PyPE Implementation

The PyPE package features a structure with three distinct layers. The innermost layer features the “building block functions” that codify the basic features common across all the PE models available in PyPE. The middle layer is PyPE’s “CESModel” class, which combines the building block functions with user-supplied data in order to calibrate key parameters and organize model inputs. In addition to calibrating CES demand parameters from user-supplied inputs, the CESModel object contains functions that ensures model inputs are organized consistently, and are ready to be used by PyPE’s user-facing, simulation-ready models. PyPE currently houses 3 different competition-specific, simulation-ready models. Users can choose between perfect, monopolistic, and Bertrand competition in the domestic market being analyzed.⁶ The different features of these models are described in section 2. The remainder of this section provides additional details on the layered structure of PyPE and can be useful for advanced users interested in contributing to PyPE functionality.

3.1 Building block functions

At its core, any PE model is simply a system of equations, describing the demand and supply features of a particular market or industry. By focusing on certain aspects of demand and/or supply, PE modellers aim to capture the market dynamics that are most relevant for their selected market or industry and policy experiment. PyPE facilitates this task by codifying and storing functions, such as the CES demand function, that are commonly featured in standard PE models. By having access to a set of functions that have already been validated and tested, PyPE enables practitioners to focus on adding new features to the existing PE models, rather than spend model development constructing and redefining the same set of functions that are commonly used across many different types of PE models.

⁶Ongoing development of the PyPE package will include the creation of new competition-specific frameworks.

In its current iteration, the PyPE submodule `CES_demand` houses the individual functions needed to solve for consumer demand in a CES framework. Given appropriate user-supplied data inputs, the `CES_demand` functions calibrates the consumer preference parameters, the CES industry price index, and the industry-wide demand parameter. These calibrated parameters serve as key PE model inputs for calculating the effect on prices and quantity from a change in tariffs.

PyPE users have direct access to these building block functions. For instance, the CES price index given in equation (3) can be accessed by a PyPE user by calling the function `PriceIndex` from the `CES_demand` module as shown below. Here `PriceIndex` function takes a single value for the Armington elasticity of substitution and arrays of length i for producer prices, tariff rates, and consumer preference weights.⁷ The function returns a single value representing the CES industry price index.

With access to the `PriceIndex` function, users can quickly calculate how the industry price changes in response to an increased tariff on a producer. With a few lines of code as shown below, a PyPE user can determine that the CES price index will increase by 3.8% following the addition of a 25% tariff on the second producer. While this example assumes perfect pass-thru of tariffs to consumers, other PyPE functions also allow for producer prices to change depending on the sort of competition employed in the market of interest as discussed in more detail in section 2.

⁷In this example, we supply the CES preference weights (`demand_shifters`) exogenously. In practice, preference weights will be calculated from user-supplied price and consumer expenditure data using the `CESWeights` function housed in the `CES_demand` module.

```

1 #Import the CES_demand module
2 from pype.central_functions import CES_demand
3
4 #User-provided inputs
5 armington = 3.0 #Industry elasticity of substitution
6 demand_shifters = [.7, .2, .1] #Preference weights by source
7 p0 = [1,1,1] #List of prices, in this case normalized to 1
8 tf0 = [0, 0, 0] #Initial tariff rates
9 tf1 = [0, .25, 0] #Final tariff rates with 25% tariff on producer 2
10
11 #Compute CES Price Index
12 initial_Pindex = CES_demand.PriceIndex(elasticity = armington, weights=
    demand_shifters, prices = p0, tariffs=tf0)
13
14 final_Pindex = CES_demand.PriceIndex(elasticity = armington, weights =
    demand_shifters, prices = p0, tariffs = tf1)
15
16 print((final_Pindex/initial_Pindex)-1) #Change in CES price index of 3.8%

```

A key feature of PyPE's building block functions is that they are formulated to be agnostic to the number of sources/producers within a market. As such, PyPE functions and competition-specific models can be executed to solve the PE models with any number of user-specified firms/sellers participating in the domestic market. In doing so, users have significantly more flexibility in their policy applications than existing tools that often require the number of firms/sellers to be pre-defined and hard coded.

The PyPE building block functions represent the most basic modeling features and tools for organizing data inputs and outputs. End users that are interested in using PyPE for running model simulations from completed models will rarely interact with building block functions directly but should know that those building block functions are being used in

the background by the other levels of the PyPE framework. These functions can also be helpful for developing new model features and tools for performing PE analysis, and future development of PyPE will focus on expanding the universe of building block functions to help model new forms of demand, competition, and interactions between different markets and industries.

3.2 Organizing inputs and calibration using a CESModel object

As an object-oriented programming language, Python allows data and functions which act upon that data to be grouped together and accessed as objects of a specific class. PyPE utilizes this feature of Python and organizes the building block functions with user-supplied input data as a specialized class object called CESModel. The CESModel object ensures that input data and simulation results are consistently structured, regardless of the specific PE model or equilibrium conditions being used. All user-supplied and calibrated parameters are stored in the same location, and can be accessed using the same methods regardless of which competition-specific PE model is being used. Doing so helps ensure a consistent logic for developing new PE models and adding new features to existing models. At the same time, the CESModel object ensures users can easily run simulations using the same input data with different types of competition models and thus test the sensitivity of their results to different assumptions on market structure.

The competition-specific PE models housed in the PyPE package requires user to provide certain data inputs and parameter values. Due to the CES demand structure, a number of these inputs are the same irrespective of the competition-specific PE model being used for simulations.⁸

To successfully initialize the CESModel object in PyPE, users need to provide the fol-

⁸Future iterations of PyPE feature additional objects for other, non-CES demand structures such as linear demand. New demand structures will likely require different data requirements and take on different calibration steps.

lowing data inputs at minimum:

1. Value of domestic shipments and imports from each source (country, firm, plant, etc.).⁹
2. Initial and new tariff rates for the different sources supplying the domestic market
3. Information on the relevant elasticities in the market. All PE models in PyPE require users to provide the elasticity of substitution. Users also have the option to provide the industry demand elasticity, which is otherwise assumed to be -1 to match a Cobb-Douglas structure of demand across different product groups. Lastly, for perfect competition PE models, users also need to enter the elasticity of supply by source.

Once initialized, the CESModel object stores the substitution and industry demand elasticities as attributes. The user-supplied matrix of market shares of each source of supply is also added to the CESModel object. Using PyPE’s building block CES functions and user-supplied inputs, the CESModel calibrates and stores an initial industry price index, P , and CES preference parameters, b_i . CESModel objects also store the functions for solving for CES demand which are used to solve for new equilibrium conditions during PE model simulations. In practice, most users will rarely interact with CESModel objects directly and will instead run model simulations using PyPE’s user-facing functions that draw heavily from the CESModel class. However, PyPE users interested in developing new CES models can benefit from having access to the CESModel class as a starting point.

3.3 Off-the-shelf competition-specific models for PE simulations

By combining user-provided data with built-in CES functions, the CESModel object has almost everything needed to solve a PE model. The last remaining requirement is for the

⁹Users can optionally supply information on firm/product prices, although all prices are normalized to 1 as a default behavior. In our experience, doing so improves the ability of Scipy’s Fsolve function to solve for prices in the new equilibrium, while still allowing for results to be reported in terms of percentage changes.

user to select how sources of supply compete with one another when supplying the domestic market. In its current version, PyPE offers users three finalized, competition-specific models: perfect competition, monopolistic competition, and Bertrand competition. These models combine user-supplied inputs, calibrated CES parameters, and the competition-specific equilibrium condition to execute PE simulations. Results from the simulations are organized and stored in the model object and output to an excel spreadsheet format. The type of market competition chosen by the user determines the market-clearing equilibrium conditions for conducting the policy simulation in PyPE. Thus, users should ensure that the selected market competition matches the key characteristics of the industry being analyzed.

PyPE users interested in running PE simulations, rather than developing new models can use these competition-specific models to easily execute a simulation with little coding or programming required. Model users supply data inputs while the off-the-shelf PyPE model performs the actual task of calibrating demand parameters, solving for new equilibrium conditions, and reporting simulation results. Section 4 provides some simple examples of how users can execute a PE simulation using PyPE's different competition-specific models.

3.3.1 Additional User Features for Modeling Perfect Competition

PyPE users analyzing markets operating under perfect competition have the option to also select the functional form of the supply curve in order to better capture certain seller characteristics. Currently, the options are isoelastic, perfectly elastic, perfectly elastic, and asymptotic. These can be chosen separately for each producer (country). Each of these forms requires one parameter chosen by the user and one calibrated parameter (for some forms, the user chosen parameter is unused).

The 'standard' supply form is isoelastic, so the price elasticity of supply is the same at all points on the curve. The elasticity ϵ_i is supplied by the user while the scalar α_i is calibrated from the data.

The perfectly elastic and perfectly inelastic supply forms are extreme variants of the ‘standard’ supply form. Both of these functions only use a calibrated parameter α_i , but that parameter is interpreted differently. With perfectly inelastic supply, quantities do not respond at all to prices. For this form, the calibrated parameter is the quantity produced for all prices. This is the same result as using the ‘standard’ isoelastic form and setting the constant elasticity to zero. With perfectly elastic supply, any quantity can be produced as long as price is greater than or equal to the constant marginal cost. For this form, the calibrated parameter is the constant marginal cost. While any quantity supplied is possible when price is greater than or equal to the marginal cost of this form, the equation is defined in equilibrium so that the quantity supplied in that case is equal to quantity demanded.

4 PyPE demonstration

In this section we demonstrate how to use PyPE. We begin with a simple example featuring three sources of production: domestic, non-subject, and subject. Each producer faces zero tariffs in the initial equilibrium. Following a policy change, the subject source of supply faces a tariff. As such, simulation results represent the effect of the new tariff on market outcomes. In order to demonstrate PyPE flexibility, we run simulations for each available competition framework in PyPE and conclude the section by discussing how choice of supply framework influences model outcomes in this simple example.

All the PyPE simulations follow these simple steps. First, the user enters producer-level data and tariffs using Python’s Pandas package as demonstrated in the below code:

```
1 #User-provided producer-level data inputs
2
3 import pandas as pd #Import the Pandas package
4 initial_prices=[1,1,1]
5 producer_expenditures=[70,10,20]
```



```

6 initial_tariffs=[0,0,0]
7 final_tariffs=[0,0,0.25]
8 producer_data=pd.DataFrame({
9     "producer":["Domestic","Non-Subject","Subject"],
10    "prices0":initial_prices,
11    "expenditures0":producer_expenditures,
12    "tariffs0":initial_tariffs,
13    "tariffs1":final_tariffs
14    })

```

Table 1 summarizes the selected model data used to generate the PE simulation results featured in this section.

Table 1: Selected parameters for PyPE demonstrations

Parameter	Values
Sources/Producers	Domestic, Non-subject, Subject
Source market shares	70, 10, 20
Source initial tariffs	0, 0, 0
Source final tariffs	0, 0, 0.25

The next step is for the user to select the model class from PyPE's competition modules that best fits their market characteristics. Users then initiate their competition-specific model object with the user-provided producer-level dataframe, an industry-wide price elasticity of demand, and the elasticity of substitution. To ensure consistency across models, an elasticity of substitution of 3 and an industry demand elasticity of -1 are used as the elasticity parameters in all these simulations. Upon initialization, PyPE runs the desired simulation, stores inputs and results within the model object, and constructs an excel results spreadsheet at the user-specified output path. The next sections will show how users can execute the three different competition frameworks in PyPE.

4.1 Model simulation: Monopolistic competition

To use the data given in Table 1 with a PE model of monopolistic competition, we can run the following code. The first line imports the requisite Monopolistic model object, while the second line constructs a model object with the user supplying producer data, elasticity values, and an output file path:

```
1 #Import the Monopolistic Competition model from PyPE
2 from pype.monopolistic_competition.moncomp_model import Monopolistic
3
4 moncomp_model=Monopolistic(parameter_matrix = producer_data ,
5                             elasticity = 3.0,
6                             industry_elasticity = -1,
7                             output_path ="DesiredFilePath.xlsx")
```

Table 2 summarizes firm-level outcomes from the monopolistic competition scenario. Firms do not change their prices in response to the tariff, consistent with the constant markup assumption underlying monopolistic competition. Thus, consumers will face a full pass through of the 25 percent increase in tariff, resulting in a 25 percent increase in prices paid for products from the subject producer. All other prices remain the same with the higher tariff leading to changes in the amount of good sold from each producer in the market.

As expected, total expenditures and quantities consumed from the subject producer drop significantly by 31 percent and 44 percent respectively in response to domestic consumers facing higher prices for their product after the full pass through of the 25 percent tariff. Conversely, domestic and non-subject firms benefit from the tariff on subject firms, experiencing a symmetrical increase of 7.75 percent in quantities and expenditures following the imposition of the tariff.

Table 2: Results from simulations of Monopolistic Competition model

Source	expenditures0	expenditures1	quantities0	quantities1	expenditures % change	prices % change	quantities % change
Domestic	70.00	75.43	70.00	75.43	7.76	0.00	7.76
Non-subject	10.00	10.78	10.00	10.78	7.76	0.00	7.76
Subject	20.00	13.79	20.00	11.03	-31.03	0.00	-44.83

4.2 Model simulation: Perfect Competition

In addition to the selected model parameters featured in table 1, the perfect competition model in PyPE requires users to supply a source-specific "form-supply" and "paramB" supply elasticity parameter. For this demonstration, we select the "standard" form for the supply curves and a value of 2.0 for the supply elasticity for each of the three sources of supply. The code to run the simulation under Perfect Competition is given below with the simulation results reported in table 3

```

1 #Import the Perfect Competition model from PyPE
2 from pype.perfect_competition.perfcompetition_model import PerfComp
3
4 producer_data["paramB"]=[2,2,2] #Supply elasticities
5 producer_data['form_supply']='standard' #Supply functional form
6
7 pc_model=PerfComp(parameter_matrix = producer_data,
8                   elasticity = 3.0,
9                   industry_elasticity = -1,
10                  output_path = "DesiredFilePath.xlsx")

```

Table 3: Results from simulations of perfect competition model

Source	expenditures0	expenditures1	quantities0	quantities1	expenditures % change	prices % change	quantities % change
Domestic	70.00	72.37	70.00	71.57	3.38	1.11	2.24
Non-subject	10.00	10.34	10.00	10.22	3.38	1.11	2.24
Subject	20.00	17.30	20.00	15.64	-13.52	-11.56	-21.78

Again, model simulation results show a reduction in market shares and price declines for the source subjected to the new tariff. Since we have assumed an upward-sloping supply curve,

the subject producer reduces its price by 11.6 percent after the tariff is imposed, slightly counteracting the 25 percent ad valorem tariff. Still, consumers purchase 21.8 percent lower quantities from the subject producer. As expected, the domestic and non-subject producer benefit from the imposition of the tariff. Both domestic and non-subject producers are able to raise their prices by 1.10 percent while consumers increase the quantities purchased from both non-subject sources by 2.24 percent.

4.3 Model simulation: Bertrand

The following code runs a PE simulation under a Bertrand competition for data given in Table 4:

```
1 #Import the Bertrand Competition model from PyPE
2 from pype.imperfect_competition.bertrand_model import Bertrand
3
4 bertrand_model=Bertrand(parameter_matrix = producer_data ,
5                           elasticity = 3.0,
6                           industry_elasticity = -1,
7                           output_path = "DesiredFilePath.xlsx")
```

Table 4 contains results from the PyPE simulation of a market characterized by Bertrand imperfect competition. The results table features an additional **Marginal Costs** column, which is each producers' marginal costs calibrated using initial market conditions. Intuitively, the firm with the largest market share, A, features the lowest marginal costs. More generally, we see that calibrated marginal costs are inversely related to producers' initial market share.

The 25 percent tariff change leads to large changes in the market characterized by Bertrand competition. Each producer experiences significant changes in consumer expenditures with the subject firm seeing a drop in revenues of nearly 22 percent, due to both a reduction in prices and quantities. The subject producer reduces its prices, however by a

substantially smaller level than the magnitude required to fully cover the 25 percent tariff. As such, consumer prices for the subject product increases. Consumer expenditures on domestic producer increased nearly 4 percent with the domestic producer able to increase its prices by 6 percent. There is a larger effect on consumer expenditures on the non-subject firm mostly due to an increase in quantities sold. These pricing strategies represent a notable contrast to the relatively uniform changes in consumer prices across each firm in the monopolistic and perfect competition scenario.

Table 4: Results from simulations of Bertrand Competition model

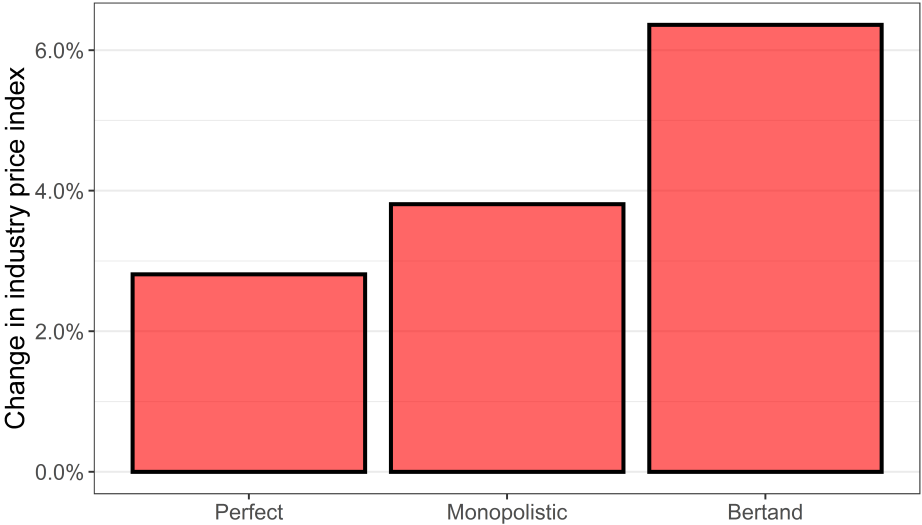
Source	Marginal Costs	expenditures0	expenditures1	quantities0	quantities1	expenditures % change	prices % change	quantities % change
Domestic	0.38	70.00	72.75	70.00	68.43	3.93	6.31	-2.24
Non-subject	0.64	10.00	11.60	10.00	11.52	15.97	0.65	15.23
Subject	0.62	20.00	15.65	20.00	12.77	-21.75	-1.98	-36.13

4.4 Summarizing competition-specific models

Figure 1 depicts the change in the CES industry price indices across the different modeling scenarios. Intuitively, we see that industry-wide prices are most affected in the Bertrand simulation, and most muted in the perfect competition model. The change in the industry price observed in the monopolistic competition scenario nearly represents a midpoint between perfect competition and Bertrand scenarios.

A more robust discussion of trade-offs and appropriate applications of the various competition-specific models currently housed in the PyPe framework is beyond the scope of this introductory paper. However, figure 2 provides comparisons of simulated changes in industry price indices and major producer-specific outcomes across the different competition-specific modeling scenarios. As the figure shows, the choice of type of competition to be applied to a modeling scenario does have notable implications for how firms adjust prices and consumers respond in the wake of a change in tariff policy. From the figure, the negative effects on quantity and expenditure for the subject producer are larger in the monopolistic competi-

Figure 1: Changes in CES price indices across simulated model scenarios



tion simulation compared to perfect and Bertrand competition scenarios. PyPE thus can be a helpful tool for understanding how these different types of competition can influence simulated outcomes. In the future, PyPE developers plan to augment the package with new tools to allow model users to seamlessly produce visualizations and other analyses for performing this type of sensitivity analysis and compare outcomes across a variety of simulations. Ultimately, we believe the ease of which users can access and compare different model simulations within a single PyPE framework can help end users generate interesting new insights into the implications of modeling assumptions for specific research questions.

5 Future Modifications

Regarding future development, PyPE will focus on two distinct areas. For end users and individuals with less programming and modeling experience, developers will continue to work on expanding the suite of “off-the-shelf” models and model features so that PyPE has more ability to analyze different markets and sectors.

Figure 2: Changes in producer-level outcomes across simulated model scenarios



In the immediate future, PyPE developers will focus on enabling:

- a Nested-CES Option with user-defined nest assignment,
- non-CES demand structures such as linear demand,
- new competition modules such as Cournot competition, and
- additional user features such as sensitivity analysis and results visualizations

As PyPE is an open source project, it stands to benefit from contributions and collaborations with the larger PE modeling community. To that end, PyPE developers will continuously encourage modelers to use the package to “Pythonize” existing PE models or create new ones. In doing so, contributors will be adding their work to an ever-growing repository of models while helping develop new tools and frameworks for future model building efforts and policy applications.

References

- Ahmad, S. (2019). Conducting profitability analysis in partial equilibrium models with monopolistic competition. USITC Office of Economics Research Note RN-2019-07B, United States International Trade Commission, Washington, DC.
- Armington, P. S. (1969). A theory of demand for products distinguished by place of production. *Staff Papers* 16(1), 159–178.
- Francois, J. F. and H. K. Hall (1997). Partial equilibrium modeling. *Applied methods for trade policy analysis: A handbook*, 122–155.
- Krugman, P. R. (1979). Increasing returns, monopolistic competition, and international trade. *Journal of international Economics* 9(4), 469–479.
- Riker, D. (2018). Modeling the financial impact of tariffs in concentrated product markets. USITC Office of Economics Research Note RN-2018-10B, United States International Trade Commission, Washington, DC.
- USITC (2019). U.s.-mexico-canada trade agreement: Likely impact on the u.s. economy and on specific industry sectors. Investigation Number: TPA 105-003 publication 4889, United States International Trade Commission, Washington, DC.
- USITC (2021a). Caribbean basin economic recovery act: Impact on u.s. industries and consumers and on beneficiary countries. Investigation Number: 332-227 publication 5231, United States International Trade Commission, Washington, DC.
- USITC (2021b). Raspberries for processing: Conditions of competition between u.s. and foreign suppliers, with a focus on washington state. Investigation Number: 332-577 publication 5194, United States International Trade Commission, Washington, DC.