Codebook for:
The economics of skyscrapers: A synthesis *

Gabriel M. Ahlfeldt† Jason Barr‡

May 4, 2024

Abstract

This codebook is part of the toolkit for Ahlfeldt and Barr Wolf (2022): The economics of skyscrapers: A synthesis, *Journal of Urban Economics*, 129. The toolkit does not cover all stages of the analysis presented in the article. Instead, it covers a subset of codes that are crucial for the quantification and simulation of the model. The codebook summarizes the primitives and endogenous objects of the model and introduces selected numerical algorithms in pseudo-code. The focus is on algorithms that are essential for the quantification and simulation of the respective quantitative models.

*We thank Andrea Herrera for comments and suggestions.
†Humboldt University. Email: g.ahlfeldt@hu-berlin.de
‡Rutgers University. Email: jmbarr@newark.rutgers.edu

A Ahlfeldt and Barr (2022)

This section covers *Ahlfeldt and Barr (2022): The economics of skyscrapers: A synthesis. Journal of Urban Economics, 129*, https://doi.org/10.1016/j.jue.2021.103419. The model consists of exogenous parameters, fundamentals, and endowments as well as endogenous objects tablulated in Table 1.

A.1 Equilibrium

Given the model's exogenous parameters $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \tilde{c}^U, r^a, \bar{U}\}$ and the exogenous fundamentals $\{\bar{a}^U, \bar{S}^U\}$, the general equilibrium of the model is referenced by $\{a^U, \tilde{S}^U, r^U, S^U, \bar{p}^U, n(x), L(x), y\}$, where $U \in C, R$ indexes commercial (C) and residential (R). These 13 components of the equilibrium are determined by the following 13 equations:

1. Production amenity: $a^C(x) = \tilde{A}^C(x)^{\frac{1}{1-\alpha^C}}(y^C)^{\frac{\alpha^C}{1-\alpha^C}}$
2. Residential amenity: $a^R(x) = \tilde{A}^R(x)^{\frac{1}{1-\alpha^R}}(y^R)^{\frac{1-\alpha^R}{\alpha^R}}$
3. Constrained commercial height: $\tilde{S}^C(x) = \min\left(\left(\frac{a^C}{c^C(1+\theta^C)}\right)^{\frac{\theta^C}{\theta^C-\omega^C}}, \bar{S}^C\right)$
4. Constrained residential height: $\tilde{S}^R(x) = \min\left(\left(\frac{a^R}{c^R(1+\theta^R)}\right)^{\frac{\theta^R}{\theta^R-\omega^R}}, \bar{S}^R\right)$
5. Commercial land rent: $r^C(x) = \frac{a^C}{1+\omega^C}(\tilde{S}^C)^{1+\omega^C} - c^C(\tilde{S}^C)^{1+\theta^C}$
6. Residential land rent: $r^R(x) = \frac{a^R}{1+\omega^R}(\tilde{S}^R)^{1+\omega^R} - c^R(\tilde{S}^R)^{1+\theta^R}$
7. Realized commercial height: $S^C(x) = \tilde{S}^C(x),\ S^R(x) = 0,\ \text{if } r^C(x) \geq r^R(x),\ r^C \geq r^A$
8. Realized residential height: $S^R(x) = \tilde{S}^R(x),\ S^C(x) = 0,\ \text{if } r^R(x) > r^C(x),\ r^R \geq r^A$
9. Horizontal commercial rent: $\bar{p}^C(x) = \frac{a^C(x)}{1+\omega^C}S^C(x)^{\omega^C}$
10. Horizontal residential rent: $\bar{p}^R(x) = \frac{a^R(x)}{1+\omega^R}S^R(x)^{\omega^R}$
11. Workplace employment: $L(x) = \frac{\alpha^C}{1-\alpha^C}\frac{\bar{p}^{-C}(x)}{y^C}S^C(x)$
12. Residence employment: $n(x) = \frac{S^R(x)}{y^R}\frac{\bar{p}^{-R}(x)}{1-\alpha^R}$
13. Labour market clearing: $\int_{-x_0}^{x_0} L(x)dx = \int_{-x_1}^{x_0} n(x)dx + \int_{x_0}^{x_1} n(x)dx = N$

Table 1: Codebook for AB2022

Object	Description
	Structural parameters
α^U	Share of non-floor space input (C) or expenditure (R)
β^U	Agglomeration elasticity
τ^U	Amenity decay
$\tilde{\omega}^U$	Height elasticity of production/residential amenity
ω^U	Height elasticity of rent: $\omega^U = \frac{\tilde{\omega}^U}{1-\alpha^U}$
θ^U	Height elasticity of construction cost
c^U	Baseline construction cost
$\bar{a}^U$	Fundamental production/residential amenity
$\tilde{a}^U$	Location specific component in production/residential amenity
	Exogenous characteristics
r_a	Agricultural land rent
$\bar{S}^U$	Height limit
	Endogenous variables and scalars
y	Wage
x_0	Outer margin of commercial zone
x_1	Outer margin of residential zone
$L(x)$	Labour input
$n(x)$	Labour supply
$\tilde{A}^C(x)$	Productivity shifter: $\tilde{A}^C(x) = \bar{a}^C\tilde{a}^C N^{\beta^C} e^{-\tau^C\lvert x\rvert}$
$\tilde{A}^R(x)$	Amenity shifter: $\tilde{A}^R(x) = \bar{a}^R\tilde{a}^R N^{\beta^R} e^{-\tau^R\lvert x\rvert}$
$a^C(x)$	Commercial floor space price shifter: $a^C(x) = \tilde{A}^R(x)\left(\frac{1}{1-\alpha^R}\right)\left(y^R\right)^{\frac{1}{1-\alpha^R}}$
$a^R(x)$	Residential floor space price shifter: $a^R(x) = \tilde{A}^C(x)\left(\frac{1}{1-\alpha^C}\right)\left(y^C\right)^{\frac{\alpha^C}{\alpha^C-1}}$
$r(x)$	Land rent
$\bar{p}^U(x)$	Floor space rent
$S^{*U}(x)$	Profit-maximizing building height: $S^{*U}(x) = \left(\frac{a^U}{c^U(1+\theta^U)}\right)^{\frac{1}{\theta^U-\omega^U}}$
$\tilde{S}^U(x)$	Constrained height choice by the developer
$S^U(x)$	Actual building height conditional on the land use allocation
$\bar{f}^R$	Residential horizontal floor space per capita from the Marshallian demand

A.2 Algorithms

In this section, we introduce algorithms used by AB to simulate the theoretical model and invert structural fundamentals that match the Chicago skyline. We add some algorithms that allow matching the city's population for didactic purposes. We abstract from the empirical analysis of data. We refer to objects whose values we solve numerically as target objects. Within the solver, we refer to guessed values when values of target objects are input into the construction of other objects. We refer to predicted values when the values of target objects are computed as functions of other objects (which typically depend on guesses). Typically, we solve for the values of target objects using iterative procedures in which we update our guesses to weighted combinations of guessed and predicted values.

A.2.1 Stylized city structure

In the algorithmic implementation of the equilibrium solver, we use the above system of equations and treat wage and population $\{y, N\}$ as target objects. Once we have the values for our targets, we can use the recursive structure of the above equations to solve for all endogenous objects, conditional on given primitives. This is implemented in Algorithm 1. We use Algorithm 1 in Algorithm 3 to solve for all other endogenous objects starting from guesses of the target objects $\{y, N\}$. We then aggregate local labour demand $L(x)$ and labour supply $n(x)$ to the city level. We use aggregate labour demand and supply in Algorithm 2 to compute the market-clearing wage and equilibrium employment, which we use to update our guesses of the target objects $\{y, N\}$. Algorithm 3 iterates until our guesses do not change any more. By default, Algorithm 3 returns a stylized city with smooth monotonic gradients emerging from the center. This is because the location-specific component of fundamental amenities (a_i^U) are set to a uniform unit value.

Algorithm 1: Solver for endogenous variables: `SOLVER`

Data: Given values for primitives $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \bar{a}^U, \tilde{c}^U, \bar{S}^U, r^a, \tilde{U}\}$
Given values of target objects wage and population $\{y, N\}$

Compute $\bar{p}^U(x)$ using bid-rent eqs.
Compute $\tilde{S}^U(x)$ using profit-maximizing building height eq.
Compute $r^U(x)$ using land bid rent eq.
Allocate land to use with the highest land rent
Compute local workplace employment $L(x)$ using MRS eq. within commercial zone
Compute local residence employment $n(x)$ using Marshallian demand eq. within residential zone
Compute labour demand (total L within commercial zone)
Labour supply (total $\hat{N}$) within residential zone

Result: $L(x), n(x), \bar{p}^U(x), r^U(x), S^U(x), L, \hat{N}$

Algorithm 2: Finding equilibrium wage: `WAGE`

Data: Given values for primitives $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \bar{a}^U, \tilde{c}^U, \bar{S}^U, r^a, \tilde{U}\}$
Given values for labour supply and labour demand $\{L, \hat{N}\}$
Guess of wage y
Algorithm 1

while $L \neq \hat{N}$ **do**
- Update y to $\hat{y} = y \times \left(\frac{\hat{N}}{L}\right)^{\rho>0}$
- Use Algorithm 1 to local workplace and residence employment
- Update labour demand (L) within commercial zone
- Update labour supply $(\hat{N})$ within residential zone

Result: Updated values $\{\hat{y}, L, \hat{N}\}$

Algorithm 3: Finding the equilibrium: `FINDEQ`

Data: Given values for primitives $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \bar{a}^U, \tilde{c}^U, \bar{S}^U, r^a, \tilde{U}\}$
Guesses of wage and employment $\{y, N\}$
Algorithm 1
Algorithm 2

Solve endogenous variables for guesses of $\{y, N\}$ using Algorithm 1
while N *changes* **do**
- Use Algorithm 2 to clear labour market and optain new $\{\hat{y}, N\}$
- Update guesses to weighted combination of old guess of $\{N\}$ and updated value $\{N\}$

Result: Equilibrium values of $L(x), n(x), \bar{p}^U(x), r^U(x), S^U(x), y, N$

A.2.2 Fuzzy gradients

The same solvers introduced in A.2.1 can be used in a setting with more realistic fuzzy height gradients, simply by changing the values of the location-specific component a_i^U. If a_i^U values are non-monotonic in distance from the center, endogenous outcomes such as rents, heights, and densities will also change non-monotonically. Therefore, we index all variables by i in this section.

It is possible to invert the location-specific component a_i^U to rationalize a distribution of heights. Similarly, we can scale amenities to target a desired total population. Intuitively, a higher level of residential amenities attracts residents until congestion of the housing market and higher house prices restore the reservation utility level. Algorithm 6 implements an iterative procedure to create a city with a given height profile and population. First, it iteratively updates amenities using Algorithm 4 until there is a perfect correlation between the height profile in the model and the data. Then it iteratively scales residential amenities using Algorithm 5 until the population matches the target.

The inverted amenities can be used as a starting point for counterfactuals. Changing any of the primitives conditional on the inverted fundamentals, delivers counterfactual solutions, just like in the case with the stylized city structure.

Algorithm 4: Updating amenities to match heights `CONV`

Data: Given values for primitives $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \bar{a}^U, \tilde{c}^U, \bar{S}^U, r^a, \tilde{U}\}$
Given values of wage and population $\{y, N\}$
Given values of model-generated heights, S_i^U
Observed heights, H_i^U
Algorithm 3

1 Adjust $\bar{a}_i^U$ using a function of the adjustment factor $\frac{H_i^U}{S_i^U}$

2 Solve for endogenous objects using Algorithm 3

Result: Updated values of amenities a_i^U and equilibrium values of $L_i, n_i, \bar{p}_i^U, r_i^U, S_i^U, y, N$

Algorithm 5: Updating amenities to match total employment `EMP`

Data: Given values for primitives $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \bar{a}^U, \tilde{c}^U, \bar{S}^U, r^a, \tilde{U}\}$
Given values of wage and population $\{y, N\}$
User-specified target population, $\bar{N}$
Algorithm 3

1 Adjust $\bar{a}_i{}^R$ using a function of the adjustment factor $\frac{\bar{N}}{N}$

2 Solve for endogenous objects using Algorithm 3

Result: Updated values of amenities a_i^U and equilibrium values of $L_i, n_i, \bar{p}_i^U, r_i^U, S_i^U, y, N$

Algorithm 6: Inverting amenities a_i^U: `INVERT`

Data: Given values for primitives $\{\alpha^U, \beta, \omega^U, \theta^U, \tau^U, \bar{a}^U, \tilde{c}^U, \bar{S}^U, r^a, \tilde{U}\}$
Given values of wage and population $\{y, N\}$
Given values of model-generated heights, S_i^U
Observed heights, H_i^U
User-specified target employment, $\bar{N}$
Algorithm 4
Algorithm 5

while $Corr(S_i^U, H_i) \neq 1$ **do**
  Use Algorithm 4 to update a_i^U and obtain new S_i^U
while $N \neq \bar{N}$ **do**
  Use Algorithm 5 to update a_i^U and obtain new N

Result: Updated values of amenities a_i^U and equilibrium values of $L(x), n(x), \bar{p}^U(x), r^U(x), S^U(x), y, N$