

# Technical Report: Methodology Behind the Advanced Demographic Modeling App (WPP 2024)

Chat-GPT-5-Pro, Manuel A. Rivas

November 11, 2025

## Abstract

This report explains the modeling approach implemented in the *Advanced Demographic Modeling App* (WPP 2024). The app reads UN World Population Prospects (WPP) data for 2023 and projects population using three models: (i) a deterministic exponential model, (ii) a stochastic geometric Brownian motion (Itô) model, and (iii) a Galton–Watson branching process. We show how the app maps demographic inputs such as Total Fertility Rate (TFR), replacement fertility, mean age at childbearing (MAC), and net migration rate into the mathematical parameters of each model. The math is presented with intuition suitable for advanced high-school students.

## Contents

<b>1</b>	<b>What the App Does, In Plain Language</b>	<b>2</b>
<b>2</b>	<b>Data Handling and Units</b>	<b>2</b>
2.1	Input table and filtering . . . . .	2
2.2	Population units . . . . .	3
2.3	Key demographic inputs . . . . .	3
<b>3</b>	<b>From Demography to a Growth Rate</b>	<b>3</b>
3.1	Replacement fertility and basic reproduction number . . . . .	3
3.2	Malthusian (per-year) growth rate . . . . .	3
3.3	Adding migration . . . . .	3
<b>4</b>	<b>Model 1: Deterministic Exponential Growth</b>	<b>4</b>
4.1	Differential equation and solution . . . . .	4
4.2	“Time to collapse” (code feature) . . . . .	4
4.3	Algorithm in the app . . . . .	4
<b>5</b>	<b>Model 2: Stochastic Itô (Geometric Brownian Motion)</b>	<b>4</b>
5.1	Why a stochastic model? . . . . .	4
5.2	Exact simulation step for GBM . . . . .	4
5.3	Setting the volatility from a TFR slider . . . . .	5
5.4	Algorithm in the app . . . . .	5

<b>6</b>	<b>Model 3: Galton–Watson Branching (Generation-by-Generation)</b>	<b>5</b>
6.1	Core idea . . . . .	5
6.2	Generation length and timeline . . . . .	5
6.3	Distributional approximation used in the code . . . . .	6
6.4	Algorithm in the app . . . . .	6
<b>7</b>	<b>Scenarios and How They Modify Inputs</b>	<b>6</b>
<b>8</b>	<b>Putting It Together: Code-to-Math Map</b>	<b>7</b>
<b>9</b>	<b>Interpretation Tips (High-School Intuition)</b>	<b>7</b>
<b>10</b>	<b>Model Limitations and Design Choices</b>	<b>7</b>
<b>11</b>	<b>Worked Micro-Example (Numbers Picked for Illustration)</b>	<b>8</b>
<b>12</b>	<b>Reproducibility Checklist</b>	<b>8</b>
<b>A</b>	<b>Deriving <math>r = \ln(R_0)/G</math></b>	<b>8</b>
<b>B</b>	<b>Closed-Form Step for GBM</b>	<b>8</b>

## 1 What the App Does, In Plain Language

The app lets you pick a place (a country or the world), then reads the latest (2023) estimates from the UN WPP dataset. From those data it builds a handful of parameters:

- initial population  $P_0$  (people at the start of 2023),
- total fertility rate (TFR), the average number of children per woman,
- mean age at childbearing (MAC), used as a rough “generation time”,
- life expectancy (displayed but not used in the core equations), and
- net migration rate (per 1,000 people per year).

You can then choose a **scenario** (Base, Low/High Fertility, Catastrophic, or Custom) that tweaks these inputs, and select one of three **models** to project the population forward by a chosen number of years. The app plots either a single projection curve (deterministic) or an average band across many random simulation paths (stochastic).

## 2 Data Handling and Units

### 2.1 Input table and filtering

The app loads a CSV named `output.csv`, skipping the first 14 metadata rows and converting relevant columns to numeric types. It then filters data for:

```
Year = 2023, Variant = "Estimates", Type ∈ {Country/Area, World}
```

and selects your chosen *Location*.

## 2.2 Population units

The UN WPP table stores many counts in thousands. The app converts the January 1 population to actual people by multiplying by 1,000:

$$P_0 = \text{TPopulation1Jan} \times 1000.$$

## 2.3 Key demographic inputs

- **TFR** is read directly as a number of births per woman.
- **MAC** is read in years; the app uses it as an estimate of the average time between generations (a common heuristic in simple population models).
- **Net migration rate** is provided “per 1000 population per year”. The app divides by 1000 to convert that to an annual fractional growth (e.g., +3 per 1000 becomes +0.003 per year).

# 3 From Demography to a Growth Rate

## 3.1 Replacement fertility and basic reproduction number

Let  $F$  denote the TFR under the chosen scenario and  $F_{\text{rep}}$  be the *replacement* TFR (default 2.1, or 2.3 in the Catastrophic scenario). The app defines the reproduction ratio

$$R_0 = \frac{F}{F_{\text{rep}}}.$$

Intuition:  $R_0 > 1$  means, on average, each generation is larger than the previous;  $R_0 < 1$  means the opposite.

## 3.2 Malthusian (per-year) growth rate

Using MAC, denoted  $G$  (in years), as a generation-length proxy, the app sets the intrinsic growth rate

$$r_{\text{fert}} = \frac{\ln R_0}{G}.$$

This comes from the idea that after one generation (length  $G$ ), the size multiplies by  $R_0$ . If growth were smooth and continuous, then  $P(t) \approx P_0 e^{r_{\text{fert}} t}$ , so matching  $P(G) = P_0 R_0$  gives  $e^{r_{\text{fert}} G} = R_0$  and hence  $r_{\text{fert}} = \frac{\ln R_0}{G}$ .

## 3.3 Adding migration

The migration term  $m$  is added to the drift:

$$r = r_{\text{fert}} + m, \quad \text{where } m = \frac{\text{NetMigrationRate}}{1000}.$$

This treats net migration as an extra per-year fractional change, independent of births and deaths. It is a simplification, but reasonable for a first-order model.

## 4 Model 1: Deterministic Exponential Growth

### 4.1 Differential equation and solution

The model assumes the population changes at a constant rate  $r$ :

$$\frac{dP}{dt} = rP.$$

Solving this yields

$$P(t) = P_0 e^{rt}.$$

If  $r > 0$ , population grows; if  $r < 0$ , it shrinks. The *doubling time* is  $T_d = \frac{\ln 2}{r}$  (only meaningful if  $r > 0$ ).

### 4.2 “Time to collapse” (code feature)

For  $r < 0$ , the app reports an estimate of the time  $t_*$  until  $P(t)$  reaches roughly one person by solving  $1 = P_0 e^{rt_*}$ :

$$t_* = -\frac{\ln P_0}{r}.$$

This is not a literal prediction of extinction; it just quantifies the speed of exponential decline under constant  $r$ .

### 4.3 Algorithm in the app

```
# Deterministic Exponential
pops = P0 * np.exp(r * t) # t = 0,1,2,... years ahead
```

## 5 Model 2: Stochastic Itô (Geometric Brownian Motion)

### 5.1 Why a stochastic model?

Real populations fluctuate because TFR, mortality, and migration vary over time. To capture randomness, the app uses a standard continuous-time stochastic model:

$$dP_t = \mu P_t dt + \sigma P_t dW_t,$$

where  $W_t$  is a standard Wiener process (random noise). Here  $\mu$  is the average growth rate and  $\sigma$  is the volatility (how wiggly the path is). The app sets  $\mu = r$  from Section 3.

### 5.2 Exact simulation step for GBM

For a time step  $\Delta t = 1$  year, the exact update for GBM is

$$P_{t+\Delta t} = P_t \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z\right),$$

with  $Z \sim \mathcal{N}(0,1)$ . The app simulates many paths and plots the mean and the (mean  $\pm$  one standard deviation) band.

### 5.3 Setting the volatility from a TFR slider

The app offers a slider for the standard deviation of TFR, called `tfr_std`. It converts that to a per-year population volatility via

$$\sigma = \max\left(\frac{\text{tfr\_std}}{F \cdot G}, 0.02\right).$$

This is a pragmatic mapping: larger TFR uncertainty and shorter generation time both imply a larger year-to-year volatility in population growth. The floor 0.02 prevents trivial zero-volatility cases.

### 5.4 Algorithm in the app

```
# Stochastic GBM
paths = np.zeros((len(t), num_paths))
paths[0] = P0
for i in range(1, len(t)):
    dW = np.random.normal(0, np.sqrt(1), num_paths) # dt = 1 year
    paths[i] = paths[i-1] * np.exp((r - 0.5*sigma**2) * 1 + sigma * dW)
mean_pop = np.mean(paths, axis=1)
std_pop = np.std(paths, axis=1)
```

**Interpretation.** Because GBM is lognormal, the average path tends to exceed the median path when  $\sigma > 0$  (a known property of multiplicative noise). The shaded band is *not* a confidence interval for the mean; it is the empirical standard deviation of simulated populations at each time.

## 6 Model 3: Galton–Watson Branching (Generation-by-Generation)

### 6.1 Core idea

Imagine a generation  $g$  has  $N_g$  people. Each person has a random number of children; the expected number is  $m$ . In the app,

$$m = R_0 = \frac{F}{F_{\text{rep}}},$$

so  $m > 1$  means growth,  $m < 1$  means eventual extinction is likely in the true branching model.

### 6.2 Generation length and timeline

The model advances one generation at a time using the mean age at childbearing  $G$  as the generation length. If we project  $Y$  years ahead, the number of generations simulated is

$$\#\text{gens} = \left\lfloor \frac{Y}{G} \right\rfloor + 1.$$

Years are then labeled as 2023,  $2023 + G$ ,  $2023 + 2G$ , ...

### 6.3 Distributional approximation used in the code

If each person has a Poisson number of children with mean  $m$ , then the total children in generation  $g$  has mean  $N_{g-1}m$  and variance  $N_{g-1}m$ . For large populations, a normal approximation is reasonable:

$$N_g \approx \mathcal{N}(N_{g-1}m, N_{g-1}m).$$

The app draws from this normal distribution and clips negative values to zero (a practical safeguard).

### 6.4 Algorithm in the app

```
gen_pops = np.zeros((num_generations, num_sims))
gen_pops[0] = P0
for g in range(1, num_generations):
    mean = gen_pops[g-1] * m
    std = np.sqrt(gen_pops[g-1] * m)
    gen_pops[g] = np.random.normal(mean, std, num_sims)
    gen_pops[g] = np.maximum(gen_pops[g], 0)
```

#### Important notes.

- This is a *generation-based* model; migration is *not* added separately in the branching step (unlike the exponential/GBM drift where it was).
- In classical branching theory, the extinction probability is 1 when  $m \leq 1$ , and the smallest solution  $q \in (0, 1)$  to  $G(q) = q$  when  $m > 1$  (with  $G$  the offspring generating function). The app does not compute  $q$ ; it focuses on simulated means and spreads.

## 7 Scenarios and How They Modify Inputs

The app defines five scenario presets:

#### Base

Use the observed 2023 values as-is (replacement  $F_{\text{rep}} = 2.1$ ).

#### Low Fertility

Decrease  $F$  by 0.5 (floor at 0.5).

#### High Fertility

Increase  $F$  by 0.5.

#### Catastrophic

Set  $F = 0.8$ , cut  $P_0$  by half, raise replacement to  $F_{\text{rep}} = 2.3$  (reflecting harsher mortality), and halve the migration rate.

#### Custom

Sliders let you set  $F$ ,  $G$ ,  $P_0$  scaling,  $F_{\text{rep}}$ , and migration explicitly.

These feed into either  $r$  (for exponential/GBM) or  $m = R_0$  (for branching), and drive the resulting paths.

## 8 Putting It Together: Code-to-Math Map

Code symbol / step	Mathematical meaning
TPopulation1Jan * 1000	$P_0$ (initial population, people)
TFR $\rightarrow F$	Total fertility rate
MAC $\rightarrow G$	Generation time (years)
NetMigrationRate/1000 $\rightarrow m$	Per-year migration fraction
R0 = F / replacement	$R_0$
$r = \ln(R_0)/G + m$	Drift for exponential/GBM
GBM: $\exp((r - 0.5\sigma^2) + \sigma Z)$	Exact one-year lognormal step
Branching: Normal(mean, var)	$N_g \sim \mathcal{N}(N_{g-1}m, N_{g-1}m)$

## 9 Interpretation Tips (High-School Intuition)

- **Exponential (deterministic)** is like steady compound interest. A fixed annual rate  $r$  makes the population curve a smooth exponential.
- **GBM (stochastic)** is compound interest that wiggles each year. Sometimes growth is a bit better, sometimes worse, with average  $r$  and randomness  $\sigma$ .
- **Branching (generation-by-generation)** thinks in steps of  $G$  years. Each generation is a random multiple of the previous one, on average multiplied by  $m = R_0$ .
- **Migration** is an extra plus or minus percentage per year in the drift of the first two models.
- **Replacement** ( $\approx 2.1$ ) marks where births per woman keep the population steady in the long run (ignoring migration). Below that, generations tend to shrink.

## 10 Model Limitations and Design Choices

- (a) **Generation time  $\approx$  MAC.** Using MAC for  $G$  is a simplification; true generation time depends on full age-specific fertility and survival schedules.
- (b) **Single-number fertility.** TFR compresses an age-pattern into one number; real dynamics need age-structured models (Leslie matrices) for precision.
- (c) **Migration as drift.** The app treats net migration as a smooth fractional addition. Shocks (e.g., wars) would require time-varying  $m$  or state changes.
- (d) **Branching normal approximation.** The code uses a normal approximation to a Poisson sum. This is accurate for large  $N$ , but can be rough for small populations.
- (e) **Volatility mapping.** The  $\sigma$  formula is heuristic. It anchors randomness to TFR uncertainty and generation time to give a tunable, interpretable knob.
- (f) **UI default note.** The in-code comment recommends the stochastic Itô model for global projections, but the dropdown's default index actually selects the deterministic option when "World" is chosen; this can be adjusted in the code if desired.

## 11 Worked Micro-Example (Numbers Picked for Illustration)

Suppose for a country we have  $P_0 = 50$  million,  $F = 1.8$ ,  $F_{\text{rep}} = 2.1$ ,  $G = 30$  years, and net migration rate +3 per 1000 per year. Then

$$R_0 = \frac{1.8}{2.1} \approx 0.857, \quad r_{\text{fert}} = \frac{\ln(0.857)}{30} \approx -0.0052 \text{ yr}^{-1}, \quad m = \frac{3}{1000} = 0.003.$$

So the net drift is  $r \approx -0.0052 + 0.003 = -0.0022$  (about  $-0.22\%$  per year). The deterministic projection would be  $P(t) = 50 e^{-0.0022t}$  million. If you pick  $\sigma = 0.03$  in GBM, the mean path still follows roughly this drift, but with visible wiggles and a growing uncertainty band over time.

## 12 Reproducibility Checklist

- Use WPP 2024 data filtered to 2023, `Variant = Estimates`, and `Type`  $\in$  {Country/Area, World}.
- Compute  $P_0$ ,  $F$ ,  $G$ ,  $m$  as described; choose scenario presets or custom sliders.
- For exponential / GBM, set  $r = \ln(F/F_{\text{rep}})/G + m$ ; for GBM, choose  $\sigma = \max(\text{tfr\_std}/(FG), 0.02)$ .
- For branching, use  $m = F/F_{\text{rep}}$  and advance in steps of  $G$  years with the normal approximation.
- Plot either  $P(t)$  (deterministic) or the mean  $\pm$  one standard deviation across simulated paths (stochastic).

### A Deriving $r = \ln(R_0)/G$

Suppose each generation multiplies population by  $R_0$  over  $G$  years. A smooth exponential path  $P(t) = P_0 e^{rt}$  must satisfy  $P(G) = P_0 R_0$ , hence  $e^{rG} = R_0$  and  $r = \ln(R_0)/G$ .

### B Closed-Form Step for GBM

For  $dP_t = \mu P_t dt + \sigma P_t dW_t$ , Itô's lemma gives  $d(\ln P_t) = (\mu - \frac{1}{2}\sigma^2) dt + \sigma dW_t$ . Integrating over  $\Delta t = 1$ ,

$$\ln P_{t+1} - \ln P_t = (\mu - \frac{1}{2}\sigma^2) \cdot 1 + \sigma Z, \quad Z \sim \mathcal{N}(0, 1),$$

so  $P_{t+1} = P_t \exp((\mu - \frac{1}{2}\sigma^2) + \sigma Z)$  as used in the app.