Microeconomics: Theory, Methodology, and Application

Editor
Assoc. Prof. Melahat Batu Ağırkaya, Ph.D.

DOI: 110.5281/zenodo.17396087

Microeconomics: Theory, Methodology, and Application

Editor

Assoc. Prof. Melahat Batu Ağırkaya, Ph.D.

Publisher
Platanus Publishing®

Editors in Chief
Assoc. Prof. Melahat Batu Ağırkaya, Ph.D.

Cover & Interior Design
Platanus Publishing®

The First Edition
October, 2025

ISBN

978-625-6454-13-2

Platanus Publishing®
Address: Natoyolu Cad. Fahri Korutürk Mah. 157/B, 06480, Mamak, Ankara, Turkey.

Phone: +90 312 390 1 118
web: www.platanuspublishing.com
e-mail: platanuskitap@gmail.com

Platanus Publishing®

CONTENTS

Chapter 1

Forced Migration and Sustaining the Well-Being of Refugees: Evidence From Ukraine

Orhan Aygün[1] & Aysun Hızıroğlu Aygün[2]

As of 2024, approximately 6% of the world's population is on the move, driven by factors such as economic hardship, political unrest, and natural disasters. These circumstances often lead to refugee crises and the displacement of millions of people, creating significant humanitarian challenges. Displaced individuals face immense hardships, including losing their homes and separation from family (Ibáñez and Vélez, 2008; Fiala, 2015). Unlike economic migrants who typically have the time and ability to plan their movement, those fleeing crises often embark on sudden and unplanned journeys to the nearest safe destination. In their new settlements, they face difficulties finding a decent job, including the language barrier, lack of legal work status, skill mismatch, and unfamiliarity with the host country's labor market (Lindley, 2002). In the case of forced migration, hosting countries also experience strains on their resources and infrastructure as they struggle to provide adequate housing, healthcare, and employment opportunities (Roza and Sviatschi, 2021; Hızıroğlu Aygün et al., 2021). Addressing these complex issues requires coordinated international efforts, including humanitarian aid, robust asylum policies, and long-term solutions to promote the integration of refugees into the host countries.

International organizations and the governments of the hosting countries support refugees by assessing their needs and providing urgent help in terms of humanitarian cash transfers. Refugees can benefit from cash transfers, especially during settlement (Ulrichs et al., 2017). The cash transfers could be particularly beneficial in situations where refugees' access to paid employment is limited, like the case of Syrian refugees in Turkey, whose formal employment was not allowed until 2016 (Hızıroğlu Aygün et al. 2024). However, relying on humanitarian assistance and other types of transfers is insufficient for sustaining the refugee families' socioeconomic well-being. In the later stages of their integration, the refugee families willing to support themselves should be supported via complementary policies. Frattini et al. (2018) suggest that reducing uncertainty

[1] Dr. Öğr. Üyesi, Boğaziçi Üniversitesi, İktisat Bölümü, ORCID: 0000-0003-0132-3429
[2] Doç. Dr., İstanbul Teknik Üniversitesi, Ekonomi Bölümü, ORCID: 0000-0003-4513-1859

about the refugee status and allowing residential mobility would ease the integration of refugees across European countries.

Job matching policies or other mechanisms that match refugees with the host regions could help refugees overcome the difficulties they face in finding employment opportunities in their new countries. However, a central mechanism enabling efficient matching of the refugees with the host countries does not exist. Therefore, the existing relocations may not always lead to the best outcomes for refugees or the host economies.

Even though forced-displacement is not motivated by the same reasons as economic migration, these situations too may include a selection of the host country and the region of residence within the host country. In the case of the Syrian refugees fleeing from the civil war, proximity to the province of origin was an important criterion for choosing the place to move to in the early stages. However, as Aksu et al. (2022) showed, refugees later moved to more developed provinces, which could provide broader employment opportunities within Turkey. Resettlement is difficult for refugees, especially considering the deprivation they experience as a result of the factors that led to their displacement. Moreover, refugees have limited income and lack access to the necessary resources to organize the movement of their family members. In the case of the temporariness of migration, employers also lack the willingness to employ refugees and incorporate them into the local labor markets. Therefore, a more organized method of help, starting with selecting a destination of migration, could improve refugee families' well-being and offer a long-term solution for their integration into the local labor markets.

A recent example of forced population movements is the emergency in Ukraine resulting from the war that started with Russia's invasion of Ukraine in February 2022. As a response to the emergency situation caused by the Ukrainian humanitarian crisis, a Regional Refugee Response Plan (RRP) was put in effect with an inter-agency partnership (UNHCR, 2022). Coordinated by the UNHCR, the RRP partners provided the first response to the emergency situation at the border points and neighboring countries. The partnership's objectives include ensuring the safety and international protection of the refugees and providing humanitarian assistance.

As the refugee crisis in Ukraine reached its third year, 3.7 million people in Ukraine were displaced, and 6.3 million Ukrainian refugees resided abroad. European countries host almost 6 million refugees from Ukraine (UNHCR, 2024). As the war continues, RRP emphasizes the need for the socioeconomic integration of refugees into the host countries and the reinforcement of social cohesion between the refugees and host countries (UNHCR, 2024).

In this chapter, we first discuss the institutional context for the relocation of asylum seekers in the European Union (EU) and the mechanisms for accommodating refugees across European countries. Then, we investigate the employment opportunities of the Ukrainian refugees who had to leave their country amid the war with Russia. We use a micro-level dataset collected by the UNHCR (UNHCR, 2023a) to discuss the current situation. Our focus is on the efficiency of the existing mechanism through which refugees usually choose their destination country in the absence of a central mechanism for matching host countries with the refugees.

The Legal Framework for the Asylum Seekers in Europe: Dublin III Regulations

Involuntary movement comes with many difficulties, including lacking job security and difficulty maintaining a living standard. Therefore, choosing an optimal place to migrate is crucial for overcoming adaptation challenges and mismatch problems in the labor market. EU regulations, namely the Dublin III regulations (EU Regulation No 604/2013), require asylum seekers to apply in the EU country they first entered, restricting them from traveling between EU countries.

Naturally, EU-border countries such as Greece and Italy are the initial destinations for protection seekers, resulting in a substantial bureaucratic and economic burden for these nations. This became a prominent problem within the EU following the 2015 refugee crisis as the number of asylum seekers surged in Europe. Moreover, confining asylum seekers to these countries causes labor market imbalances, with asylum seekers often struggling to build new lives or making illegal secondary movements to other EU countries. Facing these movements, the countries within the Dublin system can request outgoing transfers and receive incoming transfers of asylum seekers. In 2022, under the Dublin Regulation, Germany (68,706) and France (44,881) reported the largest number of outgoing requests to other EU Member States. Italy received the highest number of incoming requests (32,797), followed by Austria (24,455). However, only 4,158 outgoing and 3,699 incoming transfers were reported by Germany, and 3,311 outgoing and 1,453 incoming transfers were reported by France (EUROSTAT, 2023). This gap between the requested and actual transfers is cited as proof of the shortcomings of the Dublin III procedures (AIDA, 2023). These shortcomings of Dublin III led to inefficiencies in matching asylum seekers with EU countries.

To address this issue, the EU Commission proposed updating the Dublin III regulations and enforcing Dublin IV in 2016. Under this new regulation, other EU countries would share the burden based on a quota system in cases of disproportionate applications (Tubakovic, 2017). This “corrective allocation mechanism" would have ensured that all member states share the bureaucratic and economic burden of border countries while also creating new opportunities for asylum seekers. This was presented as a potential to improve matching quality to help combat labor market inefficiencies.

Matching quality is crucial for the economic and sociological welfare of both hosting countries and asylum seekers. In matching theory, a field of economics, stability in matching is essential for efficiency. Gale and Shapley (1962), in their individual-institution matching problem, defined a stable match as a situation where there is no unilateral or bilateral objection. In this context, stability means that no country or individual receives an unacceptable match, and no individual voluntarily makes a secondary move to another country. The literature shows that stable matching is efficient according to individual and institutional preferences.

Recent studies in the literature, such as Delacretaz et al. (2023) and Ahani et al. (2021), address the asylum matching problem by designing centralized matching mechanisms that consider individual and country preferences based on application processes. These mechanisms aim to create stable matchings, preventing inefficiencies and secondary movements. While centralized, these mechanisms allow countries to form their own preferences, ensuring the independence of member states.

Refugees from Ukraine

Russia's invasion of Ukraine on February 24th, 2022, started an ongoing humanitarian crisis, as a result of which millions of Ukrainians left their countries or were displaced internally. Ukrainian refugee families mainly moved to neighboring countries in Europe with health and safety concerns.

The Temporary Protection Directive for Refugees from Ukraine enables legal status and rights for refugees from Ukraine in EU countries. The legal status of refugees from Ukraine is different than that of asylum seekers in Europe. As the Temporary Protection Directive was invoked, they were given "the residency rights, access to the labour market, access to housing, social welfare assistance, medical or other assistance, and means of subsistence." by the EU. This is intended to reduce the burden on the national asylum processes. It is also expected with the Temporary Protection Drive that the flexibility to move across Europe would balance the effort to host the refugees across EU countries.[3]

Thanks to the Temporary Protection Directive, the situation in Ukraine did not directly affect Dublin transfers as the EU countries did not apply the transfer procedures for temporary protection. However, the Ukrainian refugee crisis

3 https://migrant-integration.ec.europa.eu/news/eu-invokes-temporary-protection-directive-help-those-fleeing-ukraine_en

indirectly affected the transfer system as the refugee-hosting neighbor countries suspended their Dublin transfers due to staff shortages (AIDA, 2023).

Table 1 shows the number of Ukrainians with temporary protection by country across Europe. It provides a detailed overview of how different countries, both neighboring and non-neighboring, have accommodated Ukrainian refugees in 2024 as a percentage of their total population in 2023. This data, sourced from EUROSTAT and the UNHCR, reflects the distribution of Ukrainian refugees across various European nations, highlighting the significant variations in refugee hosting capacities and the corresponding demographic implications.

Table 1: Number of People from Ukraine in the Neighboring and Non-Neighboring Countries

Country	Ukrainians with Temporary Protection	% of the Population
Neighboring Countries		
Poland	953,930	2.6
Rep. of Moldova	115,000*	4*
Romania	154,870	0.8
Slovakia	119,675	2.2
Non-Neighboring Countries		
Bulgaria	50,970	0.8
Czech Republic	345,405	3
France	62,905	0.9
Germany	971,430**	1.15
Italy	164,725	0.3
Lithuania	77,145	2.7

Source: EUROSTAT, non-EU citizens who fled Ukraine and were under temporary protection as of April 2024. The population data is obtained from the EUROSTAT for the year 2023.
* Statistics for Moldova are obtained from UNHCR's report, which shows the number in December 2023. Numbers for Germany are obtained from UNHCR's website, and the numbers are shown in September 2023.

Among neighboring countries, Poland hosts 953,930 Ukrainians, making up 2.6% of its population, while the Republic of Moldova has 115,000 Ukrainians, constituting approximately 4% of its population. Romania hosts 154,870 Ukrainians, accounting for 0.8% of its population, and Slovakia has 119,675 Ukrainians, making up 2.2% of its population.

Among non-neighboring countries, Bulgaria hosts 50,970 Ukrainians, representing 0.8% of its population. The Czech Republic provides protection to 345,405 Ukrainians, constituting 3% of its population. France hosts 62,905 Ukrainians, accounting for 0.9% of its population, while Germany has 971,430 Ukrainians, making up 1.15% of its population. Italy hosts 164,725 Ukrainians, representing 0.3% of its population, and Lithuania has 77,145 Ukrainians, constituting 2.7% of its population.

Ukrainian refugees' introduction to the hosting labor markets happened smoothly compared to others, such as the Syrian refugees (Ludek et al., 2023). The Ukrainian refugees' higher education and English language knowledge most likely explain the easier transition. Also, some of the neighboring countries' labor markets were acquainted with the Ukrainian workers as before the war began, Ukrainian immigrants took up the majority of foreign employment in some of these hosting countries (Duszczyk and Kaczmarczyk, 2022). However, for the refugees fleeing the war, skill mismatch could still pose a difficulty for entry into the labor market (Bannikova, 2022). Moreover, the refugee households consisted of children and women, as prime-age men were not allowed to leave the country. Therefore, as single parents, refugee women's ability to work was limited as they tried to start a new life with children who needed to be taken care of and supported due to the war's effects on their physical and mental health.

In the fourth round of "the Intentions and Perspectives of Refugees from Ukraine" survey, UNHCR collected data from 3827 refugee households in Europe between April and May 2023. The survey sample in the neighboring countries consisted of beneficiaries of the UNHCR for multi-purpose cash assistance. One member from each household in the sample was surveyed about their situation, living arrangements in the host country, and intentions for the future. Data was collected via phone surveys in the four neighboring countries which are Moldova, Poland, Romania, and Slovakia. In the remaining countries across Europe, data was collected through online surveys. More details about the sampling strategy for the data collection can be found in UNHCR's Lives on Hold: Intentions and Perspectives of Refugees from Abroad report (UNHCR, 2023b). Since the data collection and sampling methods were different for the Ukrainian refugees in the neighboring and non-neighbor countries, the data does not allow for comparison across these two groups so we report and interpret the analysis for these two groups in separate tables.

Tables 2 and 3 show the summary statistics for the respondents below age 60 in our dataset in the neighboring and non-neighboring countries, respectively. We limit the sample to respondents who are younger than 60 because our main focus for discussing refugees' settlement efficiency is through employment.

Table 2 provides a detailed overview of Ukrainian refugees in neighboring countries, including their labor market status, household size, educational attainment, intentions to return to Ukraine, host country distribution, age, and gender. The data shows that 34% of the respondents are formally employed, while 12.6% are in informal employment, 15.3% are unemployed, and 30.4% are not in the labor force. A small percentage are engaged in self or family employment (0.9%) or are in training/school (2.8%). 39% of the employed stated that their current occupation is at the same or higher level of the previous job they had in Ukraine.

Intentions to return to Ukraine are high, with 69.6% planning to return permanently sometime in the future, 16% intending to return within three months, and 6% undecided. Only 4.3% have decided not to return.

Table 2: Summary Statistics for the Respondents in the Neighboring Countries

Labor Market Status	Mean	S.D.	***Household size***	Mean	S.D.
Formal Employment	0.340	0.474	1	0.113	0.316
Informal Employment	0.126	0.332	2	0.281	0.449
Unemployed	0.153	0.360	3	0.310	0.463
Self/Family Employment	0.009	0.093	4	0.160	0.367
Training/School	0.028	0.166	5 or more	0.129	0.335
Not in LF	0.304	0.460	NA	0.007	0.086
Other	0.039	0.195			
Occupation in Comparison to Job in Ukraine					
Same or Higher Level*	0.388	0.488	***Educational attainment***		
Intentions to go to Ukraine			Lower	0.224	0.417
Yes	0.696	0.460	Technical or Vocational	0.191	0.393
Yes, in 3 months	0.160	0.367	Bachelor or higher	0.584	0.493
No	0.043	0.203	Prefer not to answer	0.001	0.025
Undecided	0.060	0.238	***Host country***		

Prefer not to answer	0.040	0.196	Poland	0.256	0.437
Age			Moldova	0.233	0.423
18-34 yrs	0.291	0.454	Romania	0.269	0.443
35-59 yrs	0.709	0.454	Slovakia	0.242	0.428
Female	0.914	0.280			

Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 1,622. * Observation number for the employed respondents is 783.

Regarding educational attainment, 22.4% have lower education, 19.1% have technical or vocational training, and 58.4% hold a bachelor's degree or higher. Household sizes vary, with 11.3% consisting of single individuals, 28.1% having two members and 31% having three members. Age distribution shows that 29.1% of the respondents are between 18 and 34 years old, while 70.9% are between 35 and 59 years old. The majority of the respondents are female, comprising 91.4% of the sample.

The host country distribution indicates that 25.6% of the refugees are in Poland, 23.3% in Moldova, 26.9% in Romania, and 24.2% in Slovakia. This comprehensive data set is sourced from the UNHCR's "Survey of Intentions and Perspectives of Refugees from Ukraine #4" microdata.

Table 3 offers a similar summary of statistics of Ukrainian refugees in non-neighboring countries, encompassing their socioeconomic integration and demographic profiles. The data shows a diverse engagement in the labor market, with 30.3% in formal employment, 9.0% in informal employment, and 23.6% currently unemployed. Table 3 also highlights that 14.7% of Ukrainian refugees in non-neighboring countries are currently in training or school. Additionally, 13.2% are categorized as not in the labor force, indicating a group focusing on household duties, caregiving, or other non-economic activities. 36% of the employed stated that their current occupation is at the same or higher level of the previous job they had in Ukraine.

Table 3: Summary Statistics for the Respondents in the Non-Neighboring Countries

Labor Market Status	Mean	S.D.	***Household size***	Mean	S.D.
Formal Employment	0.303	0.460	1	0.202	0.402
Informal Employment	0.090	0.287	2	0.262	0.440
Unemployed	0.236	0.425	3	0.256	0.437
Self/Family Employment	0.035	0.183	4	0.133	0.340
Training/School	0.147	0.354	5 or more	0.091	0.288
Not in LF	0.132	0.339	NA	0.055	0.227
Other	0.057	0.232			
Occupation in Comparison to Job in Ukraine					
Same or Higher Level*	0.359	0.480	***Educational attainment***		
Intentions to go to Ukraine			Lower	0.118	0.323
Yes	0.562	0.496	Technical or Vocational	0.192	0.394
Yes, in 3 months	0.139	0.346	Bachelor or higher	0.683	0.465
No	0.064	0.246	Prefer not to answer	0.006	0.080
Undecided	0.229	0.420	***Host country***		
Prefer not to answer	0.006	0.080	Bulgaria	0.041	0.199
Age			Czech Rep.	0.132	0.338
18-34 yrs	0.263	0.440	France	0.039	0.195
35-59 yrs	0.737	0.440	Georgia	0.044	0.205
Female	0.881	0.324	Germany	0.168	0.374
			Italy	0.079	0.270
			Lithuania	0.05	0.218
			Spain	0.043	0.203
			Turkey	0.056	0.230

Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 1,724. * Observation number for the employed respondents is 763.

A majority intend to return to Ukraine (56.2%), with 13.9% planning to do so within three months.

The refugee population is predominantly aged 35-59 years (73.7%), and females make up a significant majority (88.1%) of respondents. Educational attainment is notably high, with 68.3% holding bachelor's degrees or higher, indicating a highly educated refugee population. Household sizes vary, with significant proportions in two-person (26.2%) and three-person (25.6%) households.

Germany hosts the largest share of refugees (16.8%), followed by the Czech Republic (13.2%), and Italy (7.9%).

Analysis of the Refugees' Employment Status in the Place of Settlement

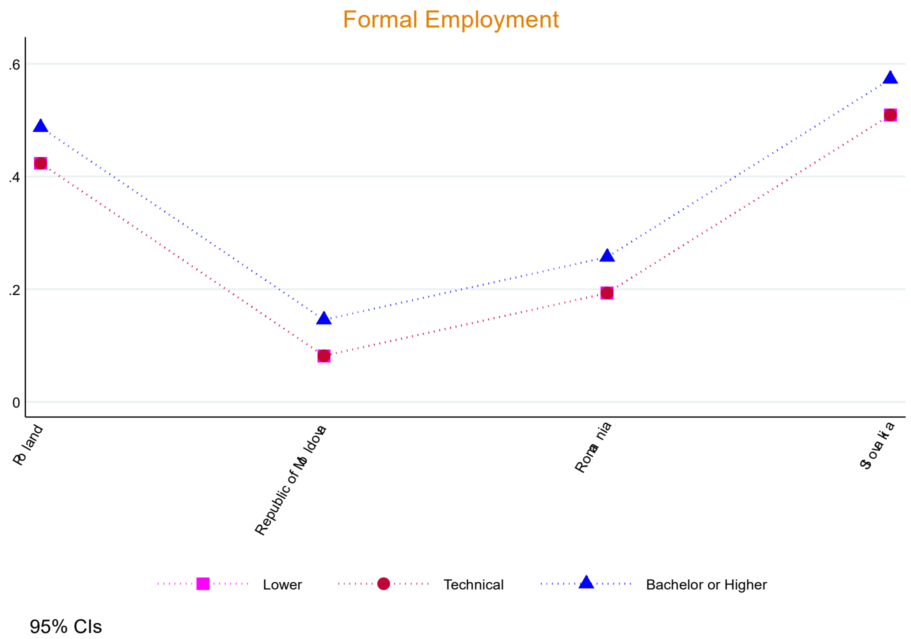
Next, we analyze if the refugees' settlements matches with their intentions, skills and management of family responsibilities in the neighboring countries. We focus on their labor market status as an outcome variable. We treat refugees' new settlements as a good match if the survey respondent reported being employed formally, informally or in self or family business in the host country. If these employment situations exist, we consider this to be a sign of an ability to adapt to the labor market of the hosting economy. Because the refugees who could support their families through employment by the time of data collection could manage movement and resettlement, understand the institutional necessities of the host country, manage the paperwork, and find employment, which we can interpret as a good selection of the country to migrate to. While this criterion doesn't fully gauge the efficiency of resettlement decisions, it does suggest their ability to successfully adapt and integrate into the new environment. This perspective is valuable for understanding the outcomes of refugee resettlement efforts. We also analyze another variable to measure the match of the skills of the refugees that is constituted of their views on the comparison of their current occupation with their previous job in Ukraine. We treat the jobs as a good match if the employed respondents stated that their current job is the same or higher level than their job in Ukraine.

We run a separate Ordinary Least Squares (OLS) regressions for each outcome variable. We control for the respondent's age, educational attainment, gender, age category, household size and country of residence in all regressions. We report the predicted probabilities for each outcome variable by country at educational attainment categories.

Figure 1 illustrates the predicted probabilities of formal employment for refugees in neighboring countries. The highest predicted probability is in Slovakia, followed by Poland, with nearly 60% among those holding a bachelor's degree or higher. Conversely, refugees in the Republic of Moldova[4] and Romania exhibit significantly lower predicted probabilities of formal employment.

[4] Like the EU countries, Moldova also offers temporary protection to the refugees from Ukraine which provide rights to work.

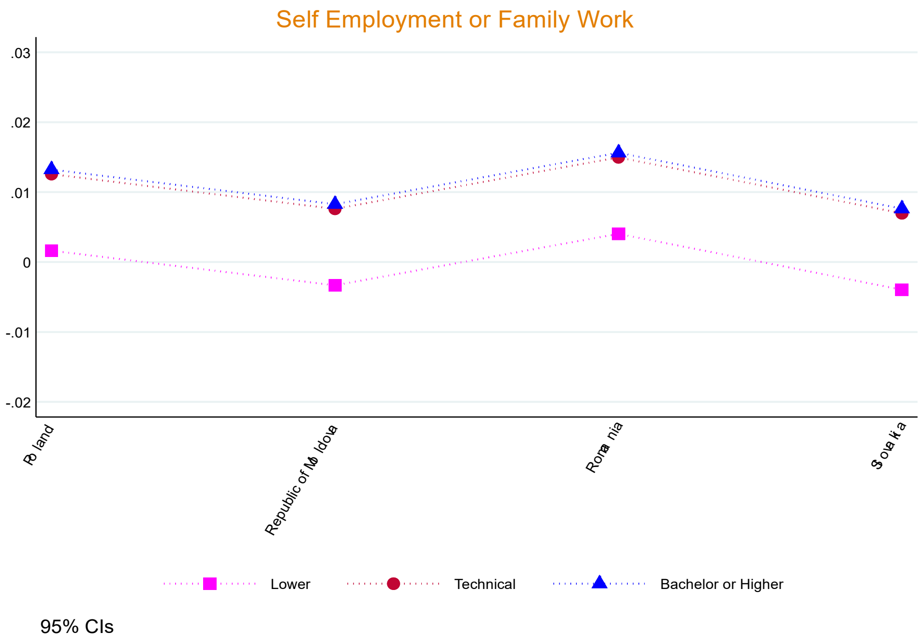
Figure 1: Predicted Probabilities for Formal Employment of Ukrainian Refugees in the Neighboring Countries



Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 1,622.

Controlling for variables such as gender, age, educational attainment, and household composition, the disparities in these predicted probabilities suggest that not all refugee-country matches are efficient concerning labor market integration. If the matches were more efficient, we would observe more uniform predicted probabilities across these neighboring countries, as more refugees who are able and willing to work would choose to settle in countries with higher predicted probabilities of formal employment.

Figure 2: Predicted Probabilities for Self-Employment or Family Work of Ukrainian Refugees in the Neighboring Countries

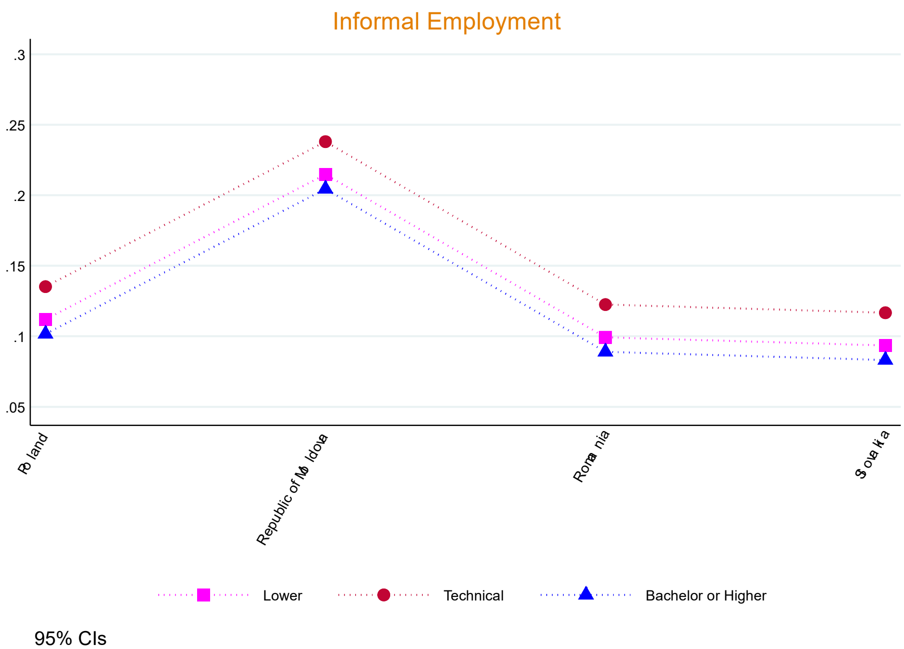


Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 1,622.

One of the major disruptions in forced migration is the loss of assets, making refugees less likely to engage in self-employment. This pattern holds true for Ukrainian refugees in neighboring countries, as illustrated in Figure 2. The predicted probabilities for self-employment across all educational attainment categories in four neighboring countries we studied are exceedingly low, ranging from 0 to 0.02%. There is minimal variation across these countries, though refugees with higher education or technical schooling in Romania have a slightly higher predicted probability of self-employment.

Next, we estimate the predicted probabilities for the likelihood of holding an informal job. As shown in Figure 3, Moldova has the highest predicted probability of having an informal job among the neighboring countries we studied. Despite Moldova granting temporary protection to Ukrainian refugees with the right to work, similar to the status offered by the EU, the predicted probability of working in informal jobs is substantially higher in Moldova. One likely explanation is the high ratio of informal employment in the Moldovan labor market which was recorded as 25% in 2018 (ILO, 2018). Conversely, the predicted probabilities in Poland, Romania, and Slovakia are very similar, as depicted in Figure 3.

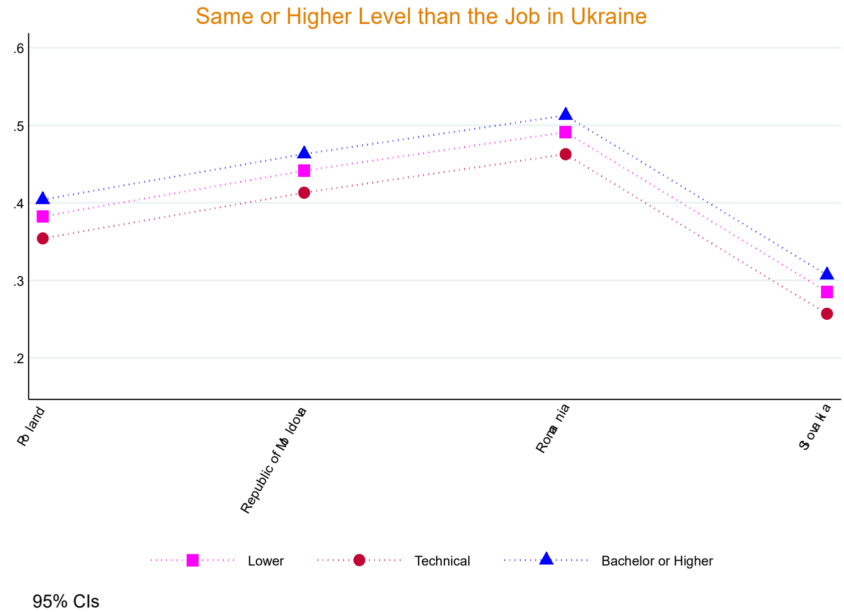
Figure 3: Predicted Probabilities for Informal Employment of Ukrainian Refugees in the Neighboring Countries



Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 1,622.

Finally, we examine the predicted probability of working in an occupation at the same or higher level in the host country compared to the previous occupation held in Ukraine. This outcome variable applies only to individuals employed in their new settlement. Figure 4 shows the predicted probabilities for this outcome variable. The predicted probability of working in the same or higher level occupation is as high as 50% in Romania, followed by Moldova and Poland. In contrast, only 30% of employed refugees in Slovakia believe their current occupation is on par with or better than their previous occupation in Ukraine.

Figure 4: Predicted Probabilities for Having an Occupation with the Same or Higher Level as the Previous among the Employed Ukrainian Refugees in the Neighboring Countries



Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 783.

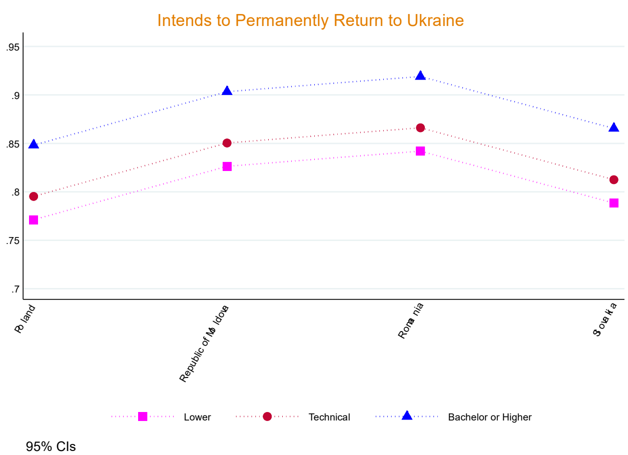
Overall, our analysis results indicate that the labor market outcomes for the Ukrainian refugees across different countries show great variability. This case is present where the refugees have the same status (i.e. temporary protection) across EU and Moldova. Our results suggest that it is hard for the refugees to resettle and move to a country with a higher prospect of holding formal employment or a job that is higher or similar level as the one they previously had. This would be even harder for the asylum seekers with restricted relocation rights via the Dublin procedure. Therefore, a central mechanism that would allocate the refugees to the host countries by the needs and capacity of the labor market and the refugees' ability and willingness to work would improve the efficiency of the labor markets in the case of a refugee crisis from forced migration. As a final exercise, we investigate the refugees' intentions to return to Ukraine permanently, as shown in Figure 5. As the figure shows, the predicted probability of having the intention to go back to Ukraine permanently is quite high. In all countries, refugees with a bachelor's or higher have the highest predicted probability of going back home one day. The countries where the predicted probability of formal refugee employment is high have a lower likelihood of hosting refugees who want to return to Ukraine permanently. Even though we cannot comment on the direction of causality between these two outcomes, we present it as suggestive evidence of the association between refugees' intentions to relocate and labor market status.

Overall, our analysis indicates significant variability in labor market outcomes for Ukrainian refugees across different countries despite having the same status (i.e., temporary protection) in the EU and Moldova. Our findings suggest that it is challenging for refugees to resettle in a country with better prospects for formal employment or a job equivalent to or better than their previous occupation. This challenge is even greater for asylum seekers with restricted relocation rights under the Dublin procedure.

Therefore, implementing a central mechanism to allocate refugees to host countries based on labor market needs and the refugees' ability and willingness to work could enhance labor market efficiency during a refugee crisis caused by forced migration.

As a final exercise, we investigate the refugees' intentions to return to Ukraine permanently, as depicted in Figure 5. The predicted probability of intending to return to Ukraine permanently is quite high in all the neighboring countries we studied. Refugees with a bachelor's degree or higher have the highest predicted probability of returning home one day. Interestingly, countries with higher predicted probabilities of formal refugee employment have a lower likelihood of hosting refugees who intend to return to Ukraine permanently. Although we cannot establish causality between these two outcomes, this finding suggests an association between refugees' intentions to relocate and their labor market status.

Figure 5: Predicted Probabilities for the Intentions to Return Ukraine among the Employed Ukrainian Refugees in the Neighboring Countries



Source: UNHCR, Survey of intentions and perspectives of refugees from Ukraine #4 microdata. Sample size is 1,622.

Conclusion

The Ukrainian refugee crisis, now in its third year, has underscored the limitations and inefficiencies of existing mechanisms for refugee resettlement across Europe. Despite the legal clarity and temporary protection offered by EU countries and Moldova, our analysis reveals that labor market outcomes for Ukrainian refugees vary significantly across host nations. These disparities highlight the challenges that arise in the absence of a centralized, preference-based matching system that accounts for both refugee characteristics and host country capacities.

While refugees' formal employment rates and occupational matching outcomes are higher in some countries—particularly where prior labor market linkages existed—the overall data suggest that many refugees are not able to realize their full economic potential in the labor markets of host countries. This misalignment indicates that even with a harmonized legal framework, spontaneous settlement patterns do not necessarily lead to efficient or equitable outcomes.

Moreover, the high intention to return among educated refugees suggests a missed opportunity for longer-term integration that could benefit both refugees and host economies. Refugees with better employment outcomes are less likely to want to return, suggesting that stronger labor market integration may also promote social cohesion and longer-term stability.

These findings collectively point to the need for a coordinated European refugee allocation mechanism—one that incorporates principles from matching theory to improve stability and efficiency. Such a mechanism could better align refugee skills and preferences with host country labor market demands and institutional capacities. The experience from the Ukrainian crisis also provides a valuable benchmark for evaluating how future forced migration scenarios could be managed more effectively with centralized, data-informed, and preference-sensitive policies.

Ultimately, improving refugee well-being requires not only humanitarian support but also structural mechanisms that facilitate economic self-sufficiency and long-term social inclusion. As the global refugee population continues to grow, the insights from the Ukrainian crisis can inform more adaptive, just, and effective refugee policy frameworks worldwide.

References

Ahani, N., Andersson, T., Martinello, A., Teytelboym, A., & Trapp, A.C. (2021) Placement optimization in refugee resettlement. *Operations Research, 69(5)*, 1468-1486.

AIDA (2023). *The implementation of the Dublin III Regulation in 2022*. https://asylumineurope.org/wp-content/uploads/2023/11/AIDA_Dublin-Update-2022.pdf Accessed through the Link on June 28, 2024

Aksu, E., Erzan, R., & Kırdar, M. G. (2022). The impact of mass migration of Syrians on the Turkish labor market. *Labour Economics, 76*, 102183.

Aygün, A., Kırdar, M. G., & Tuncay, B. (2021). The effect of hosting 3.4 million refugees on native population mortality. *Journal of health economics, 80*, 102534.

Aygün, A. H., Kırdar, M. G., Koyuncu, M., & Stoeffler, Q. (2024). Keeping refugee children in school and out of work: Evidence from the world's largest humanitarian cash transfer program. *Journal of Development Economics*, 103266.

Bannikova, K. (2022). Ukrainian refugees and the European labor market: socio-cultural markers of interaction. *Amazonia Investiga, 11(56)*, 9-17.

Duszczyk, M., & Kaczmarczyk, P. (2022). Poland and war refugees from Ukraine–beyond pure aid. *CESifo Forum, ifo Institut-Leibniz-Institut für Wirtschaftsforschung an der Universität München. 23(4)*, pp. 36-40.

Delacretaz, D., Kominers, S.D., & Teytelboym, A. (2023). Matching mechanisms for refugee resettlement. *American Economic Review, 113(10)*, 2689-2717.

EUROSTAT (2023). *Statistics on countries responsible for asylum applications (Dublin Regulation)*. Accessed on June 26, 2024, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Statistics_on_countries_responsible_for_asylum_applications_(Dublin_Regulation)&oldid=631592#:~:text=In%202022%2C%20Germany%20and%20France,and%201%20453%20incoming%20transfers.

Fiala, N. (2015). Economic consequences of forced displacement. *The Journal of Development Studies, 51(10)*, 1275-1293.

Frattini, T, L Minale and F Fasani (April, 9 2018), *(The struggle for) refugee integration into the labour market: Evidence from Europe*, Accessed on June 28, 2024, VoxEU.org.

Ibáñez, A. M., and Vélez, C. E. (2008). Civil conflict and forced migration: The micro determinants and welfare losses of displacement in Colombia. *World Development, 36(4)*, 659-676.

ILO (2020). *Overview of the informal economy in the Republic of Moldova.* Accessed on June 28, 2024. https://www.ilo.org/resource/overview-informal-economy-republic-moldova

Lindley, J. K. (2002). Economic assimilation and the labour market performance of British refugees and economic migrants. *Leverhulme Centre for Research on Globalisation and Economic Policy.*

Ludek, J., Kamionka, M., & Macková, L. (2023). High-skilled precarity: The situation of Ukrainian refugees in the Czech Republic and Poland. *Sociological Studios, 2(23)*, 51-61.

Rozo, S. V., & Sviatschi, M. (2021). Is a refugee crisis a housing crisis? Only if housing supply is unresponsive. *Journal of Development Economics, 148*, 102563.

Tubakovic, T. (2017). *A Dublin IV recast: A new and improved system?*. EGMONT Royal Institute for International Relations.

Ulrichs, M., Hagen-Zanker, J., & Holmes, R. (2022). *Cash transfers for refugees. An opportunity to bridge the gap between humanitarian assistance and social protection.* ODI.

UNHCR (2022). *Regional Refugee Response Plan. Summary and Inter-Agency Funding Requirements March-August 2022*. Accessed on June 28, 2024. https://data.unhcr.org/en/documents/details/91114

UNHCR (2023a). *Belgium, Bulgaria, Czech Republic, Germany, France, Italy, Lithuania, Moldova, Poland, Romania, Slovak Republic, Spain, Turkey: Survey of intentions and perspectives of refugees from Ukraine #4 - June 2023*. Accessed on June 28, 2024. from: https://microdata.unhcr.org

UNHCR (2023b*). Lives on hold: Intentions and Perspectives of Refugees and IDPs from Ukraine #4*. Accessed on June 28, 2024. https://data.unhcr.org/en/documents/details/101747

UNHCR (2024). *Regional Refugee Response for the Ukraine Situation.* Accessed on June 28, 2024. https://reporting.unhcr.org/ukraine-situation-regional-refugee-response-plan-6697

Chapter 2

Algorithm-Based Computable General Equilibrium Model: Python Application and Alternative Simulation Results

Muhammet Rıdvan İnce[1] & Ferhat Pehlivanoğlu[2] & Recep Tarı[3]

1. Introduction

Computational economics is an interdisciplinary science in which economics, mathematics, statistics and computer science are used together. Research areas include micro and macro-economic models and the topics such as computational econometrics and mathematical statistics. In general terms, it is aimed to explain the economic system created by complex human behaviors and the equilibrium states of the parts that make up this system by various mathematical models.

The mathematical modeling of all units representing the economy may bring along some difficult to solve problems. For instance, the modeling of a simple economy with a single household and two firms requires the simultaneous solution of 12 independent nonlinear equations including production, consumption and market clearing conditions, which is a difficult to solve problem. Therefore, it becomes inevitable to use various software for the solution of CGE models. GAMS, MPSGE and GEMPACK can be given as examples of the programs commonly used for the solution of CGE models. In this study, it was aimed to explain CGE modeling through Python, which is a general purpose software language that has recently become popular around the world. The use of Python, which also has very useful packages especially in areas such as data science, data analysis, mathematical modeling, and optimization problems, in CGE models may greatly expand the range of action of CGE modelers. Because the results obtained from CGE models can be used in many different ways with Python and can be integrated into different systems, and the results obtained with various data manipulations can be differentiated.

In this context, the study consists of three main sections. In the following two section, background, literature and information about the CGE model is provided,

[1] Kocaeli University, Department of Economics ORCID: 0000-0003-2050-2545
[2] Kocaeli University, Department of Economics, ORCID: 0000-0001-6930-0181
[3] Kocaeli University, Department of Economics, ORCID: 0000-0002-5616-0383

the equation system of the CGE model is established, and the social accounting matrix is introduced for a hypothetical economy. In the next section, after a general information about python, calibration and model solution algorithms are introduced. Furthermore, the algorithms are shown in appendix – 1 and appendix 2. In the final section, some simulation results are presented visually.

2. Background and the Key Literature Review of the CGE Models

CGE is a micro-based modeling in terms of the analysis of firm and household behaviors. On the other hand, macroeconomic identities are also used intensively. The model is basically based on the utility and profit maximization behaviors of households and firms, respectively, and macroeconomic factors such as income - spending, investment - saving, and the balance of foreign trade.

Accordingly, it can be said that economic analyses are performed based on two basic concepts of equilibrium. The first of them is the partial equilibrium analysis, and the other one is the general equilibrium analysis. In the partial equilibrium analysis, the other variables are considered to be constant based on the data of the area to be analyzed. In other words, various inferences are made only on the variables included in the analysis, and it is assumed that these variables have no effect on the other elements of the economy. Many econometric analyses are performed around this assumption and mentality. In the general equilibrium analysis, all elements of the economy are analyzed simultaneously. Unlike the partial equilibrium analysis, the effects of the shocks to any variable on the whole economy can be observed.

Both methods have their own advantages and disadvantages. Researchers choose the most appropriate method according to the type of analysis to be performed and the nature of the question to be answered. While the partial equilibrium analysis is mostly referred to by A. Marshall, the origins of the general equilibrium analysis are based on L. Walras. Nevertheless, Kenneth Arrow and Gerard Debreu (1954) made the most important theoretical contribution for the general equilibrium analysis to have the current computable general equilibrium (CGE) structure. On the other hand, leading CGE models, which are the first empirical applications of this theoretical knowledge, were developed by Leif Johansen (1960) and Herbert Scarf (1969). John Shoven and John Whalley (1973, 1977), who were the students of Scarf, demonstrated the empirical power of CGE models with their studies on taxation. After that date, CGE models have been applied to many areas such as:

- Climate change and environmental issues (Paul Block et al. 2006; Joshue Elliot et al. 2010),

- Consumption decisions of households (C. Pohl Nielson, Karen Thierfelder and Sherman Robinson 2001),
- Disease and pandemic (Channing Arndt 2002; and Xinshen Diao 2009),
- Issues related to international trade (Channing Arndt et al. 1997; Nancy Benjamin, Shantayanan Devarajan, and Robert J. Weiner 1989; Mary E. Burfisher, Sherman Robinson and Karen Thierfelder 1994; Peter Dixon, Maureen Rimmer and Marinos Tsigas 2007; Mark Gehlhar 1997; Lawrence Goulder and Barry Eichengreen 1989; Marjike Kuiper and Frank van Tongeren 2006),
- Tax related issues at national level (Shantayanan Devarajan, Karen Thierfelder and Sethaput Suthiwart-Narueput 2001; Sabine Jokisch and Laurence J. Kotlikoff 2005),
- Welfare analysis (Xinshen Diao, Rachid Doukkali and Bingxin Yu 2008; Karen Huff and Thomas W Hertel 2000),
- Other macroeconomic issues like migration, productivity etc. (World Bank 2006; Francois Bourguignon, Anne-Sophie Robilliard, and Sherman Robinson 2003; Andrea Cattaneo, Raul A. Hinojosa Ojeda, and Sherman Robinson 1999; Monica Verma and Thomas W. Hertel 2009).

3. CGE Model and Data Set

CGE models consist of systems of equations that describe the whole economy and various connections between its components. The model is based on equations such as supply, demand, utility and cost, which are directly obtained from economic theory. The equations consist of endogenous and exogenous variables and parameters. With the simultaneous solution of all equations in the model, a state of equilibrium, in which supply and demand are equalized in all markets depending on relative prices, is reached (Mary E. Burfisher 2011, 3).

In this study, a hypothetical economy, including two households consisting of those who mainly earn wage income $\{l\}$ and those who mainly earn capital income $\{k\}$, three sectors $\{a, s, i\}$ consisting of agriculture *(AGR),* service *(SERV)* and industry *(IND),* government and investment agent, was built. In this hypothetical economy, there are two production factors, including labor $(\bar{L})$ and capital $(\bar{K})$. The household has all production factors, rents them to firms through the factor market and receive wage (w) and capital (r) income in return. The household saves (S) some of this income (M) and transfers some of them to the government as a direct tax (T^d). Firms perform their productions by using the production factors in return for labor and wage payments and supply these products to the goods market. Labor and capital are perfect liquids between the sectors. The household, investment agent and the government make payments to the sectors in return for the goods they demand from the goods market, and these payments constitute the income of the sectors. The sectors pay taxes (indirect tax) T^z to the government at the rate of production tax over the production amount. Markets have a competitive nature. It is assumed that producers and consumers have no effect on prices. Finally, in the state of equilibrium, all goods and factor markets are cleared, in other words there is no excessive or deficient demand in the economy, and all factors of production are used.

3.1. CGE Equation System

The equation system of an economy with two households, three firms, investment agent, government, direct taxes and production taxes can be established as follows.

Objective Function;

$$max\, UU = \prod_l C_{la}^{\theta_{la}} + \prod_k C_{ka}^{\theta_{ka}}, \quad \forall a$$ (Total welfare: $Welf_l + Welf_k$)

Constraints;

Production Block *(24 equations)*

$$Y_s = b_s\, L_s^{\alpha_s} K_s^{\beta_s} \qquad \forall s \tag{1}$$

$$L_s = \frac{\alpha_s}{w} P_s^y Y_s \qquad \forall s \tag{2}$$

$$K_s = \frac{\beta_s}{r} P_s^y Y_s \qquad \forall s \tag{3}$$

$$INT_{as} = \frac{x_{as}}{P_a^c} Z_s \qquad \forall a, s, i \tag{4}$$

$$Y_s = y_s Z_s \qquad \forall s \tag{5}$$

$$p_s^z Z_s = y_s P_s^y + INT_{as}\, P_a^c + INT_{ss}\, P_s^c + INT_{is}\, P_i^c \qquad \forall s \tag{6}$$

Government Block *(10 equations)*

$$T_l^d = t_l^d M_l \qquad \forall l \tag{7}$$

$$T_a^z = t_a^z P_a^z Z_a \quad \forall a \tag{8}$$

$$M_g = (\textstyle\sum_l T_l^d) + (\textstyle\sum_a T_a^z)\,, \quad l = \{l,k\} \;\; a = \{a,s,i\} \tag{9}$$

$$S_g = mps_g\, M_g \tag{10}$$

$$G_a = \frac{\mu_a}{P_a^c} M_g(1 - mps_g) \quad \forall a \tag{11}$$

Investment / Saving Block *(6 equations)*

$$S_l = mps_l M_l \quad \forall l \tag{12}$$

$$S = (\textstyle\sum_l S_l) + S_g, \quad l = \{l,k\} \tag{13}$$

$$I_a = \frac{\lambda_a}{P_a^c} S \qquad \forall a \tag{14}$$

Household Block *(10 equations)* **(15)**

$$M_l = wl_w L + rl_r K \tag{16}$$

$$M_k = wk_w L + rk_r K \tag{17}$$

$$M_l^d = M_l\big(1 - mps_l - t_l^d\big) \qquad \forall l \tag{18}$$

$$C_{la} = \frac{\theta_{la}}{P_a^c} M_l^d \qquad \forall l, a \tag{19}$$

Market Clearing Conditions *(8 equations)*

$$Z_a = C_{la} + C_{ka} + I_a + INT_{aa} + INT_{as} + INT_{ai} \qquad \forall a \tag{20}$$

$$\bar{L} = L_a + L_s + L_i \tag{21}$$

$$\bar{K} = K_a + K_s + K_i \tag{22}$$

$$P_a^c = P_a^z(1 + t_a^z) \qquad \forall a \tag{23}$$

The simultaneous equation system between (1) - (23) above consists of 45 parameters, 2 exogenous variables, 58 equations and the same number of endogenous variables that need to be solved, where $s = \{a, s, i\}$ and $l = \{l, k\}$.

Parameters	$\alpha_s, \beta_s, b_s, x_{as}, y_s, t_l^d, t_a^z, mps_g, mps_l, \mu_a, \lambda_a, \theta_{la}, l_w, l_r,$
Exogenous Variables	$\bar{L}, \bar{K}$
Endogenous Variables	$Y_s, L_s, K_s, INT_{as}, T_l^d, T_a^z, M_g, S_g, G_a, S_l, S, I_a, M_l, M_l^d, C_{la},$ $P_a^y, P_a^z, P_a^c, r, w$

The production block consists of 24 equation systems. The production process is divided into two stages. In the first stage, sectors produce the composite factor with the $Cobb\ Douglass$ production function, and in the second stage, the sectoral output is obtained by combining the composite factor with intermediate inputs within the scope of the $Leontief$ function. In both stages, the aim of the firms is profit maximization. Within the equation system, where $s = \{a, s, i\}$, the following equation number (24) shows the composite factor production profit function of the sectors, and the equation number (1) shows the previously stated production technology constraint. Where, Y_s represents the composite factor production amount of the sector s, L_s and K_s represent the labor and capital factor demands, respectively, b_s; represents the technology parameter, α_s and β_s represent the input share parameters, P_s^y represents the composite factor price, w and r represent labor and capital prices, respectively ($\alpha_s + \beta_s = 1 \quad \forall s$).

$$\max \pi_s = P_s^y Y_s - (wL_s + rK_s) \qquad \mathbf{(24)}$$

$$Y_s = b_s\, L_s^{\alpha_s} K_s^{\beta_s} \qquad \mathbf{(1)}$$

If the optimization problem of the sectors that will maximize profits under technology constraint is solved for composite factor production amounts (Y_s), labor (L_s) and capital factor (K_i) by the Lagrange multiplier (ϑ_i), labor (2) and capital (3) factor demand functions are obtained for each firm.

$$\mathrm{L}(Y_s, L_s, K_s; \vartheta_s) \equiv \left(P_s^y Y_s - (wL_s + rK_s)\right) + \vartheta_s \left(b_s\, L_s^{\alpha_s} K_s^{\beta_s} - Y_s\right)$$

$$L_s = \frac{\alpha_s}{w}\, P_s^y Y_s \qquad \mathbf{(2)}$$

$$K_s = \frac{\beta_s}{r} P_s^y Y_s \qquad \mathbf{(3)}$$

Equations number (4), (5) and (6) in the equation system represent the second stage of production. Where, INT_{as}; shows the amount of intermediate inputs belonging to the sector a used by the sector s, Z_s; shows the output of the sector s, p_s^z; shows the output price of the sector s, x_{as} and y_s show the sector a input ratio and the composite factor ratio required for one unit of sector s output, respectively. Since perfect competition conditions are valid in all sectors, Equation (6) ensures that excess profit is zero. In this equation P_a^c, P_s^c and P_i^c represent the demand price of sectors.

Since it is assumed that the ownership of all factors of production in the economy belongs to the household, household incomes consist of factor incomes, including labor and capital. In the model, households are divided into two groups: those with (weighted) labor income$\{l\}$ and those with (weighted) capital income$\{k\}$. For instance, in the budget equation (16), which represent the total income of household, l_w parameter shows the ratio of wage income of the household in group l to total wage income in the economy, similarly, l_r parameter shows the ratio of capital income of the household in group l to total capital income in the economy. The aim of the household is utility maximization. Where $l = \{l, k\}$, the following equation (25) shows the household utility function and equation (26) shows the budget constraint. Where, C_{la}; represents the amounts of consumption (consumption of the goods a of the household with (weighted) labor income$\{l\}$), P_a^c; represents the consumer (demand) price of the goods a, M_l^d; represents the disposable income (equation (18)), θ_{la} represents the preference parameters for goods ($\sum_a \theta_{la} = 1$).

$$max(UU) = \prod_a C_{la}^{\theta_{la}} \tag{25}$$

$$\sum_a P_a^c C_{la} = M_l^d \tag{26}$$

If the household optimization problem, which will maximize its utility with alternative goods baskets under budget constraint, is solved for the consumption of goods (C_{la}) by the Lagrange multiplier (γ), the demand function for each good will be obtained as shown in equation (19).

$$\mathrm{L}(C_{la}; \gamma) \equiv \prod_a C_{la}^{\theta_{la}} + \gamma\left(M_l^d - \sum_a P_a^c C_{la}\right)$$

$$\frac{\partial \mathrm{L}}{\partial C_{la}} = \theta_{la} \frac{\prod_a C_{la}^{\theta_{la}}}{C_{la}} - \gamma P_a^c = 0 \quad \forall a$$

$$C_{la} = \frac{\theta_{la}}{P_a^c} M_l^d \qquad \forall a \qquad (\mathbf{19})$$

Since there is a production tax in the model, supply and demand prices differ. The difference between supply and demand prices is determined by the production tax rate (t_a^z) applied to the sectors.

$$P_a^c = P_a^z(1 + t_a^z) \qquad (\mathbf{23})$$

Macro closure rules are important in CGE models (Nobuhiro Hosoe, Kenji Gasawa , and Hideo Hashimoto 2010, 122). In this model, a savings-driven macro closure rule was adopted in terms of investment and savings. Namely, savings are determined exogenously within the marginal propensity to save (mps) and investment expenditures also vary depending on savings (Mathias Dewatripont et al. 1991, 145). Investment agent demands investment goods within the scope of distribution parameter λ_i by collecting household $\{S_l, S_k\}$ and government savings $\{S_g\}$ $(\sum_i \lambda_i = 1)$. As shown in Equation (14), the investment goods demand function and the household demand function are identical in terms of *mathematical format*. In this system of equations, S represents the total savings in the economy, which is the sum of household $(\sum_l S_l)$ and government savings (S_g).

In view of macro closure rules, there are also various alternatives with regard to government expenditures (Mathias Dewatripont and Gilles Michel 1987, 67). For instance, in a model in which the budget balance is assumed, government expenditures can be determined exogenously and tax rates can be determined endogenously. Alternatively a model that allows for a budget deficit can be established. In this model, it is assumed that government expenditures are determined endogenously and tax rates are determined exogenously within the scope of the budget balance. The government has two categories of tax revenues, direct(T_l^d, T_k^d) and indirect (T_a^z, T_s^z, T_i^z). In the model, it is also assumed that the government saves $\{S_g\}$. The government uses the net tax revenue (after savings are subtracted – the right hand side of equation (10)) to demand public goods from the sectors within the distribution parameter μ_i. As in the investment goods demand function, the public sector demand function (equation 11) is also identical to the household demand function in terms of mathematical format.

After the goods and factor demands are determined, the market clearing conditions should be met. In the model, in addition to 3 goods markets, there are two factor markets, labor and capital. Market clearing conditions will be provided by a set of goods prices that equalize the supply and demand quantities in the goods market and by at equilibrium wages (w) and interest rates (r) that equalize factor supply and demand in the factor market. Equations (20), (21) and (22) show the clearing requirements for the goods market, labor market and capital market, respectively. $\bar{L}$ and $\bar{K}$ are exogenous and represent the household labor and capital endowments, respectively. Equation (23) equates the supply and demand prices in all markets as stated before.

The Walras law is the most important issue to be considered in equation systems of CGE models. The Walras law indicates that the market $n.$ will be in equilibrium in the case that $(n-1)$ markets are in equilibrium in an economy of n markets (M. Alejandro Cardenete, Ana-Isabel Guerra, and Ferran Sancho 2012, 8). Therefore, one of the equations in the system of equations that expresses the market equilibrium condition is redundant, and if it is included in the model, the solution will not be achieved. Because mathematically, all equations must be independent from each other. Since the exclusion of one of the equations from the model will break the equality of number of equation and number of endogenous variable, one of the endogenous variables should be exogenously given to the model. Therefore, relative prices, not absolute prices, can be solved in CGE models. In this model, the wage level was chosen as the *numeraire* (Appendix – 1, row: 192).

3.2. Data Sets

The basic data sets of CGE models are Social Accounting Matrix (SAM). SAM is a database that simultaneously shows goods and factor flows among sectors in an economy. A hypothetical SAM was used in this study. The details of this SAM are presented in Table 1.

SAMs are square matrices in which economic agents are written in row and column headings. In this system, SAM rows show the income of the relevant economic agent and its columns show its expenditures or costs. In terms of macroeconomic equilibrium, each row total should be equal to the corresponding column total. This type of SAM also shows the initial equilibrium of the model.

SAMs have a great importance in obtaining the initial values of the model and in calculating the parameters from these initial values. This calculation method is called calibration.

For example, preference parameters for HOH_l are calibrated from equaiton (19) as follows;

$$C_{la} = \frac{\theta_{la}}{P_a^c} M_l^d \qquad \forall l, a \qquad \mathbf{(19)}$$

From equation (19):

$$\theta_{la} = \frac{C_{la}^0 P_a^{c0}}{M_l^d} \qquad \forall l, a \qquad \mathbf{(27)}$$

In equilibrium; *Total Household consumption = Disposable income*

$$M_l^d = C_{la}^0 P_a^{c0} + C_{ls}^0 P_s^{c0} + C_{li}^0 P_i^{c0} \qquad \textbf{(28)}$$

From equation (27) and (28);

$$\theta_{la} = \frac{C_{la}^0 P_a^{c0}}{C_{la}^0 P_a^{c0} + C_{ls}^0 P_s^{c0} + C_{li}^0 P_i^{c0}} \qquad \textbf{(29)}$$

Where, C_{la}^0 and P_a^{c0} are initial values of C_{la} and P_a^c. It can be seen from Table 1 that the $C_{la}^0 P_a^{c0}$ is equal to 20. In CGE models all prices are equaled to "1" (unit prices) in initial equilibrium for simplicity. Hence, the initial values of C_{la}, C_{ls} and C_{li} can obtained from SAM directly as 20, 11 and 11 respectively.

As a result $\theta_{la} = \frac{20*1}{20*1+11*1+11*1} \sim 0.4767.$

To obtain the SAM values again within the scope of a set of economic equations and equilibriums involving different assumptions by using the parameters obtained as a result of the calibration process is called as the base run equilibrium. In other words, while the SAM in Table 1 represents the initial equilibrium, to obtain the initial equilibrium values by using the equation systems in section (2.1) within the scope of the parameters obtained from this SAM is called as the base run equilibrium (Hosoe, Gasawa and Hashimoto 2010, 129).

Table 1: Social Accounting Matrix

	AGR	SERV	IND	LAB	CAP	IDTAX	GOV	INV	HOH1	HOH2	TOTAL
AGR	4	5	3				8	2	20	21	**63**
SERV	16	6	14				13	5	11	14	**79**
IND	10	4	10				18	20	11	29	**102**
LAB	15	32	13								**60**
CAP	13	22	50								**85**
IDTAX	5	10	12								**27**
GOV						27			6	12	**45**
SAV							6		7	14	**27**
HOH1				50	5						**55**
HOH2				10	80						**90**
TOTAL	**63**	**79**	**102**	**60**	**85**	**27**	**45**	**27**	**55**	**90**	

Source: Complied by Authors

At this point, the CGE model was established. From now on, model results can be obtained by changing the value of any parameter or exogenous variable (namely, by giving a shock to the economy) in line with the research purpose of the modeler. The equilibrium achieved after the shock is the counterfactual equilibrium solution.

4. Python General Structure and Model Algorithm

Python is a high-level, object-oriented, interpretative and modular programming language designed by Guido van Rossum. Python 1.0 released in

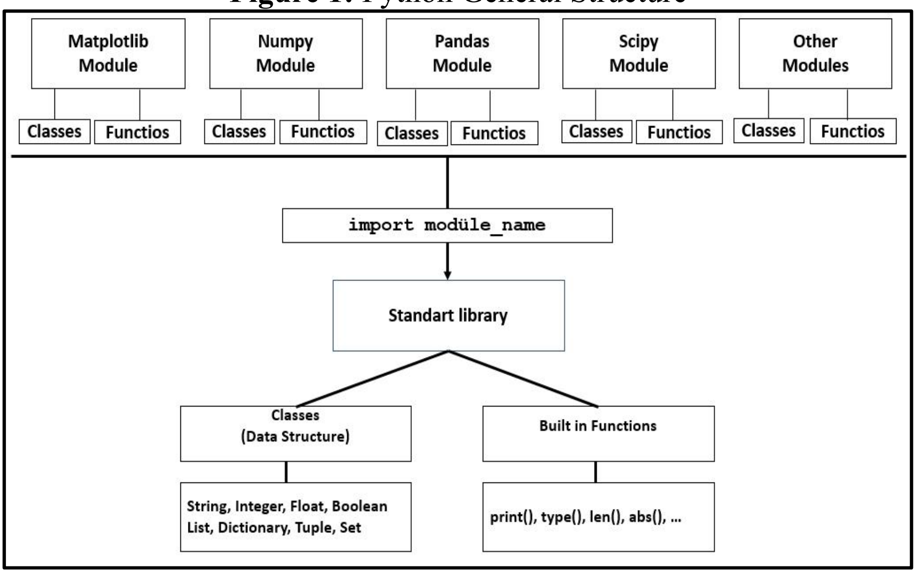
1994, and the latest version used in this study is Python 3.9. Nowadays, it is developed by the Python software foundation.

The success of the Python software language, which has gained great popularity all over the world especially in the last decade, is attributed to the following factors (John Hunt 2019,1): being flexible and easy to learn, providing a more standard programming language than some competitors such as R in the field of data science, its programming ability in almost any operating system, having many additional libraries (modules) that you can improve the general features of the language, and of course being open source.

4.1. Python General Structure

The general structure of the Python software language is presented in Figure 1 below:

Figure 1: Python General Structure



Source: Complied by Authors

Unlike its other features, two most important features of Python are that it is object-oriented and has a modular structure. In Python, the *modules* are structures consisting of *classes* and *functions*, and each *module* is generally written for different operations. For instance, while two and three dimensional graphs are created with *Matplotlib* module, *Pandas* module is very useful in data analysis. Nevertheless, many operations can be done with the standard library (*module*), which started to be used with the installation of Python, and other modules are integrated into the standard module when needed. The *import module_name* command is used for this operation. After the relevant *module* is imported, the *submodules*, *classes* and *functions* of this *module* can be used. Within the standard library, there are several data structures (classes) defined by python developers such as *String*, *List*, and *Dictionary*. The created data structure can be defined to a variable and two different operations can be performed on this variable. The first of them is passing the generated data as an argument to one of the built-in functions, user defined function or user defined classes, the second one is the use of methods belonging to the class in which the data is created.

Table 2: Python Basic Data Operation

```
1  var = ["CGE", "SAM", 5, 2.2]
2  print(var)
3
4  var.append("Economy")
```

For instance, in the 1st row of Table 2, a data was created from the *list class* and assigned to the variable named "*var*", this data was pass as an argument to the *print()* function in the 2nd row, and in the 4th row, the data named "*Economy*" was been added to the end of the list by using the *append* method of the *list class* in which the variable was created.

In Python, unlike built-in data structures (classes) and functions, users can also define class structure and function. Table 3 shows an example for it.

Table 3: User Defined Class and Function

```
 1  def Compute(x,y):
 2      return x + y
 3
 4  class CGE():
 5      def __init__(self, x, y):
 6          self.x = x
 7          self.y = y
 8      def compute(self):
 9          return self.x + self.y
10
11  Compute(5, 2)
12
13  model1 = CGE(5, 2)
14  model1.compute()
```

A function named *Compute* taking the parameters x and y was created in the first row. The return value of this function was defined as the sum of the x and y values that would be pass as arguments to the function with the return statement. In row 11, function named *Compute* was called and the values 5 and 2 were pass as arguments to this function.

In the fourth row, a class named *CGE ()* was created. The *__init__* method and the *self* parameter have a special place in class structures. The *__init__* method is an built-in structure in Python and is called a *constructor*. The *__init__* method works when an object (instance) is created from the classes, and the general features of the object are determined with the *self* parameter. In the fifth row, the __*init*__ method was defined and it was stated that the class would take two parameters called x and y. In the sixth and seventh rows, these parameters were defined for the object to be created. In the eighth row, a method belonging to this class was defined. The return value of this method is the sum of the x and y variables to be passed as arguments to the class. As it can be seen, the definition of the method is exactly the same as in functions. In Python, methods can be expressed as functions defined in a class. In the 13th row, an object is created by passing arguments 5 and 2 to the CGE () class and in the 14th row, the *compute* method of this class was called.

In this study, information about the modules used in addition to the standard module is as follows: **Matplotlib:** It is used in creating two and three dimensional graphics. **Numpy:** It is used in scientific calculations with arrays and matrices. **Pandas:** It is used for data analysis. **Scipy:** It is used in the fields of mathematics and engineering. It is especially useful for optimization problems. In this study, *Scipy* module was used to solve the CGE model.

4.2. Calibration Algorithm

Class structures were used in the creation of the CGE model in Python. The class named *CGE()* will represent the general structure of the model, and various objects (instance) to be created from this class will represent the counterfactual equilibriums obtained after the shock given to the model parameter or exogenous variables.

Accordingly, when the codes in Appendix 1 were examined, the required modules for the model were imported between the rows 1 - 5 . In the seventh row, the model class named *CGE()* and right after the constructor, *init_ (self)*, method belonging to this class were created. According to the __*init*__ method, the *CGE()* class takes a parameter named SAM. This parameter represents the social accounting matrix in our model. The SAM parameter will be passed as an argument to the class *CGE()* class when calling. Therefore, it should be created just before the *CGE()* class is called. This process was performed in the rows between 283 - 285 of Appendix 2. Here, a variable named "data" was created, and the SAM prepared in excel format was introduced to Python using the *read_excel()* function of the Pandas module. This excel format is shown in Appendix 3. In Appendix 2, in the row 284, the excel table was converted into a data frame and assigned to a variable named SAM. In the row 285, the column named "index" of the created Data Frame was changed as the index value of the data frame.

When the variable named SAM is passed as an argument to the *CGE()* class, the operations to be performed in the *init* method are shown after the 10th row in Appendix 1. First, in the 11th row, a new variable named SAM was defined and assigned to the value of `self.SAM.lock`. Here, "*self*" refers to the parameter of the *__init__* method, and SAM refers to the SAM argument. "*lock*" is a method belonging to the *Pandas* class. When row and column values are passed as arguments to this "*lock"* method, it gives the intersection value of the relevant row and column.

In the 13th and 14th rows, some label of SAM, "AGR", "SERV", and "IND" variables assigned to "a", "s", "i" values, respectively, and "HOH1", "HOH2" variables were assigned to "l" and "k" values, respectively, in order to make the indexing process easier. The initial values are calculated between the 17th - 74th rows. Between the 17th and 21st rows, all prices in the model were equaled to 1, and in the next rows, the initial values of the model were obtained by the `self.SAM.lock` method, which was created previously and assigned to the SAM variable. For instance, the "`La = SAM["LAB", a]`" command in the 22nd row assigns the intersection value of "LAB" row and "a" ("AGR") column in the social accounting matrix to the "`La`" variable. Between the 77th and 121st rows, the model parameters were calibrated by using the initial values.

4.3. Model Solution Algorithm

As it was stated previously, the *Scipy* module will be used for the solution of the CGE model. In this study, the CGE model was considered as a nonlinear optimization problem. Therefore, the *minimize* function within the *optimize* module in the *Scipy* was chosen as the solution method. The *minimize* function basically takes 4 parameters(https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html).

minimize(fun, x0, bounds = (), constraints = ())

Among these parameters, ***fun*** refers to the objective function which will be minimized, ***x0***; refers to the initial values of the model variables, ***bounds***; refers to the lower and upper limits of initial values, ***constraints*** refers to model constraints. With regard to the CGE model, ***fun:*** refers to the household total welfare ($max\, UU$), ***x0:*** refers to the initial values of endogenous (model) variables obtained from SAM, ***bounds:*** (>0, None) refers to the condition that the lower limit is greater than zero for each endogenous variables, ***constraints:*** refers to CGE model equations (1 - 23 equations).

There is no separate *maximize* function in the *scipy.optimize* module unlike the *minimize* function. The basic logic here is to solve the problem by placing the (-) sign in front of the function if the objective function is to be maximized. However, in CGE models, model constraints are important rather than the objective function. Defining a function different from household utility levels as an objective function will not change the model solution.

Objective function named *objValue* was defined as a method of the *CGE()* class between the 123rd - 132nd rows in Appendix 1. The value of the *objValue* method was defined as the total welfare of HOH1 and HOH2 with the *return* statement.

Between the rows 134 and 250, the model constraints, namely the CGE model equations, were defined with the method named *constraints*. This method takes a parameter named "x". This parameter represents an array in which the endogenous variables of the model. Since there are 57 endogenous variables in the model, excluding the wage level (w) determined as numeraire, the parameter x is an array consisting of 57 pieces of data. Between the rows 135 – 191, each element of the array was assigned to the variables compatible with the CGE model by indexing method. The value of the *constraints* method was defined as a list, each element of which would create a model equation. For instance, the code block 194 - 217 represents the system of equations between 1 - 6 in section 2.1, namely the production block. The basic logic in these definitions is that all equations are equal to zero when the model is solved. For example, when the model is solved, the value of the endogenous variable "Ya" defined in row 194 should be equal to the right side of the equation.

The solution of the model is performed in the method named *SolveModel* of the *CGE()* class. In the row 253, the model constraints are defined to the variable named *cons* in dictionary format. The statement *self.constraints* in this row refers to the model constraint equations in *constraints* method defined in row 134. In the row 254, an array consisting of 57 pieces of value "1" was assigned to the x0 variable with the "*ones"* method in the *numpy* module. This array constitutes the initial values of the model. In the row 255, the lower and upper limits of the endogenous variables of the model were assigned to the variable named *bnds* with the list comprehension method. Between the rows 257 – 261, the model solution was assigned to the variable named *result* by using the *minimize* function. The standard return value of the *minimize* function are presented in Appendix 4. The return value of this function is in dictionary format, and the *x* variable of this dictionary contains model solution values according to the order defined in the *constraints* method between the rows 135 – 191. In the row 261, the statement *result.x* provides the return of the "x" value of the *result* variable, namely the array value containing the endogenous variables of the model.

5. Simulations

In CGE models, exogenous variables and model parameters can be shocked. Therefore, it means that 47 different shocks can be given to the model within the scope of 2 exogenous variables ($\bar{L}, \bar{K}$) and 45 parameters. However, 3 different shocks will be given here and the simulation results will be visualized.

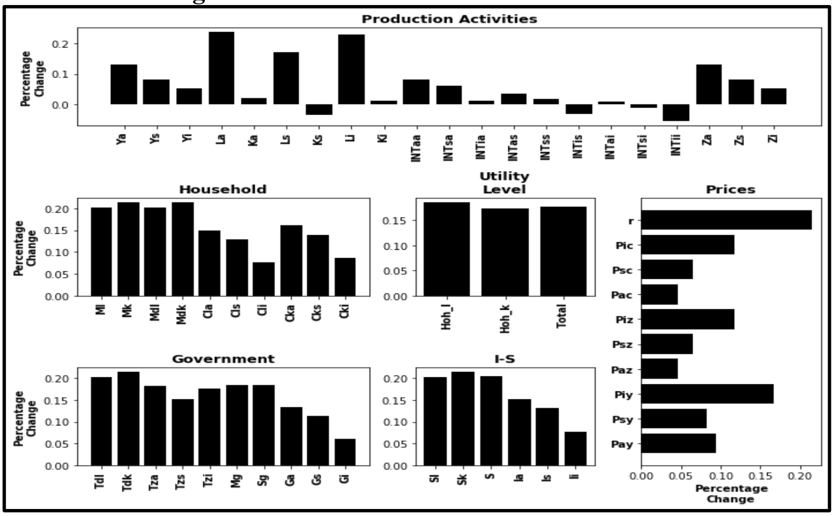
5.1. Simulation 1: Labor Shocks

Within the scope of the model, firstly, a positive shock will be given to the labor supply and the simulation results will be analyzed. As it was stated previously, each object to be generated from the CGE class represents different states of equilibrium. In this context, the base run equilibrium will be first derived from the CGE class, and then, the counterfactual equilibrium will be derived and the changes in the model variables between the two equilibrium states will be monitored.

Python codes required for the simulation are shown in appendix 2. The names of the endogenous variables of the model are shown as a list between the rows 275 - 281. The SAM parameter, which is necessary to call the *CGE()* class was produced as a data frame object between the rows 283 - 285. In row 287, an object named *m_baserun* was derived from the *CGE()* class. The model was solved immediately after using the *SolveModel* method of the CGE () class. This object represents the base run equilibrium of the model. In row 290, a second object named *m_cfrun* was derived, and in row 291, the **L** (labor) variable of this object was increased by 20%. The model is also solved in line 292. The *m_cfrun* object now represents the counterfactual equilibrium.

Since the value of endogenous (model) variables in the base run equilibrium and counterfactual equilibrium will be grouped in a data frame, an empty data frame named *df_variables* was created in row 294, and the index values were assigned as the values of the *x1* list we created previously. In the 295th row, a column named "Base" was added to the *df_variables*, and the return value of the *SolveModel()* method of the *m_baserun* object produced previously was assigned to the values of this column. The value of the *SolveModel()* method of the *CGE()* class was equaled to the variable *result.x* with the return statement. The *result.x* variable returns the solution of the model. In the 296th row, the same operations were performed for the object *m_cfrun*, which represents the counterfactual equilibrium. In the 297th row, the percentage change values in these two equilibrium solution values were calculated and assigned to the column named "diff". The "diff" column of the *df_variables* was fragmented between the 300th and the 304th rows to show the different activities in the economy. For instance, the *df_pro* variable in the 300th row shows the change values in the endogenous variables that show the production activities in the economy. *df_gov* shows the changes related to the government activities. Likewise a data frame was created between the rows 306 - 311 to show the welfare of two different households and the total welfare in the economy. After the 313th row, the changes in the economy were visualized using the *matplotlib* module. The results of the first simulation are presented in Figure 2.

Figure 2: Simulation Result of Labour Shocks



Source: Simulation Results

Figure 2 consists of six different sub-figures, each of which is a bar graph. Each sub-figure represents a different segment of the economy. For instance, the top shows the percentage changes in the model variables representing the production activities in the economy. When the simulation results were examined, except for the INT_is and INT_ii variables, all model variables were increased. In other words, it can be said that a labor shock to be applied positively in this economy will positively affect all model variables, including relative prices. Furthermore, the positive labor shock shows that both households and the total utility level in the economy increased. However, EV or CV calculations are needed for a more systematic measure of the welfare level in the economy, which are not included in this study. However, other modelers will be able to make these calculations with a few small additions to the existing codes.

Increasing the labor supply in the economy primarily reduced the price of labor, namely, the wage level relatively. As you will remember, when the CGE model was established, the wage level was chosen as numeraire and its value was fixed to 1. Prices sub-figure shows the change in the level of other prices in the economy relative to wage. The relative decrease in the wage level increased the labor demand of the sectors. When the Production Activities sub-figure is examined, it can be observed that the highest percentage changes are in the variables La, Ls and Li. The increase in labor supply increased the labor income (*Ml*) in the economy, and the relative increase in the price of capital also increased the capital income (*Mk*). The increase in income increased the demand for the goods of the sectors. Similarly, the increase in income and production as a result of increased labor supply increased the tax revenues of the government and therefore the government expenditures. The same results can also be observed in the investment - savings segment.

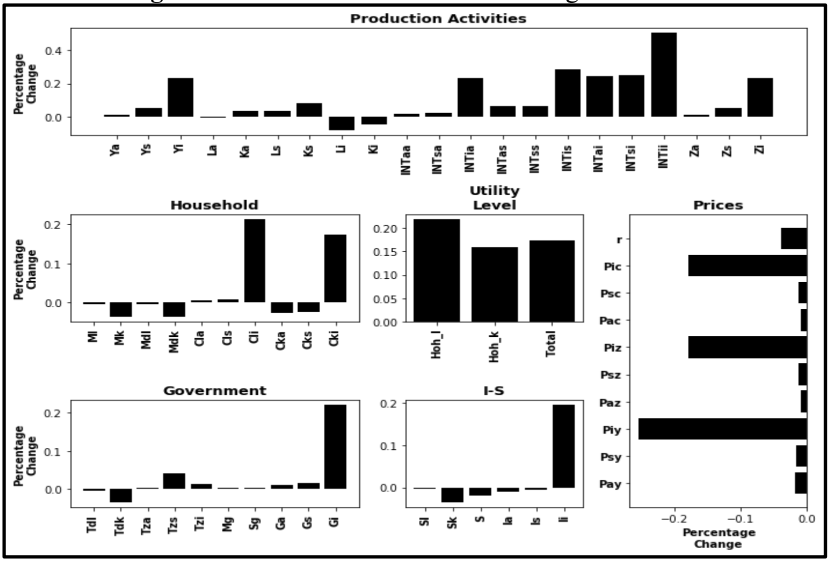
5.2. Simulation 2: Industry Sector Technological Shocks

Second, it will be assumed that there is a technological development in the industry sector. It will be sufficient to change the 291st row in the current code block as follows in order to obtain simulation results;

```
m_cfrun.b_i = m_cfrun.b_i * 1.30
```

With this code line, the b_i parameter in the equation (1), which represents the composite factor production of the sector i, was increased by 30%. The simulation results are presented in Figure 3.

Figure 3: Simulation Result of Technological Shocks



Source: Simulation Results

Although all prices in the economy decrease relatively, the decreases in composite goods (*Piy*), producer (*Piz*) and consumer (*Pic*) prices of the industry sector are more than others. Although the decreases in prices also increase the demand for the goods of each sector, the positive change in the demand for final goods of the industry sector is more than the others.

When it is considered in terms of production activities, the positive technology shock in the industry sector positively affected the value of all variables except for labor and capital factors used in this sector. Increasing technology level enabled to obtain more output with less production factors in the industry sector, which can be observed with the percentage increase in the *Zi* variable. Furthermore, since the increasing industry sector production will increase the intermediate inputs used by this sector, the production of other sectors is also positively affected. Moreover, both the decrease in the relative price of the capital and the decrease in the amount of capital used in the industrial sector reduced the capital income of the household, which caused a decrease in the direct tax revenues of the government (Tdl and Tdk) and an increase in the production tax revenues due to the increase in the production of the sectors. Finally, the positive technology shock in the industrial sector increased the utility level of the household section.

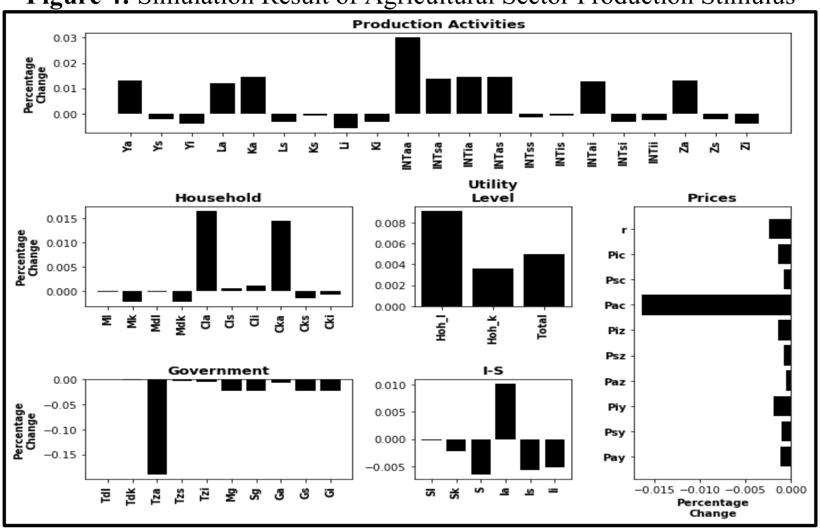
5.3. Simulation 3: Agricultural Sector Production Stimulus

Finally, the change in the economic structure as a result of the government's reduction of taxes in the agricultural sector to support agricultural production will be simulated. It will be sufficient to change the 291st row in the current code block as follows in order to obtain these simulation results;

```
m_cfrun.tz_a = m_cfrun.tz_a / 1.25
```

This code line reduced the production tax on the agricultural sector by 25%. The simulation results are presented in Figure 4.

Figure 4: Simulation Result of Agricultural Sector Production Stimulus



Source: Simulation Results

First of all, the reduction of the production tax on the agricultural sector relatively decreased consumer (demand) prices in the agricultural sector, which increased the consumption of agricultural goods by the government, investment agent and household. Therefore, as can be seen in the production block, the labor and capital factors used in the agricultural sector and intermediate inputs and output increased more than other variables. Finally, the reduction of tax rates on the agricultural sector also positively affected the utility of the household.

Conclusions

Programs and software to be used in economic analysis are important. While programming languages such as MATLAP, Fortran and C are generally used in this field, Python has been recently widely used in the economic literature. In computable general equilibrium (CGE) analyses, programs specific to nonlinear optimization problems such as GAMS or GEMPACK are mostly used. Although they are very successful in optimization problems and numerical calculations, the main disadvantage of these types of programming languages emerges at the point of diversifying the results achieved and integrating them into different fields. The main source of motivation of this study is the development of CGE algorithms with the Python software language, which is an application area in many fields, especially scientific studies around the world. It is considered that CGE modelers' use of Python software language will increase the dynamism of the studies in this field.

Nevertheless, the CGE model in this study was created in a flexible structure according to minimum standards. For instance, the Cobb Douglass (C-D) utility function was used in modeling the household and firm behaviors. However, constant elasticity of substitution (CES) utility and production functions can be used in case of analysis within the scope of different substitution elasticities in both consumption and production processes. In this case, the calibration processes of the C-D function between the 77-121st rows in Appendix 1 should be excluded and the relevant calibration processes of the CES function should be included. On the other hand, the objective function defined in row 123 should be rearranged within the scope of CES and the production processes in the constraint equations should be updated. likewise, for simplicity, the exogenous sector was not included in the model, and within the scope of CES and CET (constant elasticity of transformation), import demand and export supply functions, respectively, will be easily included in the model.

References

Arndt, C., Hertel, T., Demaranan, B., Huff, K., & McDougall, R. (1997). China in 2005: Implications for the rest of the world. *Journal of Economic Integration, 12*(4), 505–547.

Arndt, C. (2002). *HIV/AIDS and macroeconomic prospects for Mozambique: An initial assessment* (TMD Discussion Paper No. 88). International Food Policy Research Institute.

Arrow, K. J., & Debreu, G. (1954). Existence of an equilibrium for a competitive economy. *Econometrica, 22*(1), 265–290. https://doi.org/10.2307/1907353

Benjamin, N., Devarajan, S., & Weiner, R. J. (1989). The "Dutch" disease in a developing country: Oil reserves in Cameroon. *Journal of Development Economics, 30*(1), 71–92. https://doi.org/10.1016/0304-3878(89)90051-5

Block, P., Strzepek, K., Rosegrant, M., & Diao, X. (2006). *Impacts of considering climate variability on investment decisions in Ethiopia* (EPT Discussion Paper No. 150). International Food Policy Research Institute.

Bourguignon, F., Robilliard, A.-S., & Robinson, S. (2003). *Representative versus real households in the macro-economic modeling of inequality* (DELTA Working Paper 2003-05). Ecole Normale Supérieure, Département et Laboratoire d'Économie Théorique et Appliquée.

Burfisher, M. E., Robinson, S., & Thierfelder, K. (1994). Wage changes in a U.S.-Mexico free trade area: Migration versus Stolper-Samuelson effects. In J. Francois & C. Shiells (Eds.), *Modelling trade policy: Applied general equilibrium assessments of North American Free Trade* (pp. xx–xx). Cambridge University Press.

Burfisher, M. E. (2012). *Introduction to computable general equilibrium models*. Cambridge University Press.

Cardenete, M. A., Guerra, A.-I., & Sancho, F. (2012). *Applied general equilibrium*. Springer.

Cattaneo, A., Hinojosa Ojeda, R. A., & Robinson, S. (1999). Costa Rica trade liberalization, fiscal imbalances, and macroeconomic policy: A computable general equilibrium model. *The North American Journal of Economics and Finance, 10*(1), 39–67. https://doi.org/10.1016/S1062-9408(99)00018-2

Devarajan, S., Thierfelder, K. E., & Suthiwart-Narueput, S. (2001). The marginal cost of public funds in developing countries. In A. Fossati & W. Wiegard (Eds.), *Policy evaluations with computable general equilibrium models* (pp. xx–xx). Routledge.

Dewatripont, C., Ehrlich, S., Ginsburg, V., & Regemorter, V. (1991). The effects on unemployment of reducing social security contributions: A general equilibrium analysis for Belgium. In H. Don, T. van de Klundert, & J. van Sinderen (Eds.), *Applied general equilibrium modelling* (pp. 135–155). Springer.

Dewatripont, M., & Michel, G. (1987). On closure rules, homogeneity and dynamics in applied general equilibrium models. *Journal of Development Economics, 26*(1), 65–76. https://doi.org/10.1016/0304-3878(87)90052-6

Diao, X., Doukkali, R., & Yu, B. (2008). *Policy options and their potential effects on Moroccan small farmers and the poor facing increased world food prices: A general equilibrium model analysis* (IFPRI Discussion Paper No. 00813). International Food Policy Research Institute.

Diao, X. (2009). *Economywide impact on avian flu in Ghana: A dynamic CGE model analysis* (IFPRI Discussion Paper No. 00866). International Food Policy Research Institute.

Dixon, P., Rimmer, M., & Tsigas, M. (2007). Regionalizing results from a detailed CGE model: Macro, industry and state effects in the United States of removing major tariffs and quotas. *Papers in Regional Science, 86*(1), 31–55. https://doi.org/10.1111/j.1435-5957.2006.00101.x

Elliot, J., Foster, I., Kortum, S., Munson, T., Perez Cervantes, F., & Weisbach, D. (2010). Trade and carbon taxes. *American Economic Review, 100*(2), 465–469. https://doi.org/10.1257/aer.100.2.465

Gehlhar, M. (1997). Historical analysis of growth and trade patterns in the Pacific Rim: An evaluation of the GTAP framework. In T. Hertel (Ed.), *Global trade analysis: Modelling and applications* (pp. xx–xx). Cambridge University Press.

Goulder, L. H., & Eichengreen, B. (1989). *Trade liberalization in general equilibrium: Intertemporal and interindustry effects* (NBER Working Paper No. 2965). National Bureau of Economic Research.

Hosoe, N., Gasawa, K., & Hashimoto, H. (2010). *Textbook of computable general equilibrium programming and simulations*. Palgrave Macmillan.

Huff, K., & Hertel, T. W. (2000). *Decomposing welfare changes in the GTAP model* (GTAP Technical Paper No. 5). Purdue University, Center for Global Trade Analysis.

Hunt, J. (2019). *Advanced guide to Python 3*. Springer.

Johansen, L. (1960). *A multi-sectoral study of economic growth*. North-Holland Publishing.

Jokisch, S., & Kotlikoff, L. J. (2005). *Simulating the dynamic macroeconomic and microeconomic effects of the FairTax* (NBER Working Paper No. 11858). National Bureau of Economic Research.

Kuiper, M., & van Tongeren, F. (2006). *Using gravity to move Armington* (GTAP Resource No. 2044). Purdue University, Center for Global Trade Analysis.

Nielson, C. P., Thierfelder, K., & Robinson, S. (2001). *Genetically modified foods, trade and developing countries* (TMD Discussion Paper No. 77). International Food Policy Research Institute.

Scarf, H. (1969). An example of an algorithm for calculating general equilibrium prices. *American Economic Review, 59*(4), 669–677.

SciPy.org. (2021). *scipy.optimize.minimize — SciPy v1.6.2 reference guide*. Retrieved June 21, 2021, from https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html

Shoven, J. B., & Whalley, J. (1973). General equilibrium with taxes: A computational procedure and an existence proof. *The Review of Economic Studies, 40*(4), 475–489. https://doi.org/10.2307/2296582

Shoven, J. B., & Whalley, J. (1977). Equal yield tax alternatives: General equilibrium computational techniques. *Journal of Public Economics, 8*(2), 211–224. https://doi.org/10.1016/0047-2727(77)90019-6

Verma, M., & Hertel, T. W. (2009). *Commodity price volatility and nutrition vulnerability* (Markets, Trade and Institutions Division Discussion Paper No. 00895). International Food Policy Research Institute.

World Bank. (2006). *Global economic prospects: Economic implications of remittances and migration*. World Bank.

Appendix – 1: CGE Model Algorithm with Class Structure

```
from scipy.optimize import minimize
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gr

class CGE():

    def __init__(self, SAM):
        self.SAM = SAM
        SAM = self.SAM.loc

        a, s, i = "AGR","SERV","IND"
        l, k    = "HOH1","HOH2"

        #Initial Values
        r    = 1
        w    = 1
        Paz, Psz, Piz = 1,1,1
        Pay, Psy, Piy = 1,1,1
        Pac, Psc, Pic = 1,1,1
        La      = SAM["LAB", a]
        Ka      = SAM["CAP", a]
        Ls      = SAM["LAB", s]
        Ks      = SAM["CAP", s]
        Li      = SAM["LAB", i]
        Ki      = SAM["CAP", i]
        Ya      = La + Ka
        Ys      = Ls + Ks
        Yi      = Li + Ki
        INTaa   = SAM[a,a]
        INTsa   = SAM[s,a]
        INTia   = SAM[i,a]
        INTas   = SAM[a,s]
        INTss   = SAM[s,s]
        INTis   = SAM[i,s]
        INTai   = SAM[a,i]
        INTsi   = SAM[s,i]
        INTii   = SAM[i,i]
        Za      = Ya + INTaa + INTsa + INTia
        Zs      = Ys + INTas + INTss + INTis
        Zi      = Yi + INTai + INTsi + INTii
        self.L = La + Ls + Li
        self.K = Ka + Ks + Ki
        Hlw     = SAM[l, "LAB"]  # Wage income of HOH1
        Hlr     = SAM[l, "CAP"]  # Capital income of HOH1
        Hkw     = SAM[k, "LAB"]  # Wage income of HOH2
        Hkr     = SAM[k, "CAP"]  # Capital income of HOH2
        Cla     = SAM[a, l]
        Cls     = SAM[s, l]
        Cli     = SAM[i, l]
        Cka     = SAM[a, k]
        Cks     = SAM[s, k]
        Cki     = SAM[i, k]
        Sl      = SAM["SAV", l]
        Sk      = SAM["SAV", k]
        Sg      = SAM["SAV", "GOV"]
        S       = Sl + Sk + Sg
        Ia      = SAM[a,"INV"]
        Is      = SAM[s,"INV"]
        Ii      = SAM[i,"INV"]
        Ga      = SAM[a,"GOV"]
        Gs      = SAM[s,"GOV"]
        Gi      = SAM[i,"GOV"]
        Tdl     = SAM["GOV", l]
        Tdk     = SAM["GOV", k]
        Tza     = SAM["IDTAX", a]
        Tzs     = SAM["IDTAX", s]
        Tzi     = SAM["IDTAX", i]
        Mg      = (Tdl + Tdk + Tza + Tzs + Tzi)
        Ml      = Hlw + Hlr
        Mk      = Hkw + Hkr
        Mdl     = Ml - Sl - Tdl
        Mdk     = Mk - Sk - Tdk
        #CALIBRTATION
        #Parameters
        self.theta_la  = Cla / (Cla + Cls + Cli)
        self.theta_ls  = Cls / (Cla + Cls + Cli)
        self.theta_li  = Cli / (Cla + Cls + Cli)
        self.theta_ka  = Cka / (Cka + Cks + Cki)
        self.theta_ks  = Cks / (Cka + Cks + Cki)
```

```
        self.theta_ki  = Cki / (Cka + Cks + Cki)
        self.lambda_a  = Ia / (Ia + Is + Ii)
        self.lambda_s  = Is / (Ia + Is + Ii)
        self.lambda_i  = Ii / (Ia + Is + Ii)
        self.mu_a      = Ga / (Ga + Gs + Gi)
        self.mu_s      = Gs / (Ga + Gs + Gi)
        self.mu_i      = Gi / (Ga + Gs + Gi)
        self.l_w       = Hlw / self.L
        self.l_r       = Hlr / self.K
        self.k_w       = Hkw / self.L
        self.k_r       = Hkr / self.K
        self.mps_l     = Sl / Ml
        self.mps_k     = Sk / Mk
        self.mps_g     = Sg / Mg
        self.tz_a      = Tza / Za
        self.tz_s      = Tzs / Zs
        self.tz_i      = Tzi / Zi
        self.ty_l      = Tdl / Ml
        self.ty_k      = Tdk / Mk
        self.alpha_a   = La / (La + Ka)
        self.alpha_s   = Ls / (Ls + Ks)
        self.alpha_i   = Li / (Li + Ki)
        self.beta_a    = Ka / (La + Ka)
        self.beta_s    = Ks / (Ls + Ks)
        self.beta_i    = Ki / (Li + Ki)
        self.b_a       = Ya / (La**self.alpha_a * Ka**self.beta_a)
        self.b_s       = Ys / (Ls**self.alpha_s * Ks**self.beta_s)
        self.b_i       = Yi / (Li**self.alpha_i * Ki**self.beta_i)
        self.x_aa      = INTaa / Za
        self.x_sa      = INTsa / Za
        self.x_ia      = INTia / Za
        self.x_as      = INTas / Zs
        self.x_ss      = INTss / Zs
        self.x_is      = INTis / Zs
        self.x_ai      = INTai / Zi
        self.x_si      = INTsi / Zi
        self.x_ii      = INTii / Zi
        self.ya        = Ya / Za
        self.ys        = Ys / Zs
        self.yi        = Yi / Zi

    def objValue(self, x):
        Cla    = x[41]
        Cls    = x[42]
        Cli    = x[43]
        Cka    = x[44]
        Cks    = x[45]
        Cki    = x[46]
        Util_l = Cla**self.theta_la*Cls**self.theta_ls*Cli*self.theta_li
        Util_k = Cka**self.theta_ka*Cks**self.theta_ks*Cki*self.theta_ki
        return Util_l + Util_k

    def constraints(self, x):
        Ya    = x[0]
        Ys    = x[1]
        Yi    = x[2]
        La    = x[3]
        Ka    = x[4]
        Ls    = x[5]
        Ks    = x[6]
        Li    = x[7]
        Ki    = x[8]
        INTaa = x[9]
        INTsa = x[10]
        INTia = x[11]
        INTas = x[12]
        INTss = x[13]
        INTis = x[14]
        INTai = x[15]
        INTsi = x[16]
        INTii = x[17]
        Za    = x[18]
        Zs    = x[19]
        Zi    = x[20]
        Tdl   = x[21]
        Tdk   = x[22]
        Tza   = x[23]
        Tzs   = x[24]
        Tzi   = x[25]
        Mg    = x[26]
        Sg    = x[27]
        Ga    = x[28]
        Gs    = x[29]
```

```
        Gi    = x[30]
        Sl    = x[31]
        Sk    = x[32]
        S     = x[33]
        Ia    = x[34]
        Is    = x[35]
        Ii    = x[36]
        Ml    = x[37]
        Mk    = x[38]
        Mdl   = x[39]
        Mdk   = x[40]
        Cla   = x[41]
        Cls   = x[42]
        Cli   = x[43]
        Cka   = x[44]
        Cks   = x[45]
        Cki   = x[46]
        Pay   = x[47]
        Psy   = x[48]
        Piy   = x[49]
        Paz   = x[50]
        Psz   = x[51]
        Piz   = x[52]
        Pac   = x[53]
        Psc   = x[54]
        Pic   = x[55]
        r     = x[56]
        w     = 1       #Numeraire

        return [Ya       - self.b_a*La**self.alpha_a * Ka**self.beta_a,
                Ys       - self.b_s*Ls**self.alpha_s * Ks**self.beta_s,
                Yi       - self.b_i*Li**self.alpha_i * Ki**self.beta_i,
                La       - (self.alpha_a / w)* Pay * Ya,
                Ka       - (self.beta_a / r)* Pay * Ya,
                Ls       - (self.alpha_s / w)* Psy * Ys,
                Ks       - (self.beta_s / r)* Psy * Ys,
                Li       - (self.alpha_i / w) * Piy * Yi,
                Ki       - (self.beta_i / r) * Piy * Yi,
                INTaa    - (self.x_aa / Pac) * Za,
                INTsa    - (self.x_sa / Psc) * Za,
                INTia    - (self.x_ia / Pic) * Za,
                INTas    - (self.x_as / Pac) * Zs,
                INTss    - (self.x_ss / Psc) * Zs,
                INTis    - (self.x_is / Pic) * Zs,
                INTai    - (self.x_ai / Pac) * Zi,
                INTsi    - (self.x_si / Psc) * Zi,
                INTii    - (self.x_ii / Pic) * Zi,
                Ya       - (Za / self.ya),
                Ys       - (Zs / self.ys),
                Yi       - (Zi / self.yi),
                Paz * Za - (Ya*Pay + INTaa*Pac + INTsa*Psc + INTia*Pic),
                Psz * Zs - (Ys*Psy + INTas*Pac + INTss*Psc + INTis*Pic),
                Piz * Zi - (Yi*Piy + INTai*Pac + INTsi*Psc + INTii*Pic),
                Tdl      - (self.ty_l * Ml),
                Tdk      - (self.ty_k * Mk),
                Tza      - (self.tz_a * Paz * Za),
                Tzs      - (self.tz_s * Psz * Zs),
                Tzi      - (self.tz_i * Piz * Zi),
                Mg       - (Tdl + Tdk + Tza + Tzs + Tzi),
                Sg       - (self.mps_g * Mg),
                Ga       - (self.mu_a / Pac) * Mg*(1-self.mps_g),
                Gs       - (self.mu_s / Psc) * Mg*(1-self.mps_g),
                Gi       - (self.mu_i / Pic) * Mg*(1-self.mps_g),
                Sl       - self.mps_l * Ml,
                Sk       - self.mps_k * Mk,
                S        - (Sl + Sk + Sg),
                Ia       - (self.lambda_a / Pac) * S,
                Is       - (self.lambda_s / Psc) * S,
                Ii       - (self.lambda_i / Pic) * S,
                Ml       - (w*self.l_w*self.L + r*self.l_r*self.K),
                Mk       - (w*self.k_w*self.L + r*self.k_r*self.K),
                Mdl      - Ml*(1- self.mps_l - self.ty_l),
                Mdk      - Mk*(1- self.mps_k - self.ty_k),
                Cla      - (self.theta_la / Pac) * Mdl,
                Cls      - (self.theta_ls / Psc) * Mdl,
                Cli      - (self.theta_li / Pic) * Mdl,
                Cka      - (self.theta_ka / Pac) * Mdk,
                Cks      - (self.theta_ks / Psc) * Mdk,
                Cki      - (self.theta_ki / Pic) * Mdk,
                Za       - (Cla + Cka + Ga + Ia + INTaa + INTas + INTai),
                Zs       - (Cls + Cks + Gs + Is + INTsa + INTss + INTsi),
                self.L   - (La + Ls + Li),
                self.K   - (Ka + Ks + Ki),
```

```
                 Pac      - Paz * (1 + self.tz_a),
                 Psc      - Psz * (1 + self.tz_s),
                 Pic      - Piz * (1 + self.tz_i)]

    def SolveModel(self):
        cons = {"type":"eq", "fun":self.constraints}
        x0 = np.ones(57)
        bnds = [(0.001, None) for k in range(57)]

        result = minimize(self.obj_value,
                          x0,
                          bounds = bnds,
                          constraints = cons)
        return result.x

    def utility_level(self):
        x = self.SolveModel()
        Cla = x[41]
        Cls = x[42]
        Cli = x[43]
        Cka = x[44]
        Cks = x[45]
        Cki = x[46]
        Util_l = Cla**self.theta_la * Cls**self.theta_ls * Cli * self.theta_li
        Util_k = Cka**self.theta_ka * Cks**self.theta_ks * Cki * self.theta_ki
        return [Util_l , Util_k, Util_l + Util_k]
```

Appendix – 2: Simulation and Visualization of CGE Model Results

```
x1 = ["Ya", "Ys", "Yi", "La", "Ka", "Ls", "Ks", "Li", "Ki",
      "INTaa", "INTsa", "INTia","INTas", "INTss", "INTis",
      "INTai", "INTsi", "INTii","Za", "Zs", "Zi", "Tdl", "Tdk",
      "Tza", "Tzs", "Tzi","Mg", "Sg","Ga", "Gs", "Gi","Sl", "Sk", "S",
      "Ia", "Is", "Ii", "Ml", "Mk", "Mdl", "Mdk","Cla", "Cls", "Cli",
      "Cka", "Cks", "Cki","Pay", "Psy", "Piy","Paz", "Psz", "Piz","Pac",
      "Psc", "Pic","r"]

data = pd.read_excel("Data_int.xlsx")
SAM = pd.DataFrame(data)
SAM.set_index("index", inplace = True)

m_baserun = CGE(SAM)
m_baserun.SolveModel()

m_cfrun= CGE(SAM)
m_cfrun.L = m_cfrun.L * 1.2
m_cfrun.SolveModel()

df_variables = pd.DataFrame(index = x1)
df_variables["Base"] = m_baserun.SolveModel()
df_variables["CF-1"] = m_cfrun.SolveModel()
df_variables["Diff"] = ((df_variables["CF-1"] - df_variables["Base"]) /\
                         df_variables["Base"])

df_pro = df_variables["Diff"][0:21]
df_gov = df_variables["Diff"][21:31]
df_inv = df_variables["Diff"][31:37]
df_hh  = df_variables["Diff"][37:47]
df_price = df_variables["Diff"][47:]

df_utility = pd.DataFrame(index = ["Hoh_l","Hoh_k", "Total" ])
df_utility["Base"]  = m_baserun.utility_level()
df_utility["CF-1"]  = m_cfrun.utility_level()
df_utility["Diff"]  = (df_utility["CF-1"] - df_utility["Base"]) /\
                       df_utility["Base"]
df_util = df_utility["Diff"]

fig = plt.figure(figsize = (10,8))
gs = gr.GridSpec(3,7)
ax1 = plt.subplot(gs[0,:])
ax2 = plt.subplot(gs[1,:3])
ax3 = plt.subplot(gs[1:,5:])
ax4 = plt.subplot(gs[2,:3])
ax5 = plt.subplot(gs[2,3:5])
ax6 = plt.subplot(gs[1,3:5])

x1 = np.arange(len(df_pro))
y1 = df_pro.values
ax1.bar(x1, y1, color = "green")
ax1.set_ylabel("Percentage\nChange", fontsize = 7, fontweight = "bold")
```

```python
ax1.set_title("Production Activities", fontsize = 10, fontweight = "bold")
ax1.set_xticks(x1)
ax1.set_xticklabels(df_pro.index, rotation = "vertical",
                    fontsize = 8, fontweight = "bold")

x2 = np.arange(len(df_hh))
y2 = df_hh.values
ax2.bar(x2, y2, color = "orange")
ax2.set_ylabel(r"$\delta$ %", fontweight = "bold")
ax2.set_title("Household", fontsize = 10, fontweight = "bold")
ax2.set_xticks(x2)
ax2.set_xticklabels(df_hh.index, rotation = "vertical",
                    fontsize = 8, fontweight = "bold")

x3 = np.arange(len(df_price))
y3 = df_price.values
ax3.barh(x3, y3, color = "blue")
ax3.set_xlabel(r"$\delta$ %",  fontweight = "bold")
ax3.set_title("Prices", fontsize = 10, fontweight = "bold")
ax3.set_yticks(x3)
ax3.set_yticklabels(df_price.index, fontsize = 8, fontweight = "bold")

x4 = np.arange(len(df_gov))
y4 = df_gov.values
ax4.bar(x4, y4, color = "black")
ax4.set_ylabel(r"$\delta$ %",  fontweight = "bold")
ax4.set_title("Government", fontsize = 10, fontweight = "bold")
ax4.set_xticks(x4)
ax4.set_xticklabels(df_gov.index, rotation = "vertical",
                    fontsize = 8, fontweight = "bold")

x5 = np.arange(len(df_inv))
y5 = df_inv.values
ax5.bar(x5, y5, color = "pink")
ax5.set_title("IS", fontsize = 10, fontweight = "bold" )
ax5.set_xticks(x5)
ax5.set_xticklabels(df_inv.index, rotation = "vertical",
                    fontsize = 8, fontweight = "bold")

x6 = np.arange(len(df_util))
y6 = df_util.values
ax6.bar(x6, y6, color = "red")
ax6.set_title("Utility\nLevel", fontsize = 10, fontweight = "bold" )
ax6.set_xticks(x6)
ax6.set_xticklabels(df_util.index, rotation = "vertical",
                    fontsize = 8, fontweight = "bold")
plt.tight_layout()
```

Appendix – 3: Excel Format of the Social Accounting Matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	index	**AGR**	**SERV**	**IND**	**LAB**	**CAP**	**IDTAX**	**GOV**	**INV**	**HOH1**	**HOH2**	**TOTAL**	**DIFF**
2	**AGR**	4	5	3	-	-	-	8	2	20	21	**63**	0
3	**SERV**	16	6	14	-	-	-	13	5	11	14	**79**	0
4	**IND**	10	4	10	-	-	-	18	20	11	29	**102**	0
5	**LAB**	15	32	13	-	-	-	-	-	-	-	**60**	0
6	**CAP**	13	22	50	-	-	-	-	-	-	-	**85**	0
7	**IDTAX**	5	10	12	-	-	-	-	-	-	-	**27**	0
8	**GOV**	-	-	-	-	-	27	-	-	6	12	**45**	0
9	**SAV**	-	-	-	-	-	-	6	-	7	14	**27**	0
10	**HOH1**	-	-	-	50	5	-	-	-	-	-	**55**	0
11	**HOH2**	-	-	-	10	80	-	-	-	-	-	**90**	0
12	**TOTAL**	**63**	**79**	**102**	**60**	**85**	**27**	**45**	**27**	**55**	**90**	-	-

Appendix – 4: Return Value of *minimize* Function

```
     fun: 71.39529339676403
     jac: array([0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.47602558, 0.50176048, 1.89640808, 0.89932823,
       0.94794655, 2.07083416, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        ])
 message: 'Optimization terminated successfully.'
    nfev: 535
     nit: 9
    njev: 9
  status: 0
 success: True
       x: array([28.        , 54.        , 63.        , 15.        , 13.        ,
       32.        , 22.        , 13.        , 50.        ,  3.68253968,
       13.97468354,  8.82352941,  4.6031746 ,  5.24050633,  3.52941176,
        2.76190476, 12.2278481 ,  8.82352941, 58.        , 69.        ,
       90.        ,  6.        , 12.        ,  5.        , 10.        ,
       12.        , 45.        ,  6.        ,  7.36507937, 11.35443038,
       15.88235294,  7.        , 14.        , 27.        ,  1.84126984,
        4.36708861, 17.64705882, 55.        , 90.        , 42.        ,
       64.        , 18.41269841,  9.60759494,  9.70588235, 19.33333333,
       12.2278481 , 25.58823529,  1.        ,  1.        ,  1.        ,
        1.        ,  1.        ,  1.        ,  1.0862069 ,  1.14492754,
        1.13333333,  1.        ])
```