Statistisk sentralbyrå
Statistics Norway

Microsimulation of multi-regional population projections with immigrant characteristics

Model documentation and validation

TALL SOM FORTELLER

NOTATER / DOCUMENTS

2025/40

James Jia, Zhiyang Jia, Stefan Leknes, Sturla A. Løkken and Michael J. Thomas

In the series Documents, documentation, method descriptions, model descriptions and standards are published.

Published: 5. December, 2025

ISBN 978-82-587-2056-7 (electronic)
ISSN 2535-7271 (electronic)

Symbols in tables	Symbol
Category not applicable Figures do not exist at this time, because the category was not in use when the figures were collected.	.
Not available Figures have not been entered into our databases or are too unreliable to be published.	..
Confidential Figures are not published to avoid identifying persons or companies.	:
Decimal punctuation mark	.

Preface

Producing population projections for official statistics is one of Statistics Norway's core tasks. Since 2019, considerable effort has been devoted to modernizing the projection models, including experimentation with new model classes and the development of novel methods for deriving demographic assumptions used in model calibration. This document reviews one step in that development: the creation, validation and proof-of-concept demonstration of a new microsimulation projection model that incorporates both regional and immigrant characteristics, effectively combining the BEFREG and BEFINN frameworks. The work has been financed by Statistics Norway and the Research Council of Norway (Grant 352911).

Statistics Norway, 3 December 2025

Linda Nøstbakken

Abstract

This document introduces a new microsimulation model for population projections that integrates the national and regional cohort-component models, BEFINN and BEFREG, thereby ensuring consistency with established projection systems. The model extends these frameworks by incorporating additional heterogeneity through both immigrant and regional characteristics, as well as their interactions. Capturing such heterogeneity represents a key advantage of microsimulation over traditional cohort-component approaches.

Validation against the benchmark projections shows that the new model reproduces results with high precision at both aggregate and disaggregated levels. Differences between the microsimulation and benchmark models are negligible across population size and demographic components, including for immigrant groups and at municipality levels. These findings demonstrate that the microsimulation model is robust, reliable, and well-suited for future applications in population analysis.

In addition to the validation exercise, we conduct a proof-of-concept demonstration that offers a first glimpse into the analytical potential of the extended model. We highlight emerging patterns, such as differences in internal migration, that warrant deeper investigation. These preliminary insights point to promising directions for explorations in future publications.

Table of contents

1. Introduction

The macro-class cohort-component model is the workhorse of population projections, widely used by scholars, statistical agencies, and international organizations (Burch, 2018). While it is a powerful tool for planning and policy, its main limitation lies in its restricted ability to accommodate additional population heterogeneity (Van Imhoff and Post, 1998, Zagheni, 2015). This limitation stems from the model's structure: it relies on matrices with cells representing every combination of population characteristics, so the complexity grows exponentially as more characteristics are added. A microsimulation approach addresses this issue by shifting the unit of analysis to individuals, whose characteristics can be stored and retrieved more efficiently.

Statistics Norway has traditionally relied on the macro-class cohort-component approach. To produce results at both the regional level and by immigrant categories, the institution has maintained two separate frameworks: the BEFREG model for regional projections and the BEFINN model for projections by immigrant background. The transition to a microsimulation model class opens up the possibility of combining these two approaches into a single framework that incorporates both spatial and immigrant heterogeneity.

A microsimulation approach addresses the limitations of the macro-class framework by shifting the unit of analysis from cells to individuals, whose characteristics can be stored and retrieved more efficiently. The added heterogeneity enhances the realism of the framework and broadens the range of questions that can be addressed. In addition, a key advantage of microsimulation is that it naturally provides measures of model uncertainty, since demographic events are represented as stochastic processes conditional on individual characteristics.

There are several types of microsimulation models. For some recent reviews of the literature, see for instance Li and O'Donoghue (2013), Lomax and Smith (2017) and Zagheni (2015). In this work, we aim to remain as close as possible to the structure of the standard macro-class cohort-component model by developing a discrete-time dynamic microsimulation model. In discrete-time models, demographic events are simulated at fixed time intervals (e.g., annually), whereas in continuous-time models events can occur at any point in time, governed by transition intensities. Dynamic microsimulation models follow individuals over time, allowing their characteristics to evolve according to specified transition probabilities. This approach makes it possible to capture life-course dynamics, cohort effects, and long-term population changes.[1] The model is explicitly stochastic: life events for each person are simulated using Monte Carlo methods. Demographic events such as births, deaths, and migrations are modeled at the micro level as random processes affecting individuals, as opposed to deterministic aggregate calculations.

This document describes the BEFINN and BEFREG models, and outlines the details related to the construction of a new combined microsimulation model. It has been pointed out by several scholars that there is a lack of model documentation and validation in the microsimulation literature (Li and

[1] Static microsimulation models, by contrast, rely primarily on reweighting mechanisms and are less directly connected to individual-level behavioral processes.

O'Donoghue, 2013, Morrison, 2008). Following best-practice recommendations, we validate the new model against the established benchmark models. This approach not only enhances transparency but also helps identify potential coding or data errors. We also provide a brief presentation of selected results from the extended model, while a more comprehensive exploration of richer results will be presented in a companion report.

2. Official projection models at Statistics Norway

In the following, we describe the two population projection models currently used by Statistics Norway. We present the main features of the National model (BEFINN) which produces national-level projections disaggregated by immigrant background characteristics, and the Regional model (BEFREG) which produces regional-level projections disaggregated by municipalities. We review the underlying demographic assumptions dictating the results of both models, and describe the caveats and exceptions related to the particular implementation of the BEFINN and BEFREG models.

In this report, we take as our point of departure the beginning-of-year population in 2022 and corresponding model assumptions described in Leknes and Løkken (2022) and Thomas and Tømmerås (2022).

2.1 National population projection model – BEFINN

The national model produces projections of the Norwegian population up to the year 2100, with particular emphasis on the composition and contribution of persons with immigrant background. In addition to age (0-119 years) and sex (male and female), the model incorporates three further characteristics related to immigrant background: country group, immigrant generation and duration of stay. Further description of these variables follows.

Country group is based of country of origin, defined as country of birth and not the country they are emigrating from. Four groups of country backgrounds are included. The country groups are based on a combination of distance to Norway and observed immigration patterns to Norway.

- *CG1 ('Western countries')*: Europe (excluding CG2 and non-EU member states in Estern Europe), North America, Australia, and New Zealand
- *CG2 ('Eastern EU')*: Estonia, Latvia, Lithuania, Poland, Czechia, Slovakia, Hungary, Romania, Bulgaria, Slovenia, and Croatia
- *CG3 ('Rest of the World')*: Africa, South and Central America and the Caribbean, Asia, Oceania (excluding Australia and New Zealand), and all non-EU member states in Eastern Europe
- *CG4 ('Natives')*: Those born in Norway from Norwegian parents

Finer details on the categorization is found in Appendix A.

Immigrant generation. Individuals are classified into three groups according to their immigration experiences and heritage:

- Foreign-born individuals who were born to two foreign-born parents and four foreign-born grandparents
- Norwegian-born individuals born in Norway with two foreign-born parents and four foreign-born grandparents
- All others

Duration of stay. For immigrants, duration of stay is measured as the number of complete years since the first registered immigration event. As a rule of thumb, for newborn immigrants duration of stay will equal their age, while for older immigrants it is calculated as age minus age at immigration. Duration of stay is an annual count variable represented by 120 categories.[2]

The model characteristics are outlined in the upper panel of Table 2.1. Counting all interactions between variables provides a model with total matrix size of 27,302,400 cells.

Throughout this text, we will use the term *immigrant background* to refer to any combination of country background, immigrant generation and duration of stay. More detail on the model and assumptions can be found in Thomas and Tømmerås (2022).

National model assumptions

The population projection model relies on explicit demographic assumptions concerning fertility, mortality, immigration, and emigration. These assumptions are not forecasts in a strict sense, but conditional scenarios describing how the population would develop if the underlying assumptions hold.

For fertility, assumptions specify age-specific fertility rates by immigrant background. They are derived from observed historical patterns in Statistics Norway's register data, combined with expert assessments and, where relevant, international comparative evidence.

Mortality assumptions are given as age- and sex-specific death rates, with gradual improvements over time. The mortality assumptions are modeled using the Lee-Carter framework, which extrapolates past mortality trends while allowing for continued, but decelerating, life expectancy gains.

Migration assumptions are particularly detailed. Immigration is projected by country group, reflecting observed inflows, policy expectations, and external forecasts. Emigration probabilities depend on both country background and duration of stay, and are estimated from register data on past emigration behavior. Together, these assumptions form the basis for the transition probabilities used in the projection model.

2.2 Regional population projection model – BEFREG

The multi-regional model produces projections of the Norwegian population in each municipality up to the year 2050. It's characteristics include age (0-119 years), sex (male and female), and municipality of residence (357 municipalities). Additional detail can be found in Leknes and Løkken (2022).

[2] Results are often reported for aggregated categories of duration of stay, and demographic assumptions are identical for durations of 13 years or more.

Table 2.1 Model characteristics and cell count

	Minimum	Maximum	Unique values	Combinations
National model characteristics				
Year	2022	2100	79	
Age	0	119	120	
Sex	1	2	2	
Country background	1	4	4	
Immigrant generation	1	3	3	
Duration of stay	0	119	120	
Total*				**27,302,400**
Regional model characteristics				
Year	2022	2050	29	
Age	0	119	120	
Sex	1	2	2	
Municipality id	301	5444	356	
Total				**2,477,760**
Combined model characteristics				
Baseline combinations (sex, age, year)				18,960
Regional combinations (municipality)				356
National combinations (immigrant background)				1440
Total*				**9,719,654,400**

Notes: The table shows characteristics and number of combinations in the three models: regional, national and combined model. * For the total calculations all interactions are included. In practice, demographic behavior/assumptions will not be differ across immigrant characteristics for natives.

The regional model characteristics are outlined in the middle panel of Table 2.1. Counting all interactions between variables provides a model with total matrix size of 2,477,760 cells.

Municipalities. The regional model uses the municipality structure at the start of the base year as its point of reference. This means that the data used to derive demographic rates and probabilities must be geographically harmonized over time. The harmonization is carried out with the aid of GIS data and software (geographic information systems). In 2022, Norway had 356 municipalities, a notable reduction from previous years; before 2020, the number was 428.

Regional model assumptions

The regional projection model builds on the same demographic components as the national model—fertility, mortality, immigration, and emigration—but requires additional assumptions on internal migration between municipalities and regions.

Fertility assumptions are specified as age-specific fertility rates (ASFRs) for women aged 15–49 by municipality. The rates are estimated using the most recent three years of data prior to the base year and smoothed with an empirical Bayes (EB) approach, as described in Leknes and Løkken (2021). To ensure consistency, municipal-level ASFRs are adjusted to reproduce the national ASFR in the final observation year, after which trends follow the national assumptions.

Mortality assumptions are derived in a similar way. Age- and sex-specific death rates are estimated

for each municipality using the EB method. These local rates are then scaled to be consistent with the national mortality trajectories generated by the Lee-Carter model, thereby ensuring coherence between the regional and national levels.

Immigration assumptions follow the national projections of total inflows aggregated across country groups. At the municipal level, immigrants are allocated proportionally according to each municipality's historical share of new immigrants over a reference period. This approach captures observed settlement patterns while maintaining consistency with the national totals.

Emigration assumptions are also anchored in the national numbers but are operationalized at the municipal level as emigration rates by age, sex, and municipality of residence. These rates are estimated from historical register data and then rescaled so that the aggregate aligns with the national emigration assumptions.

Internal migration is modeled through a two-step procedure. First, out-migration propensities are estimated by age, sex, and municipality of residence, capturing the likelihood that an individual leaves their current municipality. Second, out-migrants are redistributed to destination municipalities using a transition (or "moving") matrix derived from observed historical inter-municipal migration flows. This approach preserves both the overall level of internal mobility and the observed directional patterns in migration streams.

2.3 Finetuning of models to serve as validation comparisons

Ideally, the models would follow precisely the setup and assumptions described in Section 2. In practice, however, the implementation of the BEFINN and BEFREG models may deviate slightly due to coding legacies, computational simplifications, coding errors, or data issues. Such discrepancies are typically minor and inconsequential for the main results and conclusions, making them difficult to detect.

For the purpose of validating the new microsimulation model, it is essential to replicate the results produced by the official models. This requires accounting for every modeling choice and demographic assumption, no matter how small. The value of this exercise lies not only in establishing a robust basis for validation, but also in helping to identify and eliminate errors. A more detailed discussion of the errors uncovered is provided in Section E.

For valid comparisons, we use variants of the microsimulation model that replicate the minor errors present in the traditional BEFINN and BEFREG results. However, when comparing the combined model with the national and regional models, we rely on corrected versions where such errors have been resolved.

3. Microsimulation implementation

The projection is done via a dynamic spatial discrete-time microsimulation model which is implemented with Python. The model follows each individual in the population, year by year, applying probabilities of giving birth, dying, moving, or out-migrating. Immigrants are chosen at random from a pool of potential

individuals with given probabilities. This produces a realistic future population where the aggregate outcomes reflect both randomness and demographic assumptions built into the model.

As can be seen from Table 2.1, the combined model would have a prohibitive size when including all characteristics. Total matrix size would in such a case be close to 1.4 billions. This is the exact reason why such a model needs to draw on the microsimulation framework to be feasible.

Python codes that implement the projection can be found at on github (`https://github.com/statisticsnorway/forsk-folksim`). We give a quick summary of the implementation in the following. Detailed description can be found in the Appendix.

3.1 Framework details

3.1.1 Individual and Population

In our current model, the unit of simulation in individual. Household formation and dissolution are not modeled. The population is represented by a collection of individuals. The population at the end of time period $t-1$ serves as the baseline population for time period t.

3.1.2 Life events

The occurrence of a life event at any given discrete time is determined stochastically by transition probabilities that, in principle, depend on all individual characteristics. These probabilities are imported into the model through assumption files; see Appendix C for details.

The life events modeled are:

Aging: Deterministic for all individual present at the end of the year
Mortality: Based on age- and sex-specific mortality rates for each region
Birth: Based on age-specific fertility rates for the females in each region
Domestic migration:

- *Out-migration:* Based on age- and sex-specific out-migration rates for each region.
- *In-migration:* The destination probabilities of the domestic migrants are assigned depending on age, sex and region of departure (moving matrix).

International migration:

- *Emigration:* Based on age- and sex-specific emigration rates for each region.
- *Immigration:* The destination probabilities of the immigrants are assigned depending on age and sex.

3.1.3 Timing of events

During time period t, we simulate life events for all individuals in the stock population at the start of that period. The sequence of the events are the following:

1. Update age, time index and duration of stay (only for immigrants). Increasing the age and time index by one effectively updates the end-of-year population from period $t-1$ to start-of-year population

for period t (age defined as end-of-year age). This is the stock population of year t. (Note that the time index of the first year of projection is set to 0, as python array indexes start with 0.)
2. Simulate the fertility events of all women and add newborns to the same region as their mothers.
3. Calculate the adjustment factor for emigration, so that the number of out-migrants are the same, in expectation, as the model assumptions.
4. Death, emigration and internal out-migration are simulated simultaneously for the individuals in the baseline population, including the newborns.
5. Simulate the new immigrants for time period t, and add them to the population. Note that immigrants will not die, out-migrate or move internally in the year of arrival.
6. Assign region destinations for immigrants and domestic movers according age, sex and origin.
7. Remove individuals that die or emigrate from the population.

The resulting population is the end-of-year population of period t. The projection is further extended by one period by repeating this procedure for the next period $(t + 1)$.

3.2 Computational Strategies

As the microsimulation model is run several times to provide different realizations of the population development, it is relatively time consuming compared to the deterministic macro-level cohort-component model. It is favorable to minimize run time. In the following, we describe computational strategies used for this purpose.

3.2.1 Parallel Computation

To reduce the running time, we have used the multiprocessing package in python to enable parallel implementation.

We have set a niceness of 5 for the simulation processes and given them slightly lower priority than the default. This allows the simulations to run effectively without slowing down other jobs on the server. In addition, we pin these processes to given cores on our linux server. Pinning multiprocessing workers to specific cores provides several important benefits. It helps maintain cache locality, so a worker's data stays in the local memory caches and avoids performance loss from core migration. Pinning also cut down context switching, as fewer migrations implies lowers TLB flushes and less kernel overhead. On the two-socket server (`stata_p4`) where we run the simulations, this is especially useful: it keeps the processes on the same socket, which avoids a lot of memory latency. Even though other users could still share those cores, pinning is a strong hint to the operation system to prioritize keeping our processes on those dedicated cores.

3.2.2 Random number generation

Each simulation process is assigned its own independent random number generator (`rng`). We initialize a single `SeedSequence` from a master seed, then use `.spawn()` to derive child seeds for each simulation. This ensures that runs are **reproducible** (the same master seed always yields the same results) while also

providing **statistically independent** random streams across processes.

Using explicit `rng` objects instead of the global `np.random` has several advantages:

- **Reproducibility**: simulations can be repeated exactly by reusing the same master seed.
- **Isolation**: each process has its own random stream, avoiding accidental correlations.
- **Clarity**: all randomness is explicit and tied to a specific `Population` object, making the code easier to understand and debug.
- **Flexibility**: different parts of the projection can be assigned separate generators if needed.

3.2.3 Memory Efficiency and Management

A central challenge in large-scale microsimulation is the processing and storage of the massive amounts of data generated by each simulation. Although end-of-year population files are saved, our primary interest often lies in aggregated summary statistics—for example, the number of primary school–age children in Oslo in 2040. To retain the granularity required for detailed post-simulation analysis, it is useful to generate multidimensional arrays indexed by `sim_index`(N), `year`(T), `age` (120), `sex` (2), `region` (356), `imm_group` (4), `gen` (3), `duration_stay` (120), and `event_type` (7), where the numbers in parentheses indicate the dimension (cardinality) of each index.

With 1,000 simulations over 30 years, even when the `duration_stay` variable is restricted to only 14 groups (0 through 12, and 13 or more), it still results in

$$1000 \times 30 \times 120 \times 2 \times 356 \times 4 \times 3 \times 14 \times 7 = 3\,014\,323\,200\,000$$

total entries (cells). If each entry is stored as a 16-bit integer (`Int16`), the required memory would be about $5{,}615$ GB (≈ 5.49 TB) — far too large to handle directly.

This highlights the need for more efficient strategies for storing and processing simulation outputs. We compute and aggregate relevant statistics on the fly, thereby avoiding the creation of full high-dimensional arrays. In other words, we split the big data array into smaller chunks and save them to disk as they are generated and then free up memory. This reduces the peak use of memory dramatically. In addition, we use compression and sparse formats, since many cells contain zeros.

Recall that we make use of multiprocessing where each worker runs independent simulations. Once one year's projection is done, they append results to a summary *Zarr* array with dimensions

$$(\texttt{sim\_index}, \texttt{year}, \texttt{age}, \texttt{sex}, \texttt{region}, \texttt{imm\_group}, \texttt{gen}, \texttt{duration\_stay}, \texttt{event\_type}).$$

The arrays are chunked along `(sim_index, year)` and compressed with `Blosc/Zstd`, ensuring both parallel writes and manageable storage overhead. Each chunk takes around 1.3-1.4 MB. The full set of 1000 runs with 30 years fits in roughly 25 GB on disk after the compression.

We also run a lightweight monitor that keeps a dictionary of completed runs and prints live updates on progress. Once a given number of simulations complete, we use `Dask` and `xarray` to wrap the `Zarr`

arrays and aggregate results across simulation indices. Aggregation is performed in blocks of years, and we computes partial sums and writes them in-place to an aggregate `Zarr` group on disk. This design allows incremental updates: newly finished simulations are picked up, aggregated, and recorded without reprocessing earlier runs. In addition, this gives live updates (rolling aggregates) so analysts can track convergence without waiting for the full run. These *partially* aggregated per–event type outputs (e.g. population, births, deaths, migrants, movers) are then written to Stata `.dta` files. These outputs are already cleaned (zeros dropped, years shifted, generations reindexed) and typed down to compact integer representations for efficiency.

In the end, the pipeline produces:

1. A **summary Zarr store** with the full simulated arrays (compressed).
2. Two **Stata data files** for each combination of simulation and year: one stores the projected end-of-year population, and the other stores the subpopulation that experienced at least one life event within that year.
3. An **Zarr group store** contains the total number of aggregated simulations and the summed counts stored in multidimensional arrays.
4. A set of **Stata data files**, one per event type, containing marginal summaries by year, age, sex, region, and other dimensions.
5. **Progress logs and diagnostics** from each worker pool, showing runtime, aggregation steps, and successful exports.
6. **Readme file** is generated when simulations are finished. The file contains basic information about the run and summary results. In addition, copies of the Python files used are stored in the results folder.

This workflow provides scalability through chunked `Zarr` storage, parallelism via multiprocessing and `Dask`, continuous monitoring of progress, and streamlined exports for downstream statistical analysis.

4. Validation of the microsimulation model

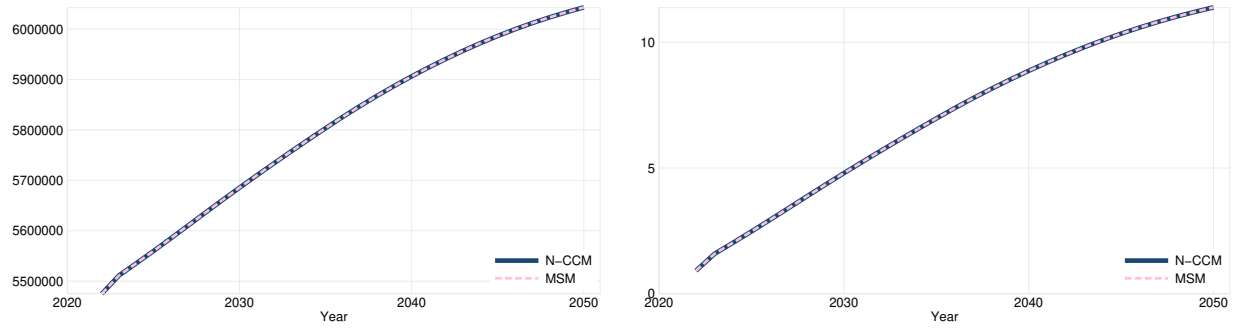
Population projections have a long legacy at Statistics Norway, and new results are expected at regular intervals by central government agencies, local administrations, and a wide range of public and private sector actors for planning and evaluation purposes. Because the projections play such a central role in decision-making, continuity and transparency in the modeling process are essential.

Any changes to the projection models must therefore be justified by clear improvements in the quality or relevance of the results. New approaches need to demonstrate added value compared to the established benchmark models, and their consequences must be carefully documented. In practice, this requires analyzing and explaining the differences in outcomes, so that the results from a new model remain predictable, interpretable, and trustworthy for users. Ensuring transparency and continuity also safeguards the long-term credibility of population projections. By documenting methodological choices and openly assessing their impact, Statistics Norway strengthens user confidence in both the projections

themselves and the models on which they are based.

In this report, we propose two major changes to the way Statistics Norway produces population projections. First, we introduce an new methodology by replacing the macro-class cohort-component models with a microsimulation model. Second, we combine the assumptions of the two existing cohort-component models: the regional model (BEFREG) and the national model (BEFINN). Validation exercises are therefore pertinent.

In this section, we demonstrate that the new microsimulation model is able to reproduce the results of both existing cohort-component models. This is achieved by holding the assumptions fixed to those of either BEFINN or BEFREG and running the microsimulation model accordingly—that is, by temporarily ignoring some of the additional heterogeneity available in the new framework. In the following section, we then show how the results change when the assumptions of both models are combined and the full range of heterogeneity in the microsimulation model is exploited. This stepwise validation is essential for ensuring transparency and continuity, as it allows users to clearly see how the new methodology relates to the established benchmark models and to build confidence in the reliability of the new results.



(a) Population size **(b)** Population growth

Figure 4.1 Comparison of projected population size between BEFINN and the MS model

Notes: The figure compares the results from the National projections and the combined microsimulation model. Panel (a) displays the comparison for population size, whereas panel (b) shows the comparison for population growth in percentage of the 2022 population.

4.1 Replication of the BEFINN results

We begin by comparing results from the macro-class cohort-component model (BEFINN) with the average outcomes from 1,000 runs of the microsimulation model. As a first step, we examine projected population size. Figure 4.1 displays overall population levels and growth, showing that the trajectories of the two models overlap almost perfectly, with no discernible differences at the aggregate level. This impression is confirmed by the numerical results in Table 4.1. By 2050, the accumulated difference amounts to only 25 persons, corresponding to a percentage discrepancy of 0.0046.

Differences in births, deaths, immigrations and emigrations up to 2050 are reported in Table 4.2. As expected, the discrepancies between the two models tend to remain very small. By 2050, the average number of births and deaths in the microsimulation model is 3.9 lower than in BEFINN, corresponding to deviations of just 0.0069 percent. The corresponding differences for immigrations and emigrations are -0.34 and -2 events, respectively. These discrepancies are negligible in practice and confirm that

Table 4.1 Comparing population results between BEFINN and the MS model

Year	Cohort-component model	Microsimulation model	Difference	
			Numerical	Percentage
2022	5,474,994	5,474,990	4	.00007
2024	5,535,393	5,535,397	-3.6	-.00007
2026	5,584,780	5,584,779	1.6	.00003
2028	5,635,495	5,635,473	22	.00039
2030	5,685,519	5,685,493	26	.00045
2032	5,733,851	5,733,841	10	.00018
2034	5,780,691	5,780,661	31	.00053
2036	5,825,936	5,825,928	8.1	.00014
2038	5,867,933	5,867,932	1.1	.00002
2040	5,906,312	5,906,297	15	.00025
2042	5,941,141	5,941,146	-5	-.00008
2044	5,972,320	5,972,347	-27	-.00045
2046	5,999,704	5,999,718	-14	-.00024
2048	6,023,261	6,023,231	31	.00051
2050	6,043,153	6,043,125	28	.00046

Notes: The table shows results from the national cohort-component model (CCM), BEFINN, and the average result from the microsimulation model (MS) after 1000 simulations. Results and differences are shown for aggregate projected population. Differences are calculated as the results from the traditional model minus the results from the microsimulation model.

the microsimulation model successfully replicates the aggregate results of the established BEFINN framework.

We next turn to population projections by immigrant background (see Table 4.3). The results show that the microsimulation model reproduces the cohort-component model with high precision across all groups. For natives, the difference in aggregate population size in 2050 is less than 0.0001 percent. Errors are somewhat larger for the immigrant groups, which are smaller in size. The differences in 2050 amount to -0.011 percent for Western immigrants, 0.002 percent for Eastern Europeans, and 0.0046 percent for the residual group (Rest of the World). Minor deviations occur in individual years, but the overall trajectories align closely, and accumulated differences by 2050 remain negligible in both absolute and relative terms.

Comparisons of demographic components likewise reveal only very small discrepancies. In 2050, the absolute differences in deaths range from 5.4 among natives to -2.7 among those in the residual immigrant category. For births, the differences are of similar magnitude, ranging from 6.1 (natives) to -1.4 (residual group). Immigration differences are minimal, while emigration differences range from 2.3 to -5.4. Taken together, these results demonstrate that the microsimulation model reproduces the outcomes of the BEFINN model with high precision, even when disaggregated by country background.

Table 4.2 Comparing aggregate component results between BEFINN and the MS model

Year	Deaths				Births			
	CCM	MS	Diff.	Percentage	CCM	MS	Diff.	Percentage
2022	42,214	42,212	1.8	.004	54,687	54,687	-.51	-.0009
2024	42,494	42,492	1.6	.004	54,635	54,640	-4.3	-.0078
2026	43,112	43,103	8.8	.020	55,391	55,396	-4.7	-.0084
2028	44,057	44,058	-1.2	-.003	56,319	56,318	.19	.0003
2030	45,279	45,283	-4.3	-.010	57,034	57,025	9.6	.0169
2032	46,710	46,705	4.9	.010	58,006	58,007	-1.4	-.0024
2034	48,274	48,271	3.1	.006	59,015	59,014	.75	.0013
2036	49,869	49,873	-3.6	-.007	60,219	60,234	-15	-.0252
2038	51,380	51,383	-3.6	-.007	60,063	60,065	-1.6	-.0027
2040	52,712	52,703	8.6	.016	59,786	59,778	8.3	.0139
2042	53,823	53,825	-1.9	-.004	59,256	59,255	1.1	.0019
2044	54,743	54,729	14	.025	58,461	58,472	-11	-.0188
2046	55,554	55,551	2.4	.004	57,491	57,489	2.2	.0039
2048	56,350	56,367	-16	-.029	56,515	56,506	9.5	.0168
2050	57,191	57,187	3.9	.007	55,666	55,663	3.9	.0069
Year	Immigrations				Emigrations			
	CCM	MS	Diff.	Percentage	CCM	MS	Diff.	Percentage
2022	67,126	67,126	-.18	-.0003	29,875	29,881	-6.5	-.022
2024	43,309	43,309	.13	.0003	31,490	31,494	-4.2	-.013
2026	42,832	42,832	-.20	-.0005	30,142	30,137	5.0	.017
2028	42,463	42,462	.69	.0016	29,258	29,263	-4.7	-.016
2030	41,920	41,921	-1.4	-.0032	28,862	28,854	7.9	.027
2032	41,386	41,385	.53	.0013	28,688	28,685	2.3	.008
2034	40,860	40,859	1.2	.0030	28,407	28,412	-5.6	-.020
2036	40,390	40,389	.98	.0024	28,284	28,280	3.8	.014
2038	39,979	39,979	-.024	-.0001	28,135	28,127	8.0	.029
2040	39,634	39,634	-.45	-.0011	27,962	27,958	4.2	.015
2042	39,307	39,306	.56	.0014	27,774	27,772	2.5	.009
2044	38,978	38,979	-.74	-.0019	27,572	27,570	1.9	.007
2046	38,627	38,627	-.48	-.0013	27,352	27,350	1.8	.007
2048	38,254	38,255	-.61	-.0016	27,111	27,115	-4.7	-.017
2050	37,876	37,876	-.34	-.0009	26,836	26,838	-2.0	-.007

Notes: The table shows results from the national cohort-component model (CCM), BEFINN, and the average result from the microsimulation model (MS) after 1000 simulations. Results and differences are shown for aggregate projected births, deaths, immigrations and emigrations. Differences are calculated as the results from the traditional model minus the results from the microsimulation model.

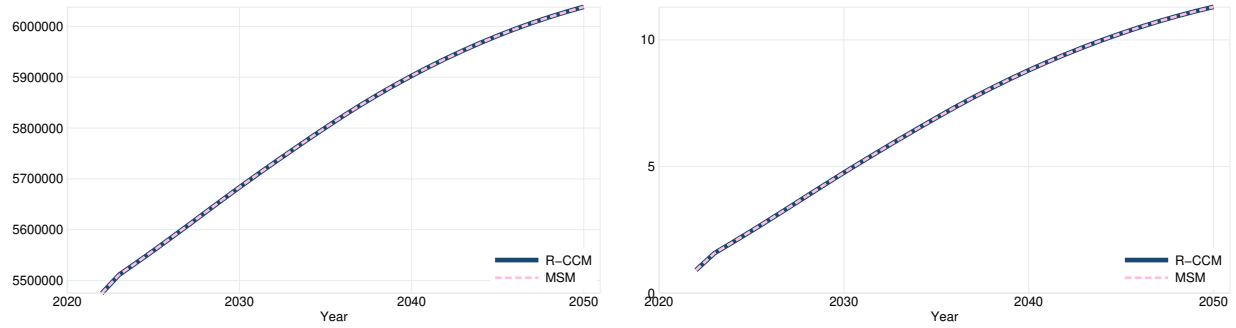
Table 4.3 Comparing aggregate results between BEFINN and the MS model, by country background

	Population				Deaths		Births		Immig.		Emig.	
	CCM	MS	Diff.	%	CCM	Diff.	CCM	Diff.	CCM	Diff.	CCM	Diff.
Country group 4 (Natives):												
2022	4404898	4404899	-0.3	-.00000	39392	0.1	39645	-1.2	5603	0.8	6059	0.9
2029	4441841	4441858	-17	-.00038	41175	-8.2	41207	-9.6	5709	1.1	5927	-2.1
2036	4480192	4480187	5.2	.00012	45355	-2.3	43204	-7.2	5816	-1	5803	-1.1
2043	4503436	4503435	0.7	.00001	48334	6.5	40867	9.8	5923	1.5	5596	-2.1
2050	4502986	4502981	4.5	.00010	49253	5.4	37754	6.1	6029	-1.8	5536	2.3
Country group 1 (Western):												
2022	185689	185690	-1.1	-.0044	1216	0.7	2259	-.28	10277	0.3	7918	0.6
2029	199382	199375	6.5	.0083	1322	-0.1	2245	-1.3	9666	-0.0	7612	-2.1
2036	209152	209156	-3.7	-.0052	1504	0.1	2401	-1.7	9059	0.6	7350	5.6
2043	216056	216067	-11	-.0016	1655	-1	2392	0.7	8682	-0.2	7086	2.7
2050	220851	220867	-16	-.0106	1833	2.3	2313	-0.7	8315	0.6	6820	2
Country group 2 (East Europe):												
2022	242756	242754	2.5	-.0010	367	-0.2	3054	0.5	10950	-0.1	6559	-2.6
2029	267847	267843	3.6	-.0041	496	0.2	2508	-1.7	7464	-0.7	5495	-5.5
2036	289098	289098	0.9	-.0038	717	-0.6	2628	1.5	6462	0.5	5109	-0.1
2043	303168	303170	-1.9	-.0031	1078	0.1	2926	0.3	5802	0.2	4779	1.7
2050	310611	310606	4.9	.0020	1618	-1	2935	-0.7	5188	0.3	4276	-0.9
Country group 3 (Other):												
2022	641651	641648	2.8	.0007	1238	1.2	9728	0.4	40297	-1.1	9339	-5.4
2029	751636	751608	28	.0101	1645	1.3	10663	-3.2	19359	-1.1	9939	1.0
2036	847493	847487	5.7	-.0004	2294	-0.8	11986	-7.7	19053	1	10022	-0.5
2043	934536	934523	13	.0019	3236	1.4	12706	3.7	18738	-1.2	10217	-0.1
2050	1008706	1008671	35	.0046	4487	-2.7	12665	-1.4	18344	0.5	10204	-5.4

Notes: The table shows results from the national cohort-component model (CCM), BEFINN, and the average result from the microsimulation model (MS) after 1000 simulations. Results and differences are shown by country group for aggregate projected population, births, deaths, immigrations and emigrations. Differences are calculated as the results from the traditional model minus the results from the microsimulation model.

4.2 Replication of the BEFREG results

Establishing that the expanded microsimulation model can reproduce the results of the national model helps alleviate concerns about coding errors. To further ensure that regional heterogeneity is accurately captured, we next replicate the results of the regional model by enforcing its assumptions while excluding the additional heterogeneity introduced by the immigrant background dimension. Figure 4.2 illustrates the comparison of projected overall population size from the two models. As shown, the trajectories are virtually identical, with no discernible differences in either population size or population growth.



(a) Population size **(b)** Population growth

Figure 4.2 Comparison of projected population size between the regional cohort-component model and the microsimulation model

Notes: The figure compares the results from the regional projections and the combined microsimulation model. Panel (a) displays the comparison for population size, whereas panel (b) shows the comparison for population growth in percentage of the 2022 population.

We further provide more detailed comparisons of the aggregate results (see Table 4.4). Differences in aggregate population size are very small. By 2050, the average result from the microsimulation model is 93 persons higher than that of the regional cohort-component model, corresponding to a deviation of 0.002 percent. These results suggest that the microsimulation model successfully reproduces the BEFREG outcomes at the aggregate population level, and the discrepancies are negligible for planning and analytical purposes.

Comparisons across demographic components paint a similar picture. In 2050, the difference in the number of deaths amounts to only 3.1, while the discrepancy for births is 15, corresponding to percentage deviations of just 0.005 and 0.027, respectively. Immigration and emigration are likewise very close, with differences of -0.34 and 8.3. For internal migration, the difference in 2050 is only 1.2, corresponding to a percentage difference of less than 0.001.

Figure 4.3 presents density plots of relative differences (in percent) between municipal population sizes ten years into the future (2032) and at the end of the projection horizon (2050). The results indicate that errors accumulate somewhat over time: the 2032 distribution is tightly centered around zero with limited dispersion, while the 2050 distribution shows slightly wider spread. Nevertheless, the deviations remain small even in 2050.

Table 4.6 reports summary statistics of the average percentage differences at the municipality level for population size, deaths, births, immigration, emigration, in-migration, and out-migration. Overall,

Table 4.4 Comparing population results between BEFREG and the MS model

Year	Cohort-component model	Microsimulation model	Difference	
			Numerical	Percentage
2022	5,474,884	5,474,895	-10	-0.0002
2024	5,534,756	5,534,789	-33	-0.0006
2026	5,583,426	5,583,439	-13	-0.0002
2028	5,633,608	5,633,660	-52	-0.0009
2030	5,683,432	5,683,479	-47	-0.0008
2032	5,731,566	5,731,571	-5.5	-0.0001
2034	5,778,197	5,778,189	8.2	0.0001
2036	5,823,244	5,823,230	15	0.0003
2038	5,864,869	5,864,850	18	0.0003
2040	5,902,708	5,902,678	30	0.0005
2042	5,936,947	5,936,900	48	0.0008
2044	5,967,642	5,967,600	42	0.0007
2046	5,994,787	5,994,727	59	0.0010
2048	6,018,364	6,018,269	95	0.0016
2050	6,038,445	6,038,352	93	0.0015

Notes: The table shows results from the regional cohort-component model (CCM), BEFREG, and the average result from the microsimulation model (MS) after 1000 simulations. Results and differences are shown for aggregate projected population. Differences are calculated as the results from the traditional model minus the results from the microsimulation model.

the discrepancies are very small. In 2050, for instance, the mean difference in population size across municipalities is negligible, effectively zero with two decimal precision, while the minimum and maximum differences are only -0.26 and 0.31 percent, respectively. Differences for demographic events are somewhat larger, though still modest. The largest absolute discrepancy in 2050 is found for emigration, at 6.4 percent. Internal migration differences are particularly small, with low minimum and maximum values. Taken together, these results show that the microsimulation model reproduces BEFREG outcomes with high precision even at the municipality level, thereby reinforcing confidence in its robustness.

Table 4.5 Comparing aggregate component results between BEFREG and the MS model

Year	Deaths		Births		Immigrations		Emigrations		Moves	
	CCM	Diff.	CCM	Diff.	CCM	Diff.	CCM	Diff.	CCM	Diff.
2022	42,206	2.7	54,571	-6.8	67,126	-0.18	29,875	-1.6	216,059	-16
2024	42,492	5.7	54,261	-2.8	43,309	0.13	31,490	-5.7	210,502	-13
2026	43,109	-6.9	55,062	11	42,832	-0.20	30,142	-2.8	202,784	-16
2028	44,055	16	56,083	-14	42,463	0.68	29,258	-0.14	199,178	-15
2030	45,286	2.4	56,960	5	41,920	-1.4	28,862	-4.5	199,264	7
2032	46,726	3.1	57,904	15	41,386	0.53	28,688	-0.73	199,136	22
2034	48,305	-4.3	58,970	7.3	40,860	1.2	28,407	-0.30	198,648	-13
2036	49,920	3.4	60,153	-3.6	40,390	0.98	28,283	-0.93	197,725	5.5
2038	51,451	4.8	59,925	-4.4	39,979	-0.02	28,135	-1.4	196,417	-8.3
2040	52,810	1.2	59,600	15	39,634	-0.45	27,962	-1.3	194,954	5.2
2042	53,956	-7.4	59,099	3.2	39,307	0.56	27,774	5.4	193,476	7.6
2044	54,916	-3.4	58,417	-0.51	38,978	-0.74	27,572	9.7	192,174	1.3
2046	55,774	-3.9	57,627	5.9	38,627	-0.48	27,352	-7.8	191,238	4.5
2048	56,623	1.5	56,823	9.6	38,254	-0.61	27,111	-5.9	190,640	7.5
2050	57,527	3.1	56,105	15	37,876	-0.34	26,836	8.3	190,313	1.2

Notes: The table shows results from the regional cohort-component model (CCM), REGFRAM, and the average result from the microsimulation model (MS) after 1000 simulations. Results and differences are shown for aggregate projected births, deaths, immigrations and emigrations. Differences are calculated as the results from the traditional model minus the results from the microsimulation model.

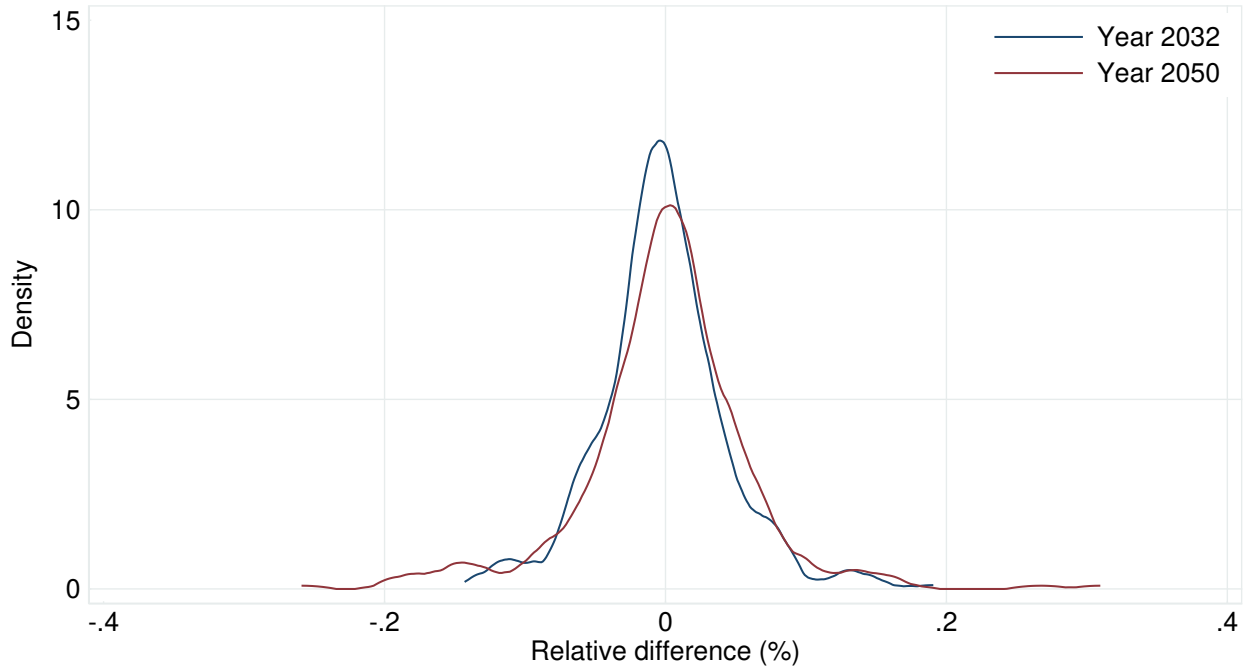


Figure 4.3 Density plot of relative differences in municipal population size results between the BEFREG model and the microsimulation model, 2032 and 2050

Table 4.6 Distribution of difference (pp.) in population and events over municipalities, by year

	Relative difference					
Year	Mean	Standard dev.	5th percentile	95th percentile	Minimum	Maximum
Population						
2022	-0.00	0.02	-0.03	0.03	-0.09	0.09
2029	-0.00	0.04	-0.07	0.07	-0.15	0.12
2036	-0.00	0.05	-0.09	0.08	-0.20	0.20
2043	0.00	0.06	-0.09	0.10	-0.26	0.45
2050	0.00	0.06	-0.11	0.09	-0.26	0.31
Deaths						
2022	-0.04	0.53	-1.04	0.76	-1.84	2.20
2029	-0.01	0.50	-0.76	0.84	-2.02	1.36
2036	0.01	0.51	-0.86	0.92	-1.79	2.17
2043	0.01	0.49	-0.79	0.83	-1.71	1.83
2050	0.02	0.54	-0.84	1.00	-2.26	2.34
Births						
2022	-0.01	0.63	-0.94	0.99	-3.83	3.10
2029	-0.01	0.68	-1.18	1.10	-3.73	3.08
2036	-0.00	0.68	-1.06	1.13	-2.78	3.08
2043	0.06	0.75	-1.07	1.37	-3.48	4.01
2050	0.02	0.70	-1.06	1.10	-2.86	2.80
Immigrations						
2022	0.04	0.89	-0.83	1.26	-4.63	8.43
2029	-0.05	0.75	-1.26	0.99	-5.53	4.35
2036	-0.03	0.80	-1.06	1.00	-7.33	3.49
2043	0.06	0.80	-1.03	1.33	-2.67	5.55
2050	0.00	0.84	-1.08	1.32	-5.59	3.42
Emigrations						
2022	-0.01	1.03	-1.82	1.61	-3.83	3.99
2029	-0.11	1.10	-2.04	1.67	-4.64	5.01
2036	-0.01	1.08	-1.69	1.52	-5.93	5.16
2043	0.03	1.17	-2.00	1.82	-5.62	5.38
2050	0.12	1.19	-1.69	2.13	-5.12	6.40
Internal in-migration						
2022	-0.03	0.32	-0.49	0.43	-1.29	1.42
2029	0.01	0.36	-0.55	0.63	-1.33	1.47
2036	0.00	0.32	-0.48	0.54	-1.55	1.16
2043	0.00	0.35	-0.57	0.54	-1.80	1.80
2050	0.01	0.33	-0.53	0.57	-1.30	1.67
Internal out-migration						
2022	0.00	0.31	-0.53	0.59	-1.26	1.07
2029	0.01	0.32	-0.50	0.50	-1.05	1.68
2036	0.02	0.34	-0.53	0.64	-1.20	1.20
2043	0.00	0.31	-0.49	0.55	-1.15	1.21
2050	0.02	0.37	-0.57	0.61	-1.71	2.36

Notes: The table shows results from the regional cohort-component model (CCM), REGFRAM, and the average result from the microsimulation model (MS) after 1000 simulations. Summary statistics of differences are shown for aggregate population, births, deaths, immigrations, emigrations and internal migration. Differences are calculated as the results from the traditional model minus the results from the microsimulation model.

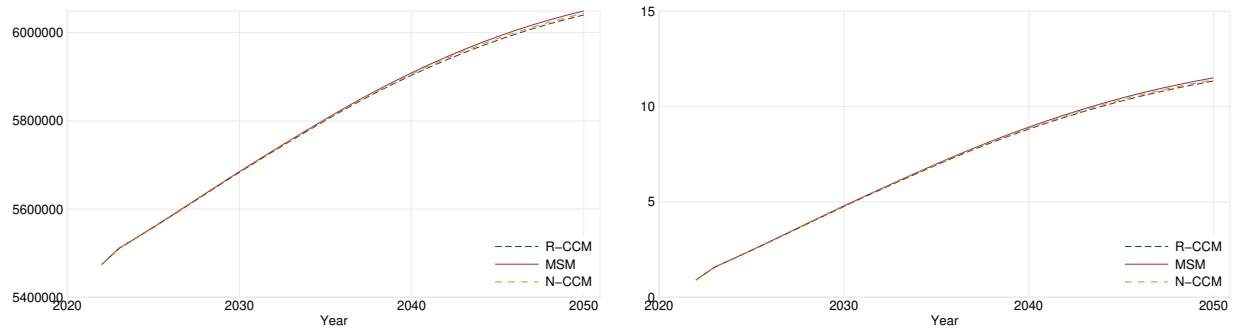
5. Combined microsimulation model

Having established that the microsimulation model is able to reproduce the results of the existing benchmark models, we now turn to the projections produced when the assumptions of the national and regional models are combined. Basically, we will allow for heterogeneity in demographic assumptions along both regional and immigrant background dimensions.

We aim to reuse the demographic assumptions from the National and Regional models described in Section 2, without the special adaptations BEFINN and BEFREG models.

Internal migration is unique to the regional model and can therefore be retained without modification, including both the internal migration rates and the moving matrix. Similarly, age- and sex-specific mortality rates vary only across municipalities and not by immigrant background, so these assumptions are also carried over from the regional model. Immigration numbers and composition are determined entirely by the national model assumptions and remain unchanged in the combined model.

Fertility and emigration, however, present a more complex challenge. In the regional model, both assumptions vary by municipality, while in the national model they vary by immigrant background. As a result, there is no straightforward way to integrate them directly into the combined framework. Instead, they must be adjusted to reflect both the geographical variation from the regional model and the group-specific variation from the national model. To achieve this, we generate a new set of assumptions using an iterative proportional fitting (IPF) algorithm applied to the fertility and emigration assumptions from both the national and regional models. The advantage of IPF is that it produces a consistent joint distribution that simultaneously preserves the marginal totals from each source, ensuring that both the regional and immigrant-dimension patterns are maintained in the combined model. A more detailed description of the method can be found in Appendix B.



(a) Population size **(b)** Population growth

Figure 5.1 Comparison of projected population size between CC models and combined MS model

Notes: The figure compares the results from the CCM projections and the combined microsimulation model. Panel (a) displays the comparison for population size, whereas panel (b) shows the comparison for population growth in percentage of the 2022 population.

We provide some preliminary evidence on how the results of the new model compare to those of the national and regional cohort-component models. A more detailed examination will be presented in a companion report. As shown in Figure 5.1, discrepancies in overall projected population are minimal. If anything, the regional cohort-component model projects a slightly lower population by 2050. Table

5.1 confirms this pattern: in 2050, the microsimulation model projects a population that is about 5,500 higher than the national model and 9,000 higher than the regional model. These findings suggest that incorporating interactions in the combined model has only a modest impact on aggregate population growth, leading to a slight upward shift in overall size. At this stage, however, we cannot assess whether population composition differs substantially across the models.

Table 5.1 Comparing population results between the CC models and the combined MS model

Year	MSM	N-CCM			R-CCM		
	Population	Population	Difference		Population	Difference	
			Numerical	Percentage		Numerical	Percentage
2022	5,474,835	5,474,990	155	0.003	5,474,895	59	0.001
2024	5,534,943	5,535,407	464	0.008	5,534,780	-164	-0.003
2026	5,584,159	5,584,770	610	0.011	5,583,427	-732	-0.013
2028	5,634,890	5,635,514	624	0.011	5,633,617	-1273	-0.023
2030	5,685,138	5,685,523	385	0.007	5,683,495	-1644	-0.029
2032	5,733,847	5,733,886	39	0.001	5,731,701	-2146	-0.037
2034	5,781,183	5,780,786	-396	-0.007	5,778,441	-2741	-0.047
2036	5,827,049	5,826,095	-954	-0.016	5,823,643	-3407	-0.058
2038	5,869,777	5,868,149	-1628	-0.028	5,865,426	-4351	-0.074
2040	5,908,963	5,906,578	-2385	-0.040	5,903,430	-5533	-0.094
2042	5,944,606	5,941,469	-3137	-0.053	5,937,854	-6752	-0.114
2044	5,976,606	5,972,675	-3932	-0.066	5,968,754	-7853	-0.132
2046	6,004,719	6,000,104	-4614	-0.077	5,996,093	-8625	-0.144
2048	6,028,832	6,023,706	-5126	-0.085	6,019,813	-9019	-0.150
2050	6,049,094	6,043,628	-5465	-0.090	6,040,052	-9042	-0.150

Notes: The table shows average results from the combined microsimulation model (MSM) after 1000 simulations, as well as results from the national cohort-component model (N-CCM) and the regional cohort-component model (R-CCM). Results and differences are shown for aggregate projected population. Differences are calculated as the results from the traditional models minus the results from the microsimulation model.

Table 5.2 reports results for the vital components—deaths and births—in the combined microsimulation model and in the cohort-component models, including both numerical and percentage differences. For deaths, the regional cohort-component model projects consistently higher numbers, with a peak difference of just over 300 in 2050, corresponding to 0.54 percent of total deaths. By contrast, the combined microsimulation model closely matches the national cohort-component model, with absolute differences below 0.01 percent.

For births, the differences are somewhat more dynamic but remain small in magnitude. Relative to the combined microsimulation model, the regional cohort-component model projects fewer births both at the beginning and at the end of the projection period, though absolute differences stay below 1 percent. When compared to the national model, the microsimulation projects slightly fewer births in the short run but somewhat more in the long run, again with deviations below 1 percent.

Table 5.3 compares projected numbers of immigrations, emigrations, and internal migrations across the two cohort-component models and the combined microsimulation model. Immigration levels are virtually identical across all three models, with only minuscule discrepancies over time. Emigration differences are somewhat larger but remain negligible, fluctuating around zero.

Table 5.2 Comparing vital component results between the CC models and the combined MS model

Year	Deaths					Births				
	MSM	R-CCM		N-CCM		MS	R-CCM		N-CCM	
		Diff.	Diff. (%)	Diff.	Diff. (%)		Diff.	Diff. (%)	Diff.	Diff. (%)
2022	42,201	2.5	0.01	11	0.027	54,520	58	0.11	167	0.31
2024	42,483	7.2	0.02	-1.9	-0.004	54,497	-232	-0.43	140	0.26
2026	43,096	8.3	0.02	16	0.036	55,289	-250	-0.45	100	0.18
2028	44,033	17	0.04	22	0.051	56,336	-250	-0.45	-7.1	-0.01
2030	45,259	25	0.05	40	0.088	57,141	-150	-0.26	-99	-0.17
2032	46,684	44	0.09	26	0.055	58,192	-248	-0.43	-159	-0.27
2034	48,244	61	0.13	25	0.052	59,262	-232	-0.39	-222	-0.38
2036	49,840	84	0.17	22	0.044	60,516	-290	-0.48	-259	-0.43
2038	51,362	90	0.18	18	0.034	60,411	-405	-0.67	-314	-0.52
2040	52,677	126	0.24	37	0.069	60,161	-490	-0.82	-359	-0.60
2042	53,810	124	0.23	20	0.037	59,646	-469	-0.79	-357	-0.60
2044	54,730	176	0.32	19	0.035	58,851	-340	-0.58	-361	-0.62
2046	55,542	226	0.41	-8.2	-0.015	57,827	-128	-0.22	-319	-0.55
2048	56,347	278	0.49	2.6	0.005	56,760	127	0.22	-223	-0.39
2050	57,199	313	0.54	-8	-0.014	55,825	341	0.61	-135	-0.24

Notes: The table shows results from the cohort-component models and the average result from the combined microsimulation model (MSM) after 1000 simulations. Results and differences are shown for aggregate projected births and deaths. Differences are calculated as the results from the traditional models minus the results from the microsimulation model.

The more notable differences appear in internal migration (measured here as out-migration rather than gross flows). Since the national cohort-component model does not include internal migration, comparisons are made only with the regional model. The combined microsimulation model projects consistently higher levels of internal migration, with discrepancies widening over time. A possible explanation is that individuals with immigrant backgrounds tend to have higher propensities to move and may also reside in, or relocate to, areas characterized by greater mobility. Additional dynamics may also arise from differences in age and sex composition.

Further investigation of this mechanism is left for future work. Taken together, these comparisons highlight that while the combined microsimulation model broadly reproduces established results, it also reveals new dynamics that warrant closer examination.

Table 5.3 Comparing migration component results between the CC and combined MS model

Year	Immigrations			Emigrations			Moves	
	MS	R-CCM Diff.	N-CCM Diff.	MS	R-CCM Diff.	N-CCM Diff.	MS	R-CCM Diff.
2022	67126	-1.6e-10	2.2e-09	29880	-3.4	1.5	218066	17
2024	43309	-7.1e-10	2.0e-10	31486	-1.5	1.8	212980	-286
2026	42832	-4.7e-10	-1.6e-10	30137	12.8	4.8	205490	-574
2028	42462	1.5e-10	-1.5e-09	29251	8.6	10.2	202139	-824
2030	41921	-3.9e-10	4.3e-10	28868	-6.7	-6.6	202557	-1069
2032	41385	7.8e-10	-2.8e-10	28682	12.0	11.0	202781	-1233
2034	40859	3.3e-10	-4.5e-10	28413	-10.0	-9.7	202610	-1262
2036	40389	1.3e-09	3.2e-10	28281	6.8	0.7	201872	-1246
2038	39979	2.6e-10	-1.6e-09	28137	-1.3	-8.1	200614	-1127
2040	39634	1.9e-09	5.8e-11	27961	-4.8	-2.1	199042	-937
2042	39306	4.7e-10	-3.2e-10	27769	4.6	-2.3	197447	-851
2044	38979	-3.7e-10	-6.8e-10	27568	2.6	5.4	195993	-824
2046	38627	5.7e-10	1.5e-10	27358	-4.4	-2.2	194941	-894
2048	38255	1.4e-09	1.1e-09	27117	-1.7	-4.2	194233	-988
2050	37876	-6.0e-10	-1.5e-09	26827	8.4	8.9	193772	-1047

Notes: The table shows results from the cohort-component models and the average result from the combined microsimulation model (MS) after 1000 simulations. Results and differences are shown for aggregate projected immigrations, emigrations and internal migration. Differences are calculated as the results from the traditional models minus the results from the microsimulation model.

6. Summary and next steps

This document has presented the construction of a new combined microsimulation model for projecting the Norwegian population by immigrant background, municipality of residence, age, and sex. Since microsimulation models are sometimes regarded as “black boxes” due to limited transparency in design choices and data inputs, we have provided a detailed account of the model’s structure and assumptions — a proof-of-concept exercise. This improves transparency to the benefit of the many users of Statistics Norway’s population projections.

In line with best practice, the new model has been validated against established benchmark models. Comparisons with the macro-type cohort-component models BEFREG and BEFINN confirm that the microsimulation model successfully reproduces their results, minimizing the risk of programming errors and strengthening confidence in its robustness.

A key advance of the new framework is that it unifies the heterogeneity previously captured separately in BEFREG (regional variation) and BEFINN (immigrant background). Looking ahead, the model offers a solid foundation for more detailed analyses of demographic behavior and policy-relevant scenarios, broadening the scope and usefulness of official population projections in Norway. While this document includes a brief exploration of extended results, a more comprehensive analysis will be presented in a companion report to follow.

References

Burch, T. K. (2018). *The Cohort-Component Population Projection: A Strange Attractor for Demographers*, pages 135–151. Springer International Publishing, Cham.

Jia, Z., Leknes, S., and Løkken, S. A. (2023). Moving beyond expectations. from cohort-component to microsimulation projections. Discussion paper no. 999, Statistics Norway.

Leknes, S. and Løkken, S. (2021). Flexible empirical Bayes estimation of local fertility schedules: reducing small area problems and preserving regional variation. Discussion paper no. 953, Statistics Norway.

Leknes, S. and Løkken, S. (2022). Befolkningsframskrivinger for kommunene 2022 (Municipal population projections 2022). Reports 2022/30, Statistics Norway.

Li, J. and O'Donoghue, C. (2013). A survey of dynamic microsimulation models: uses, model structure and methodology. *International Journal of Microsimulation*, 6(2):3–55.

Lomax, N. and Smith, A. P. (2017). An introduction to microsimulation for demography. *Australian Population Studies*, 1(1):73–85.

Morrison, R. (2008). Validation of longitudinal microsimulation models: DYNACAN practices and plans. NATSEM Working paper no. 8.

Thomas, M. J. and Tømmerås, A. (2022). Norway's 2022 national population projections: results, methods and assumptions. Reports 2022/28, Statistics Norway.

Van Imhoff, E. and Post, W. (1998). Microsimulation methods for population projection. *Population: An English Selection*, 10(1):97–138.

Zagheni, E. (2015). Microsimulation in demographic research. *International Encyclopedia of Social and Behavioral Sciences*, 15:343–346.

A. Country group classification

Country Group 1: Sweden, Denmark, Finland, Iceland, Faeroe Islands, Greenland, United Kingdom, Ireland, Isle of Man, Channel Islands, Netherlands, Belgium, Luxembourg, Germany, France, Monaco, Andorra, Spain, Portugal, Gibraltar, Malta, Italy, Holy See, San Marino, Switzerland, Liechtenstein, Austria, Greece, Cyprus, Canada, United States, Bermuda, Australia, and New Zealand.

Country Group 2: Estonia, Latvia, Lithuania, Poland, Czechia, Slovakia, Hungary, Romania, Bulgaria, Slovenia, and Croatia.

Country Group 3: All remaining countries (excluding Norway), including Africa, South and Central America and the Caribbean, Asia (excluding Cyprus), Oceania (excluding Australia and New Zealand), and all non-EU member states in Eastern Europe. Stateless people are included in this group.

Statistics Norway's comprehensive classification of grouping of countries and citizenship can be found here: `https://www.ssb.no/en/klass/klassifikasjoner/91/om`. Aggregation of some of the categories are needed to achieve the country groups described above:

$$Country\ Group\ 1 = G00 + G11 + G12 + G14 + G15$$

$$Country\ Group\ 2 = G13$$

$$Country\ Group\ 3 = G2 + G9$$

B. Description of the iterative proportional fitting method

The iterative proportional fitting (IPF) algorithm is widely used to estimate a full joint distribution when only marginal distributions across different dimensions are available. In essence, IPF allows researchers to construct a synthetic dataset that preserves known marginal totals while filling in the detailed cell values of a contingency table.

For example, suppose researchers wish to obtain detailed population data broken down by age, municipality, and immigrant group. If such a dataset is not directly available, they can instead combine auxiliary sources: one dataset providing the age-specific population at the municipal level, and another providing the national age distribution of immigrant groups. Applying the IPF algorithm yields a synthetic dataset that is consistent with both sources, effectively generating detailed population counts by age and immigrant group within each municipality.

The key property of IPF is that the resulting synthetic dataset exactly reproduces the known marginal totals. In the present context, this means that each municipality retains its observed age distribution, while the aggregate age-specific population of immigrant groups matches the national immigrant age distribution. The advantage of using IPF for model calibration is therefore clear: it produces fertility and emigration rates that remain fully consistent with the assumptions of the existing cohort-component models at both the municipal and national levels, while enabling the combined microsimulation model to incorporate heterogeneity along both dimensions simultaneously.

In this case, we use the regional model assumptions as an auxiliary data source to ensure that the marginal distribution at the age-sex-municipality level corresponds to the rates used in the regional model. Likewise, we use the national model assumptions as an auxiliary data source to ensure that the marginal distribution at the age-sex-immigrant group level corresponds to the rates used in the national model. The IPF algorithm is straightforward to implement. Below we outline the procedure, using the estimation of fertility rates as a guiding example:

1. **Fix targets.** For each of the two margins, calculate the expected number of demographic events implied by the national and regional rates. For fertility, this means first calculating the expected number of births for each age-municipality cell by multiplying the regional fertility rates with the population (of women) in the corresponding age-municipality cell.
2. **Set starting values.** Prior to the iteration, generate a set of candidate fertility rates specific to age-municipality-immigrant group cells to serve as the starting values in the IPF procedure. For simplicity, we use the arithmetic mean of the regional and national fertility rates: $r_{ajk} = (r_{aj}+r_{ak})/2$.
3. **Scale to target.** The first step in the iterative process is to calculate the expected number of births using the candidate fertility rates and the population counts.
   (a) **Regional margin:** calculate the total expected births for each age-municipality cell using the candidate rates. Adjust the candidate rates by scaling them according to the ratio of the fixed regional target to the newly calculated expected births.
   (b) **National margin:** calculate the total expected births for each age-immigrant group cell using the adjusted candidate rates. Adjust the candidate rates again by scaling them according to the ratio of the fixed national target to the newly calculated expected births.
4. **Converge.** Steps 3a and 3b are repeated until the candidate rates converge and become stable. In practice, we repeat step 3 for 100 iterations, although stability is typically achieved within 10-20 repetitions.

The final adjusted fertility rates have the attractive property that they simultaneously reproduce the number of births implied by the national model (by age and immigrant group) and by the regional model (by age and municipality).

While the IPF procedure fits our needs well in this context, it is not without limitations:

1. The resulting fertility rates can be sensitive to the choice of starting values.
2. The adjusted rates may fail to capture non-proportional patterns of underlying fertility behavior, although using more representative starting values can mitigate this issue.
3. The procedure can perform poorly when groups or individual cells contain no events or population counts, leading to instability.

C. Assumptions and input files

C.1 Fertility assumptions

fertility_rates_n.csv contains age-specific fertility rates for females by municipality and immigrant type (country group, immigrant background, duration of stay) for the base year.

fertility_trend.csv contains yearly scaling factors for adjusting the trend in fertility relative to base year.

fertility_pr2gen.csv contains the probability that a child born to a immigrant woman also has an immigrant father. If so, the child becomes a Norwegian-born of two immigrant parents ("second generation immigrant"), inheriting the country background of its mother.

C.2 Mortality assumptions

mortality.csv contains yearly age, sex and municipality specific mortality rates. Mortality rates have different trends for each age and sex group. Mortality rates are not specific to immigrant types.

C.3 Migration

immigration_fixed.csv number of yearly international immigrations and emigrations by country background. International migration numbers are taken as fixed (exogenous) assumptions by the model.

immigration_distr.csv contains the probabilities that determine the composition of each immigrant group. Conditional on year and country background, describes the distribution across age, sex and immigrant characteristics.

migration_rates.csv contains the internal migration and emigration rates. The internal migration rates are age, sex and municipality specific while emigration rates are age, sex, municipality and immigration characteristics specific.

mov_mat_rates.csv contains full probability distributions over target municipalities for movers. Conditional on municipality of departure (origin) and 20 unique age- and sex groupings.

mov_mat_link.csv is an index linking any age-sex combination to the unique age-sex groupings used in the moving matrix.

C.4 Crosswalk between `knr` and `region`

df_knr.csv is a file linking the official municipality code (“kommunenumme”), `knr`, to the internal regional index `region` used in the Python implementation.

C.5 Base population

population_base.csv contains the full population of individuals at the start of the projection base year. Individual characteristics include age, sex and municipality of residency, immigrant country background (4

groups), immigrant generation (foreign-born or Norwegian-born) and duration of stay (whole years since first registration).

D. Detailed Description of the Python Code

D.1 Project Structure

The project is organized into a set of Python modules, each with its role in setting up, running and processing the population simulation. The following is a high-level overview of its main components.[3]

- **sim_para.py**

  Defines all global parameters used across the project simulation files, such as number of simulations, projection years, maximum age, fertility range, CPU cores, and input/output paths.

- **model.py**

  Contains the `Model` class, which loads and manages demographic rate tables (fertility, mortality, migration, etc.) from CSV files. Provides methods like `set_fertility`, `set_mortality`, `set_migration`, and others to prepare these tables for use in the simulation.

- **population.py**

  Defines the `Population` class, which represents and evolves the simulated population. Each individual is stored as a row in a pandas DataFrame with attributes such as age, sex, region, immigration group, generation, duration of stay, etc. This class contains the yearly update logic (aging, births, deaths, immigration, emigration, internal moves) and methods for producing summary statistics.

- **sim_setup.py**

  Provides helper functions `build_model()` and `build_population()` to initialize a simulation-ready model and the base population from input CSV files.

- **sim_run.py**

  Contains the function `population_sim`, which runs one full simulation of the population for the specified number of years, writing results to a summary store.

- **sim_main_refactor.py**

  The main driver script. It initializes the Zarr summary stores, spawns multiple worker processes (using `multiprocessing.Pool`), and distributes simulations across available CPU cores. Completed simulations are aggregated batchwise for intermediate reference and analysis, and results are exported to Stata. A README file is also generated by this script.

- **data_processing.py**

[3] The Python scripts are continuously updated. This description reflects the code as of 23 October 2025; changes should be expected.

Provides functions for post-simulation processing: aggregating results from Zarr, exporting to Stata, and managing grouped summaries across simulations.

D.2 Module Reference

D.2.1 `population.py`

Purpose Represents the entire population as a DataFrame and advances it one year (aging, births, immigration, deaths/out-migration, internal moves). Writes yearly stock/event summaries to Zarr store.

Classes

- `Population` class: contains a the `members` (pandas DataFrame) of the population, as well as methods/logic to advance and summarize the population through stages in a given year.
- `Status` enum: `NONE=0, DEATH=1, EMIGRATION_OUT=2, INTERNAL_MOVE=3, IMMIGRANT=4`. The statuses an individual can have at the end of the year. REFER TO STATUS LOGIC OR SOMETHING

Public API (selected)

- `generate_population(file)`: stream-load base year CSV, map municipality to region, compute duration groups and age–sex group, precompute emigration probs.
- `age_population()`: +1 age; update duration_stay, duration groups, fertility flags, age–sex group.
- `apply_births()`: merge fertility rates & trend; Bernoulli births; append newborn rows; assign 2nd-gen with `pr_2nd`; refresh emigration probs.
- `immigration_step()`: sample synthetic immigrants from empirical trait distribution by origin group; append rows.
- `apply_death_move_emigration_step(sim_index)`: vectorized assignment of one outcome (death, emigration, internal move, none) for non-current-year immigrants.
- `move_assign()`: route movers and immigrants via movement matrix (year, origin, age–sex group) $\rightarrow$ destination region.
- `summary_event()`, `summary_year()`: histogramdd-style event/stock counts into Zarr (dims: year, age, sex, region, imm_group, gen, duration, event_type).
- `event_to_file()`, `save_to_file()`: optional yearly .dta exports.
- `step(sim_index)`: orchestrates the full year and outputs.

Assumptions & invariants

- Duration groups for gen $\neq 1$ forced to 0; immigrants added with special region code for “abroad” before routing.
- Exactly one of {death, emigration, internal move, none} occurs per person per year.

D.2.2 `model.py`

Purpose Load, normalize, and index all demographic tables from CSV; provide fast lookups for probabilities and routing.

Loaded tables (indices)

- Fertility: `(age, region, imm_group, gen, durgr_fert)` with per-year trend factors.
- Mortality: `(year, region, age, sex)`.
- Emigration rates: `(age, sex, region, imm_group, gen, durgr_emig)`; yearly migration totals `cg{k}_in/out`.
- Internal migration: `(year, region, age, sex)`.
- Immigration trait distribution: weighted sampling via `traits_prob`.
- Movement matrix: rows `(year, origin_region, agesexgr)`, columns destination regions.
- Link table: `(age, sex)` $\rightarrow$ `agesexgr`; municipality $\rightarrow$ region mapping.

Public API (selected)

- `set_knr_mapping`, `set_fertility`, `set_fertility_trend`, `set_fertility_pr2nd`
- `set_mortality`, `set_internal_migration`, `set_migration` (emigration)
- `set_migration_total`, `set_immigration_dist`, `set_mov_mat`, `set_link_a`

Assumptions

- `sex` recoded 1/2 $\rightarrow$ 0/1; immigrant group `imm_group = cgr % 4`; natives map to 0.
- Required columns are validated; rows with missing critical fields dropped.

D.2.3 `sim_setup.py`

Purpose Single-use builders: `build_model()` (loads all CSVs) and `build_population(model)` (stream-loads base population, precomputes fields).

D.2.4 `sim_run.py`

Purpose Run a single simulation through `population_sim`: deep-copy the template population, iterate `NUMBER_YEAR` times, write per-sim summaries to Zarr.

D.2.5 `sim_main_refactor.py`

Purpose Main simulation driver. Initializes Zarr stores (summary and aggregate), spawns worker pool with CPU affinity, monitors completions, folds finished sims into the aggregate store, and triggers exports.

Summary/aggregate shapes

- Summary: `(sim, year, age, sex, region, imm_group, gen, duration, event_type)`.
- Aggregate: `sum` dataset with `(year, age, sex, region, imm_group, gen, duration, event_type)` plus scalar `nsims`.

D.2.6 `data_processing.py`

Purpose Post-processing utilities.

- `aggregate_zarr(xarr, g, sim_indices)`: fold sims into aggregate `g["sum"]` blockwise over years.
- `to_stata_zarr_group_dim(agg_group, out_dir, dims_to_keep)`: outputs one .dta per event type with the specifications given such as which dimensions to keep.

D.3 Projection implementation

D.3.1 Initializing the Base Population

Before running any projection, the code must load a base-year population. This is done with:

Method `Population.generate_population(file)`

- Reads `population_base.csv` in chunks (pandas `chunksize = 50,000`). Each row has columns: `knr, sex, age, cgr, gen, duration`.
- Maps `knr` to internal `region` codes via the model's `knr_mapping`.
- For each row in a chunk:
  - Computes `durgr_fert` (fertility duration group) for females and `durgr_emig` (emigration duration group) for foreign-born immigrants (`cgr != 0`).
  - Derives `imm_group = cgr % 4`.
  - Calls `create_member(...)` to build a dictionary with all attributes (`id, age, sex, region, imm_group, gen, durgr_fert, durgr_emig, duration_stay, POB`, etc.).
- Collects all new individuals into a list and adds them to the `members` DataFrame using `batch_add_members()`.
- After reading all chunks, computes the `age_sex_group` and updates the `pr_emig` attributes of each member, along with an initialization to `size = len(members)` (the number of members loaded).

After this process, the `Population` object holds the full base-year population, with every individual assigned demographic attributes and probabilities, ready to be projected forward.

D.3.2 How the Projection Is Performed

The core projection evolves the population through each annual time-step:

1 **Aging and Updating Individual Probabilities**

- At the start of each year, the `age_population()` method is called, and for each row(member) of the `members` DataFrame:
  - Increments `age` by 1 and `time_index` by 1.
  - reset status variables
  - Updates `duration_stay` if the individual is an immigrant (so that migration/emigration probabilities reflect time since arrival), and the respective `durgr_emig` and `durgr_fert`.
  - Recomputes the `age_sex_group` of the members using the `agesex_table`
- This aging step ensures everyone grows older and potentially moves between probability bins as they cross age/duration thresholds.

2 **Fertility (Births) Step**

- Next, the population (after aging) run through the `apply_births()`.
- The function starts by filtering out the fertile members, and merges with the `fertility_table` on their `age`, `region`, `imm_group`, `gen` and `durgr_fert`.
- This probabilities are adjusted by a year-specific trend factor through the `set_fert_trend()`.
- Then we draw the births based on these probabilities.
- If the mother is an foreign-born immigrant, we assign the probability that the child will be a Norwegian-born with immigrant background ("second generation immigrant"), and the `imm_mother` attribute is updated.
- We then make a copy of all the mothers as newborns and then set their attributes as follows, and merge them into the main population.

```
    # Set newborn attributes
    newborns['sex'] = np.random.binomial(1, 1-NEWBORN_MALE_PROBABILITY, size=len(
        newborns))  # Randomly assign
    newborns['age'] = 0  # Newborns are 0 years old
    newborns['fertile'] = False  # Newborns are not fertile
    newborns['status_birth'] = 0  # Newborns have not given birth
    newborns['id'] = range(self.next_id, self.next_id + len(newborns))  # Assign new
        IDs
    self.next_id += len(newborns)  # Update next ID for future members
    newborns['duration_stay'] = 0  # Newborns have just arrived
    newborns['durgr_fert'] = 0  # Newborns fertility group initially set to 0
    newborns['durgr_emig'] = 0  # Newborns emigriation group initially set to 0

    newborns['region'] = mothers['region'].values  # Newborns are born in the same
        region as their mother

    # Filter newborns to set generation based on imm_mother
```

```
    # If imm_mother is True, set generation to 2 and imm_group based on the mother's
        imm_group
    newborns['gen'] = np.where(newborns['imm_mother'], 2, 3)  # 2nd generation if
        imm_mother is True, else 3rd generation


    # Set imm_group for newborns
    newborns['imm_group'] = np.where(newborns['imm_mother'], mothers['imm_group'].
        values, 0) # 0 for natives, imm_group for second generation


    # Set POB (place of birth) for newborns, same region as mother
    newborns['POB'] = newborns['region']  # Newborns are born in the region of their
        mother
    newborns['region_old'] = newborns['region']  # Keep the original region for
        newborns

    newborns['age_sex_group'] = 1
```

- Lastly, we update the emigration probabilities of all the members of the population.

3 **Immigration Step**

- The `population.immigration_step()` method is then called. For each of the four immigrant origin groups (`from_imm_group = 0..3`), the code:
  - Note that native is denoted by 0 in the code, but 4 in the befinn input data files.
  - Reads the target number of immigrants (`n`) from `migration_total` for the current year and immigrant group.
  - Samples `n` synthetic immigrants from an empirical trait-distribution table (`immigration_table`), initializing default attributes before appending them to an `all_new_immigrants` DataFrame.
  - These new immigrants are appended to the population.
- Imported immigrants start with `region = NUMBER_REGION` (a special "abroad" code).

4 **Mortality, Emigration, and Internal Moves**

- After births and immigration, the model applies `apply_death_move_emigration_step(sim_index)`:
  - Excludes those just imported (`status = IMMIGRANT`) from mortality and migration risks.
  - For all others, retrieves death probabilities from `model.mortality_table` indexed by (`year, region, age, sex`). Individuals older than `MAX_AGE` receive death probability 1.0.
  - Uses each individual's cached emigration probability (`pr_emig`) and computes internal move probability (from `internal_migration_table`) indexed similarly to the mortality table.

- A uniform random number per person decides if they die (`status=DEATH`), emigrate (`status=EMIGRATION_OUT`), move internally (`status=INTERNAL_MOVE`), or if nothing happens. Only one outcome is assigned per person.

- Next, those with `status` $\geq$ `3` (internal movers and immigrants) pass through `move_assign()`:
  - Keeps `region_old` as the pre-move region.
  - Groups individuals by (`region_old`, `age_sex_group`) and samples new destinations from `model.mov_mat`, a probability matrix indexed by (`year, origin region, agesexgr`) with columns of destination regions.
  - Updates `region` to the chosen destination.

5 **Summarizing and Cleaning Up**

- The simulation then records event outcomes:
  - `summary_event(sim_index)` builds N-dimensional histograms for mothers, deaths, immigrants, emigrants, and movers. These are added to the Zarr summary store under the corresponding event type slot.
  - `event_to_file(...)` optionally writes a Stata dataset listing individuals who had non-zero events (birth, death, migration).
- The population is then pruned by `cleanup_exit(sim_index)`:
  - Removes rows with `status=DEATH` or `status=EMIGRATION_OUT`.
  - Resets per-year flags (`status, status_birth, imm_mother`) for survivors, and updates `region_old`.
- Finally, `summary_year(sim_index)` records the full surviving population distribution (age × sex × region × immigrant group × generation × duration) into the summary store, under event type 0 (`population`).

6 **Iterating Over Years**

- The method `population.step(model)` encapsulates steps 1–5 for one year.
- In `population_sim(...)`, this is repeated for `number_year` iterations. At each iteration, summary arrays (population by age/sex/region, births, deaths, migration flows) are recorded.

Overall, this loop implements a *stochastic cohort-component microsimulation* at the individual level: each person ages, can give birth, can die, can move internally, or can emigrate, all according to probability tables that vary by age, sex, region, immigrant status, and generation. New immigrants and newborns are injected each year, and their attributes are drawn from empirical distributions.

D.3.3 The `Population` Class: Managing a Cohort of Individuals

Attributes

- `members`: A `pandas.DataFrame` where each row is one individual. Columns include:
  - `id`: Unique integer identifier for the individual.
  - `age`: Age in years.
  - `sex`: Sex (0 = male, 1 = female).
  - `fertile`: Boolean (stored as 0/1) indicating if individual is in fertility range (15–49 and female).
  - `status`: Current life/mobility status (see `Status` enum: 0 = NONE, 1 = DEATH, 2 = EMIGRATION_OUT, 3 = INTERNAL_MOVE, 4 = IMMIGRANT).
  - `status_birth`: Indicator (0/1) whether the individual gave birth this year.
  - `region`: Current region code.
  - `imm_group`: Immigration group (0 = natives, 1–3 immigrant groups).
  - `gen`: Generation (1 = first, 2 = second, 3 = third+ and natives).
  - `durgr_fert`: Duration group used for fertility probabilities (thresholded from `duration_stay`).
  - `durgr_emig`: Duration group used for emigration probabilities (thresholded from `duration_stay`).
  - `duration_stay`: Years since arrival in Norway (0 for natives, 0 for newborns, increments annually for immigrants).
  - `imm_mother`: Boolean (0/1) whether the individual gave birth to a Norwegian-born child with foreign-born parents ("second-generation immigrant") this year.
  - `POB`: Place of birth (region for natives, special abroad code (`NUMBER_REGION`) for immigrants).
  - `region_old`: Region of residence in the previous year (used for tracking movers).
  - `age_im`: Age at immigration (set when immigrating).
  - `age_sex_group`: Derived categorical group combining age and sex, used in internal migration matrix.
  - `pr_emig`: Cached probability of emigration, looked up from migration tables.
- `size`: Current population size (`len(members)`).
- `next_id`: Counter for assigning unique IDs to new individuals.
- `model`: The associated `Model` instance with demographic probability tables.
- `agesex_table`: Lookup table mapping (`age`, `sex`) $\rightarrow$ `agesexgr`.
- `time_index`: Current simulation year index (starts at -1 and increments each step).
- `summary_store`: A Zarr-backed storage object for saving annual summaries.
- `event_bins`: List of bin edges for histograms: `[age, sex, region, imm_group, gen, duration_stay]`.

Helper Methods

- **`year()`**: Returns calendar year as `start_year + time_index`.
- **`update_emig_probs()`**: Recomputes each member's `pr_emig` from `model.migration_table`.
- **`update_fertility_flags()`**: Sets `fertile = True` for women aged 15–49, else `False`.

Simulation Methods

- **`age_population()`**:
  - Increments `age` by 1 and advances `time_index`.
  - Increments `duration_stay` for immigrants.
  - Recomputes `durgr_fert` and `durgr_emig` from thresholds.
  - Updates `age_sex_group` from `agesex_table`.
- **`set_fert_trend()`**: Returns year-specific fertility adjustment factor (default 1.0).
- **`apply_births()`**:
  - Filters fertile women and merges with fertility probabilities.
  - Simulates births with fertility trend adjustments.
  - Determines immigrant mothers who gives birth to Norwegian-born ("second-generation") children (`imm_mother = True`).
  - Creates newborns with reset attributes (`age = 0`, `duration_stay = 0`, new `id`, `gen`, `imm_group`, `POB`, etc.).
  - Appends newborns to `members` and updates emigration probabilities.
- **`immigration_step()`**:
  - For each immigrant group, samples the required number of new immigrants from `model.immigration_table`.
  - Sets attributes: `region = NUMBER_REGION` (abroad), `POB = NUMBER_REGION`, `status = IMMIGRANT`, `age_im`, fertility flags, duration groups.
  - Assigns new IDs and appends them to `members`.
- **`apply_death_move_emigration_step(sim_index)`**:
  - Excludes new immigrants from mortality and migration.
  - Looks up mortality, emigration, and move probabilities for others.
  - Draws random outcomes to assign `status` $\in 0, 1, 2, 3$.
- **`move_assign()`**:
  - For movers (`status` $\geq$ `3`), samples new `region` from `model.mov_mat`, based on `region_old` and `age_sex_group`.

  - Updates `region` accordingly.

Output and Cleanup Methods

- **`event_to_file(filename)`**:
  - Extracts individuals with non-zero events: birth, death, emigration, immigration, internal move.
  - Outputs selected attributes: `age, sex, region, imm_group, gen, duration_stay, mother, died, emmig, immig, move_from, imm_mother`.
- **`cleanup_exit(sim_index)`**:
  - Removes rows with `status = DEATH` or `status = EMIGRATION_OUT`.
  - Resets `status`, `status_birth`, `imm_mother` for survivors.
  - Updates `region_old`.
- **`save_to_file(filename)`**:
  - Saves surviving population attributes (`age, sex, region, imm_group, gen, duration_stay`) to Stata file.
- **`summary_event(sim_index)`**:
  - Aggregates event histograms (mothers, deaths, immigrants, emigrants, movers) and writes to Zarr summary store. Note in the zaar data files, "gen" takes value 0, 1 and 2, as we use them directly as indexes. In all other output files as well as in the rest of data manipulations, it takes value 1, 2 and 3.
- **`summary_year(sim_index)`**:
  - Aggregates full population stock (age $\times$ sex $\times$ region $\times$ imm_group $\times$ gen $\times$ duration_stay) and writes to Zarr. Note that "gen" takes the same values as in summary_event.

One-Year Step: `Population.step(sim_index)`

1. `age_population()`
2. `apply_births()`
3. `immigration_step()`
4. `apply_death_move_emigration_step(sim_index)`
5. `move_assign()`
6. `event_to_file()`, `summary_event(sim_index)`
7. `cleanup_exit(sim_index)`
8. `save_to_file()`, `summary_year(sim_index)`

After this function, the DataFrame `members` represents the surviving and updated population at year $t + 1$, with newborns and immigrants added, exits removed, and all events and stock recorded.

D.3.4 The `model` Class: Loading and Storing Demographic Inputs

The `model` class holds all input tables and related utilities. It is instantiated once and then passed around to populations and individuals to read rate tables.

Constructor (`__init__`)

- Accepts empty `pandas.DataFrame()` defaults for `mortality`, `immigration`, `migration_total`, `fertility`, `fertility_trend`, `pr_2nd`, `migration_table`, plus:
  - `im_dist_table`: placeholder list for immigrant trait distribution, one DataFrame per year.
  - `mov_mat`: DataFrame for internal move probabilities.
  - `table_link_a`, `table_link_r`: mapping tables to group ages and regions into broader categories.
  - `knr_mapping`: DataFrame mapping Norwegian municipality codes ("knr") to internal region indices.
  - `durgr_fert_thresholds`, `durgr_emig_thresholds`: predefined duration-group thresholds for fertility and emigration.
- All these are stored as attributes so that downstream code can look up rates.

Data-Loading Methods

Each `set_*` method reads a CSV from the input path (`path`) and transforms it into a `pandas.DataFrame` indexed appropriately:

`set_knr_mapping(self, file)`

- Reads a CSV mapping "knr" $\rightarrow$ "region".
- Stores a `pandas.Series` `knr_mapping` indexed by municipality code.

`set_fertility(self, file, knr_mapping)`

- Reads a fertility CSV with columns `knr, age, cgr, gen, durgr_fert, pr_fert`, among others.
- Maps `knr` $\rightarrow$ `region` using `knr_mapping`.
- Derives `imm_group = cgr % 4`.
- Keeps only `{region, age, imm_group, gen, durgr_fert, pr_fert}`, then sets a multi-index on these five columns.
- Stores the result as `self.fertility_table`.

`set_fertility_trend(self, file)`

- Reads a CSV with columns `year, factor`.
- Indexes by `year` and stores it as `self.fertility_trend`.

`set_fertility_pr2nd(self, file)`

- Reads a CSV with columns `year`, `cgr_1`, `cgr_2`, `cgr_3`.
- Indexes by `year` to store second-generation probabilities (`self.pr_2nd`).

`set_mortality(self, file, knr_mapping)`

- Reads a CSV with `knr`, `year`, `age`, `sex`, `prob`.
- Maps `knr` → `region`, converts `sex` from 1/2 to 0/1 by subtracting one.
- Keeps `{year, region, age, sex, prob}`, indexes by those four columns → `self.mortality_table`.

`set_migration(self, file, knr_mapping)`

- Reads a CSV with `knr`, `age`, `sex`, `cgr`, `gen`, `durgr_emig`, `pr_emig`, `pr_mov`.
- Maps `knr` → `region`, `sex = sex − 1`, computes `imm_group = cgr % 4`.
- Keeps only `{region, age, sex, imm_group, gen, durgr_emig, pr_emig}`.
- Indexes by `(age, sex, region, imm_group, gen, durgr_emig)` → `self.migration_table`.

`set_internal_migration(self, file, knr_mapping)`

- Reads a CSV with `year`, `region`, `age`, `sex`, `pr_mov`.
- Maps `knr` → `region`, converts `sex` from 1/2 to 0/1 by subtracting one.
- Keeps `{year, region, age, sex, pr_mov}`
- indexes by those four columns → `self.internal_migration_table`.

`set_migration_total(self, file)`

- Reads a CSV with columns `year`, `cg0_in`, `cg1_in`, `cg2_in`, `cg3_in`, `cg0_out`, `...`, `cg3_out`.
- Re-indexes by `year`, ensures numeric, and renames `cg4_` columns if present → `self.migration_total`.

`set_immigration_dist(self, file)`

- Reads an "immigrant trait distribution" CSV with columns including `cgr`, `from_cgr`, `sex`, `age`, `duration`, `gen`, `traits_prob`.
- Computes `imm_group = cgr % 4` and `from_imm_group = from_cgr % 4`.
- Converts `sex = sex − 1`.
- Keeps `{from_imm_group, age, sex, duration, gen, imm_group, traits_prob}`.
- Stores any row with `traits_prob > 0` into `self.immigration_table`.

`set_mov_mat(self, file, knr_mapping)`

- Reads an internal movement CSV with `knr_origin`, `knr_destination`, `year`, `agesexgr`, `prob`.
- Maps both origin/destination `knr` → `region`, filling `NaN` with `number_region` for "abroad" origins.
- Converts `agesexgr = agesexgr − 1`.

- Keeps columns {year, region, agesexgr, d_region, prob}.
- Pivots so that index = (year, region, agesexgr), columns = d_region, values = prob → self.mov_mat.

set_link_a(self, file)

- Reads a CSV mapping (age, sex) → agesexgr.
- Converts sex = sex − 1, agesexgr = agesexgr − 1.
- Indexes by (age, sex) → self.table_link_a.

After loading all of these tables, the model object is ready to drive a microsimulation.

E. Model deviations and caveats

Translating cohort-component models into a microsimulation framework can be a valuable exercise in several ways. Because the two models should be equivalent, any differences in results can be signs of errors or mistakes in either model implementation or the demographic assumptions. In microsimulation models, the demographic behavior and timing of events has to be explicitly modeled from the perspective of the individual. This means any illogical event sequence becomes very apparent, like if an individual dies and then moves. In cohort-component models, on the other hand, all the population data and demographic rates has to be stored as matrices for the model to be run. This checks that the dimensions of the population corresponds to the assumptions, but other mistakes might be tricky to catch.

E.1 List of model deviations

Internal re-migration. Individuals who were migrating internally were not prohibited from relocating to their municipality of departure. As internal migration is defined as residing in a different municipality at the start- and end of the same year, this meant the model produced to several thousand moves less per year than intended. This error was discovered while working on an earlier version of the microsimulation model Jia et al. (2023) and was corrected for the 2022 projections.

In the development of the first microsimulation model for replicating the BEFREG model one unintended modeling quirk was discovered. The internal migration rates was used to first calculate the number of movers before redistributing them to municipalities according to a moving matrix. Compared to the microsimulation model, the CCM model had fewer total internal migrations. The cause of this was that the moving matrix allocated a portion of the internal migrants back to their municipality of origin, effectively cancelling the moves. This error was fixed in the official regional projections for 2024.

Fertility adjustment in BEFINN. In both projection models, women aged 15-49 are assumed to be of fertile age with positive fertility rates. Assumptions about the development of the TFR is stated and used to adjust the trend over time for all age-specific fertility rates across all groups (municipality of residence, country background, duration of stay, etc.). However, while comparing the number of births between the two models, it became clear that in BEFINN this adjustment factor was not applied to women aged 45–49,

only women aged 15–44. Since the fertility rates are low for older women, this only resulted in about 20 extra births after 2035. Still, these differences accumulate over time and makes the population in the CC model smaller by approximately 500 individuals in 2050.

Internal migration in BEFREG age-shifted. The internal migration rates used in the 2022 BEFINN model are age-shifted, meaning individuals are assigned the rates intended for one year older individuals. The mistake likely originates from an ad-hoc fix introduced in the previous 2020 BEFREG projections. While the 2022 internal migration rates of the regional model are correctly estimated using the end-of-year age, the model framework wrongly assumes they have been estimated using start-of-year age. So, when the rates are prepared for use in the BEFREG projections the model "corrects" for this by replacing the age variable with $age + 1$. However, this means that newborns (end-of-year age 0) are assigned the moving propensities for children of age 1 (end-of-year), which typically are lower. Similarly, for every age (up to age 69) are assigned the internal migration rates estimated on, and intended for, individuals who are one year older. The consequences of this age shift does not seem to make much difference in aggregate. Even if newborns are less likely to move and the high internal migration rates experienced in the late teens and early twenties happen one year earlier, much of the effect on the age structure of municipalities is neutralized by the internal migration flows into the municipalities also being age-shifted by one year.

Immigration age-sex distribution. In BEFREG, the assumptions for age-sex distribution of immigrants was calculated improperly. An error in a script running in the BEFINN model wrongly attributed the immigration flows for year $t + 1$ to year t instead. However, the deviations are very small relative to the correct calculations, but nonetheless introduces a discrepancy between the BEFINN and the BEFREG models. The composition differences are most noticeable in 2023, when the distribution was calculated without adjusting for the extra number of Ukrainian immigrants. This did not matter for later years, as the gradual change in immigration flows meant that the immigrant distribution shares in adjacent years would be close to identical.