

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

УДК 57.087

Отчет об исследовательском проекте на тему:
**Аналитическая и программная работа над пакетом по проведению
стохастических биологических симуляций**

Выполнила:

студентка группы БПМИ219
Лебедюк Ева Владимировна


(подпись)

24.05.2023
(дата)

Принял руководитель проекта:

Алексей Антонович Никитин
Доцент кафедры
общей математики МГУ им. М. В. Ломоносова


(подпись)

24.05.2023
(дата)

Содержание

Аннотация	3
Введение	4
Основные термины и обозначения	4
Предметная область	4
Цель исследования	5
Обзор литературы	6
1 Симуляции	7
1.1 Одномерные симуляции	7
1.2 Трехмерные симуляции	7
1.3 Примеры запуска симуляций	8
2 Сглаживания	10
2.1 Метод скользящего среднего	10
2.2 Метод простого экспоненциального сглаживания	10
2.3 Метод последовательного экспоненциального сглаживания	11
2.4 Сглаживания трехмерных симуляций	11
3 Поиск точки останова	14
3.0.1 Подбор размера окна	15
3.0.2 Подбор порогового значения	16
3.0.3 Результаты подбора параметров	16
Выводы	19
Список литературы	20
Приложения	21

Аннотация

В сфере биологии для исследования различных сообществ важную роль играет компьютерное моделирование, отдельным видом которого являются моделирование популяционной динамики.

В частности, проведение симуляций может быть полезно при исследовании систем долгоживущих неподвижных организмов, например деревьев, чью популяционную динамику в реальности проблематично отследить за относительно краткий период времени.

Также известно [1], что модели, подразумевающие пространственную однородность популяции хуже приближают реальность, чем модели, отдельно учитывающие поведение каждого из индивидов.

Поэтому объектом исследования моей работы будет модель второго типа, а именно модель Ульфа Дикманна и Ричарда Лоу.

Ключевые слова

Стохастические симуляции биологических моделей, трехмерные симуляции, сглаживания графиков, выход на плато, модель Дикманна-Лоу

Введение

Основные термины и обозначения

- **Темп рождаемости** - константа $b \in \mathbb{R}, b \geq 0$
- **Темп смертности естественной среды** - константа $d \in \mathbb{R}, d \geq 0$
- **Темп смертности от конкуренции** - константа $d' \in \mathbb{R}, d' \geq 0$
- **Ядро рождаемости** - это радиально-симметричная функция $m(x)$, такая что:

$$\begin{aligned}m(x) &\geq 0 \\ \int_{\mathbb{R}} m(x) dx &= 1 \\ \lim_{|x| \rightarrow \inf} m(x) &= 0 \\ \forall x, y \in \mathbb{R} : ||x||_2 = ||y||_2 &\Rightarrow m(x) = m(y)\end{aligned}$$

- **Ядро смертности** - это радиально-симметричная функция $w(x)$, такая что:

$$\begin{aligned}w(x) &\geq 0 \\ \int_{\mathbb{R}} w(x) dx &= 1 \\ \lim_{|x| \rightarrow \inf} w(x) &= 0 \\ \forall x, y \in \mathbb{R} : ||x||_2 = ||y||_2 &\Rightarrow w(x) = w(y)\end{aligned}$$

Предметная область

Модель Ульфа Дикманна и Ричарда Лоу, чье преимущество над аналогами состоит в том, что она учитывает положение каждого из неподвижных индивидов в пространстве, их влияние друг на друга и на популяцию в целом. Данная модель задается параметрами $d, d', b, m(x), w(x)$ и описывает популяцию растений на ограниченной области $A \subset \mathbb{R}^n, n \in \{1, 2, 3\}$. Все растения будут считаться идентичными материальными точками, отличающимися только своим положением в пространстве

Цель исследования

Обычно, чтобы добиться интересующего нас состояния равновесия модели, необходимы часы, а порой (при некоторых конфигурациях биологически параметров), даже дни для запуска достаточного количества событий Пуассоновского процесса.

Поэтому для минимизации времени их работы предлагаться останавливать работу симуляций, если был совершен выход на плато.

Подбор параметров для выхода на плато многомерных симуляций, и рассмотрение их графиков были моими основными задачами

Обзор литературы

Перед началом работы с моделью Дикманна-Лоу, я изучила и другие биологические модели. В частности, в статье [2] описана простая линейная модель процесса рождения-смерти $X_0(\cdot)$, зависящая только от коэффициентов рождаемости b и смертности d , а затем ее естественное дополнение $X_\theta(\cdot)$ – добавление “лишних” смертей от конкуренции d' .

После этого по работам [1], [3], [6] я ознакомилась с параметрами целевой модели, а так же с ее компьютерной реализацией, лежащей в репозитории [5].

Затем я генерировала 2d симуляции, чтобы сравнить их результаты с полученными численным методом на параметрах из [1].

Наконец по работам [7], [4] участников научной группы прошлых лет, я адаптировала выработанные ими на одномерных симуляциях алгоритмы на трехмерные.

1 Симуляции

Алгоритм симуляций основан на Пуассоновском процессе [3]:

- Рассматривается поведение особей в n-мерном кубе.
- Граничные условия определяют, что произойдет при рождении особи вне границы области. В случае убывающих границ, событие за пределами куба не происходит, в случае периодических - событие происходит с другой стороны.
- События рождения и смерти - пуассоновские потоки требований с фиксированными $\lambda = b_k, d_k$ - рождаемостью и смертностью для k-ого вида
- Аддитивное свойство Пуассоновских потоков позволяет производить симуляцию по одному событию за раз для каждой из имеющихся особей.
- Полный просчет всех одновременных событий называется **эпохой**

1.1 Одномерные симуляции

Задаются параметрами, соответствующими основным обозначениям:

Таблица 1.1: Параметры одновидовых симуляций

обозначение	параметр
b	b
d	d
d'	dd
$m(x)$	sd_b
$w(x)$	sd_d
Максимальный радиус взаимодействия	death_r
Координаты начальных особей по x	initial_pop_x
Размер пространства по x	area_length_x

В научной группе для таких симуляций были найдены оптимальное сглаживание [7], а также был написан алгоритм [4] поиска точки останова/выхода на плато (момент, когда численность особей перестает расти и начинает колебаться около некоторого значения).

1.2 Трехмерные симуляции

Моей же задачей стало повторить результаты научной группы, но для многомерных симуляций, в частности, трехмерных.

Для этого в таблицу 1.1 добавляются следующие параметры:

Таблица 1.2: Параметры одновидовых симуляций.

обозначение	параметр
Размерность пространства	ndim = 3
Координаты начальных особей по y	initial_pop_y
Координаты начальных особей по z	initial_pop_z
Размер пространства по y	area_length_y
Размер пространства по z	area_length_z

Так как кубическое пространство растет намного быстрее линейного, при запуске симуляций я задавала размер области так, чтобы длина любой из ее сторон не превышала 40, а dd наоборот увеличила, так как этот параметр обратно пропорционален плотности особей. Без этих решений выход на плато мною не наблюдался.

1.3 Примеры запуска симуляций

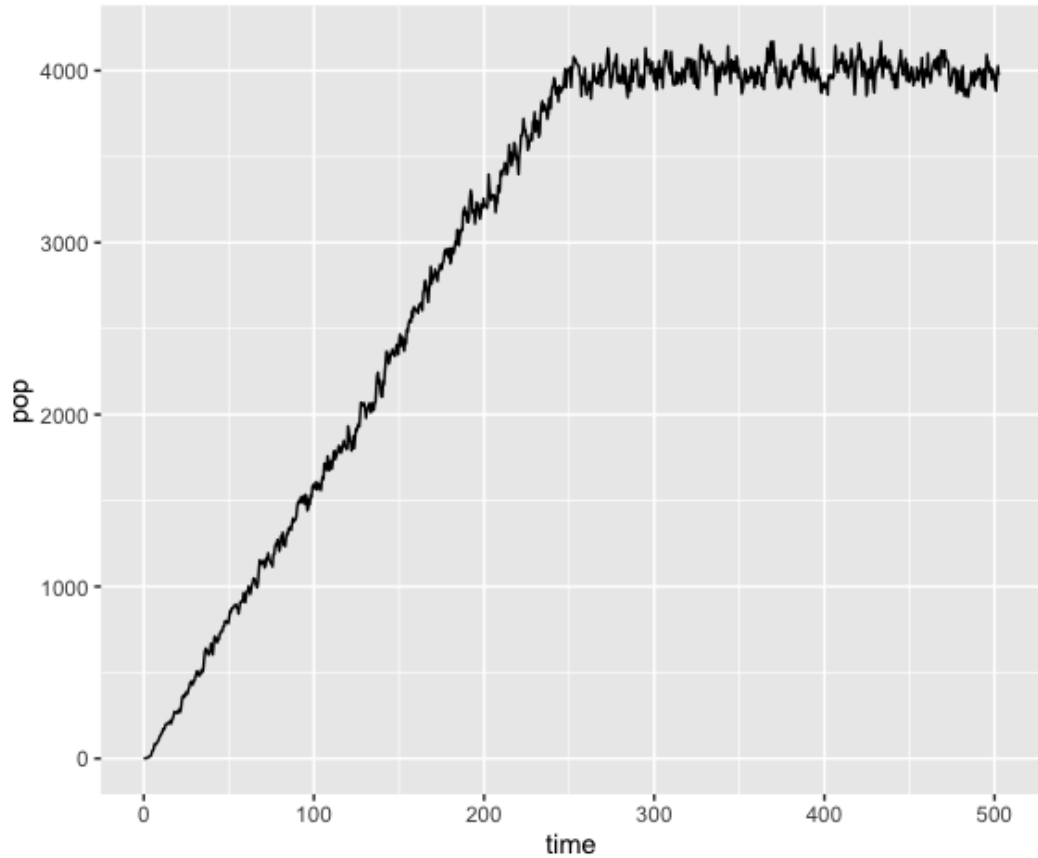


Рис. 1.1: График численность/время 1d симуляций
 Параметры запуска: $area_length_x = 400$,
 $b = 1$, $d = 0$, $dd = 0.1$, $initial_pop = 1$, $sd_d = sd_d = 1$, $death_r = 3$
 Количество эпох: 1000

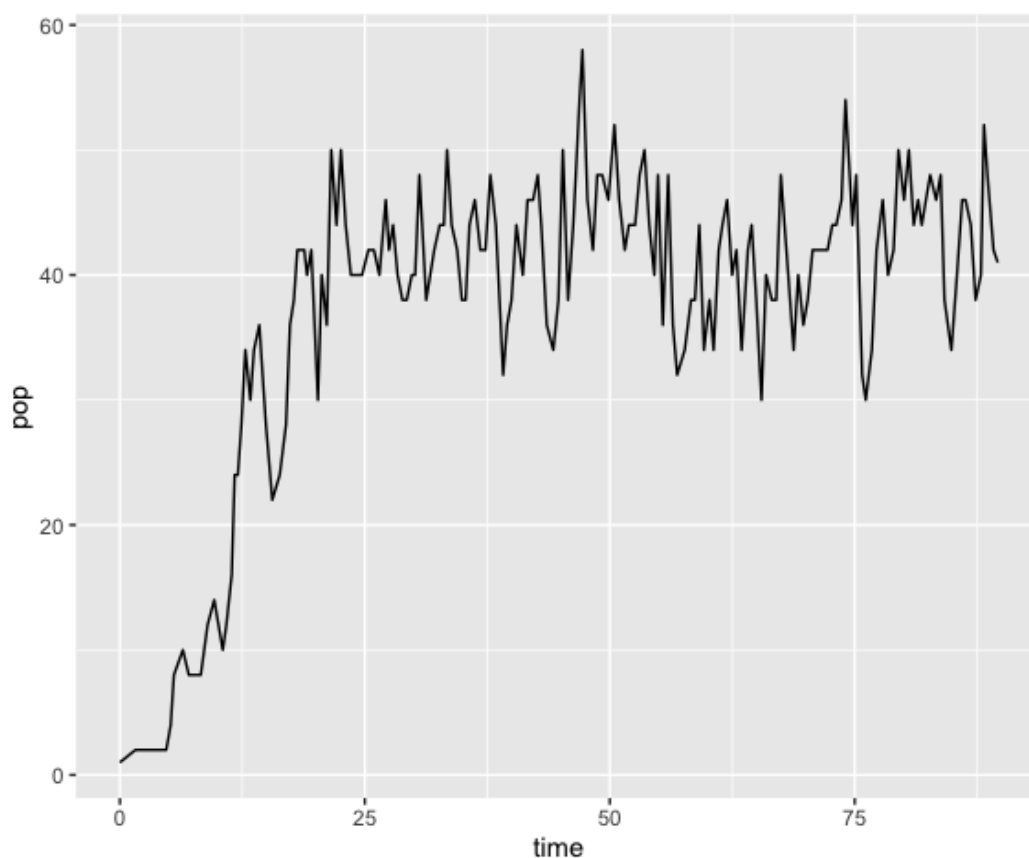


Рис. 1.2: График численность/время 3d симуляций
 Параметры запуска: $\text{area_length_x} = \text{area_length_y} = \text{area_length_z} = 10$,
 $b = 1$, $d = 0$, $dd = 25$, $\text{initial_pop} = 1$, $\text{sd_d} = \text{sd_d} = 1$, $\text{death_r} = 3$
 Количество эпох: 25000000

Заметим, что я задала во много раз большее количество эпох в трехмерном случае, чем в одномерном, так как события в многомерном случае происходят медленнее, и часто превышают лимит времени на одну эпоху.

Это надо будет учесть при работе с симуляциями в будущем, так как даже при схожих прочих параметрах в 3d пространстве популяция будет расти медленнее, чем в 1d.

2 Сглаживания

Сглаживание графика является важной частью в задаче нахождения плато. Человеческому взгляду явно заметен тренд, но найденный коллегами алгоритм может выдавать неверные значения, если этот шум на участке роста достаточно велик. Поэтому применение сглаживаний очень важно, особенно для алгоритмов, работающих в реальном времени.

В процессе исследования мною были рассмотрены следующие методы сглаживания:

2.1 Метод скользящего среднего

Самый примитивный вариант.

В результате этого алгоритма очередной член нового ряда $\{\hat{y}_t\}$ получается усреднением последних k членов старого $\{y_t\}$:

$$\hat{y}_t = \frac{1}{k} \sum_{i=0}^{k-1} y_{t-i}$$

По очевидным причинам, этот метод сложно реализовать без предподсчета, поэтому для дальнейшего поиска плато этот вариант не подходит.

К тому же, по рисунку 2.1 видно, что неверный выбор k может привести к ухудшению даже по сравнению с начальными результатами.

2.2 Метод простого экспоненциального сглаживания

В результате этого алгоритма очередной член нового ряда $\{\hat{y}_t\}$ получается из текущего значения изначального ряда $\{y_t\}$, а также всех предыдущих членов $\{\hat{y}_t\}$, входящих с затухающим значением, задаваемым $\alpha < 1$:

$$\hat{y}_t = \alpha y_t + (1 - \alpha) \hat{y}_{t-1}$$

Параметр α при этом можно изменять.

В работе [4] заметили, что этот метод, не только лучше приближает ряд, но и реализуем в реальном времени. Также коллеги получили, что для 1d симуляций параметр $\alpha = 0.25$ наиболее оптимален, но по моим наблюдениям 2.2, для 3d лучше подходит $\alpha = 0.1$.

2.3 Метод последовательного экспоненциального сглаживания

Логическим дополнением предыдущего метода является метод последовательного экспоненциального сглаживания.

Этот метод применяет предыдущий n раз, каждый раз к новому $\{\hat{y}_t^i\}$, $i \in \{1 \dots n\}$.

Формула получения \hat{y}_t^n :

$$\hat{y}_t^n = \begin{cases} \alpha y_t + (1 - \alpha)\hat{y}_{t-1}^n & n = 1 \\ \alpha y_t^{n-1} + (1 - \alpha)\hat{y}_{t-1}^n & n > 1 \end{cases}$$

Так как $\alpha = 0.1$ — лучший параметр по результатам предыдущего шага, это же значение буду использовать далее.

Результаты применения метода с $n=10$ представлены на рисунке 2.3. Можно заметить, что последние несколько графиков визуальнo одинаковы, значит в дальнейшем можно сократить n до 5.

2.4 Сглаживания трехмерных симуляций

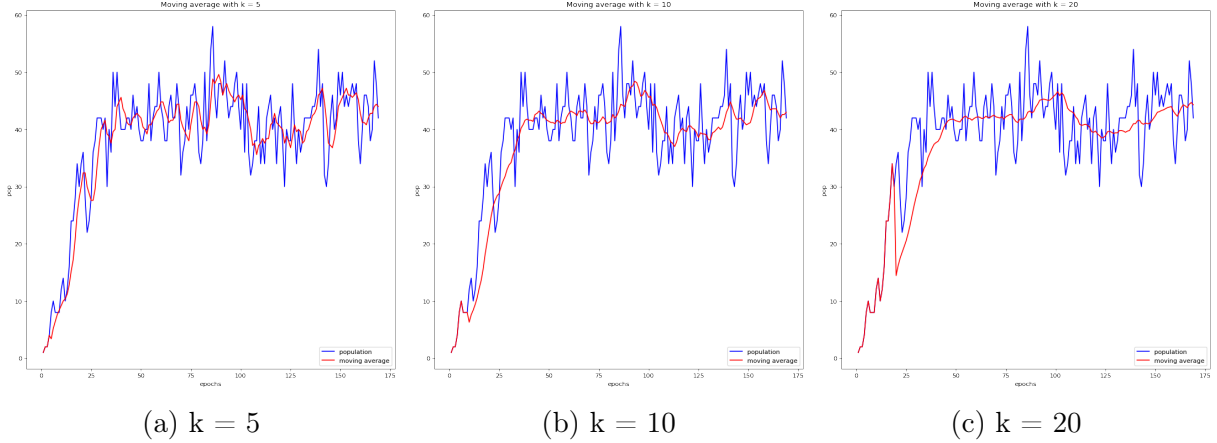
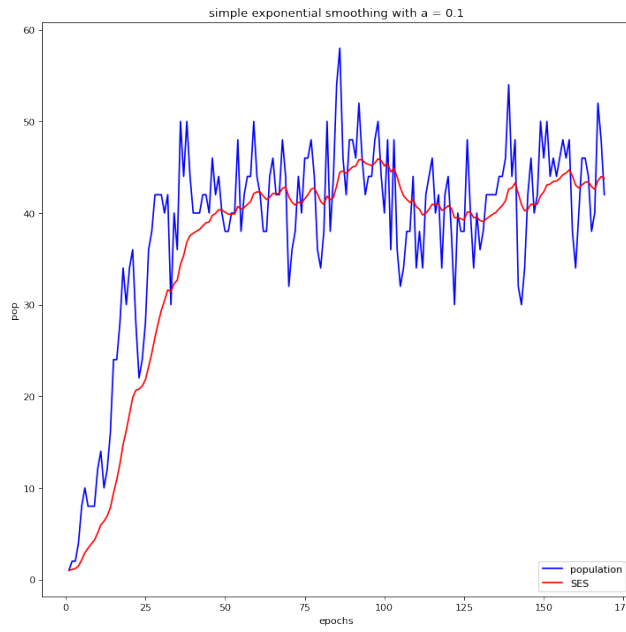
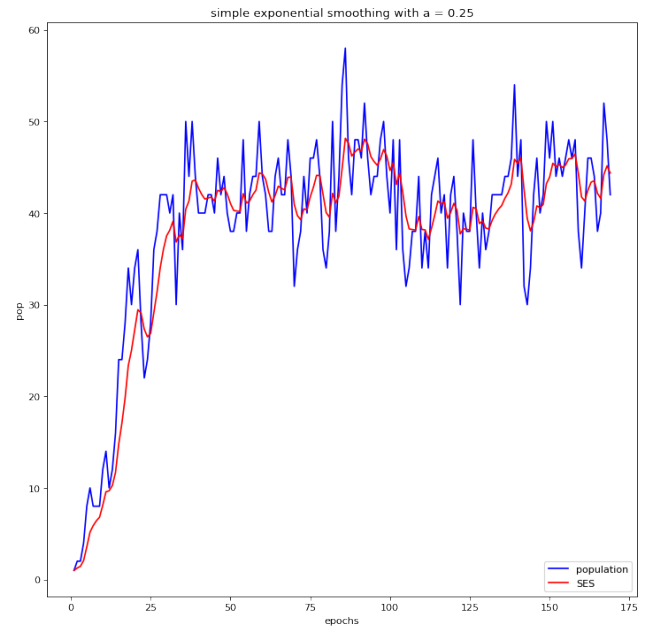


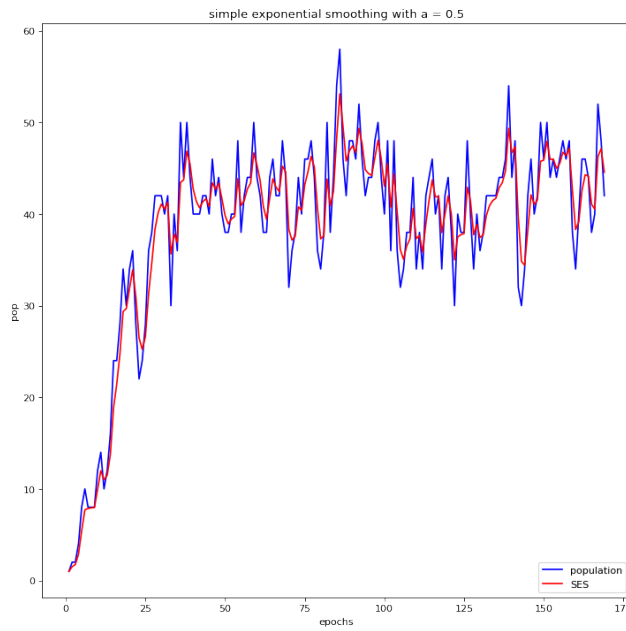
Рис. 2.1: Графики применения метода скользящего окна с различными k



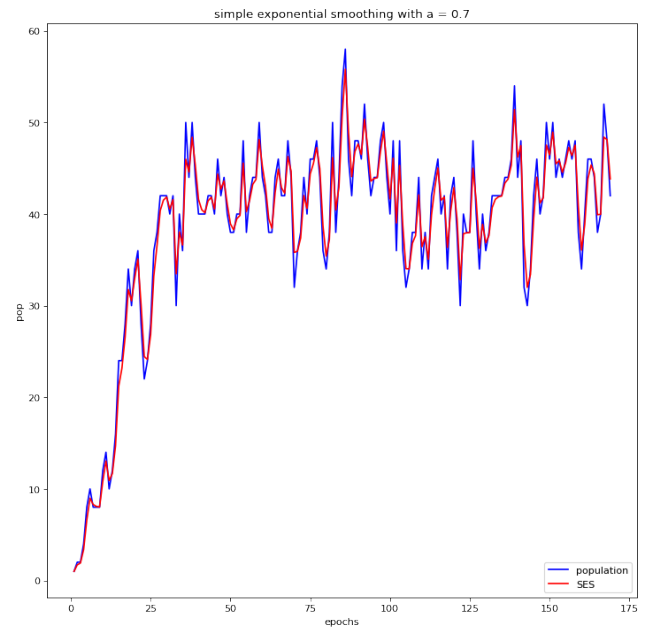
(a) $\alpha = 0.1$



(b) $\alpha = 0.25$



(c) $\alpha = 0.5$



(d) $\alpha = 0.7$

Рис. 2.2: Графики применения простого экспоненциального сглаживания с различными α

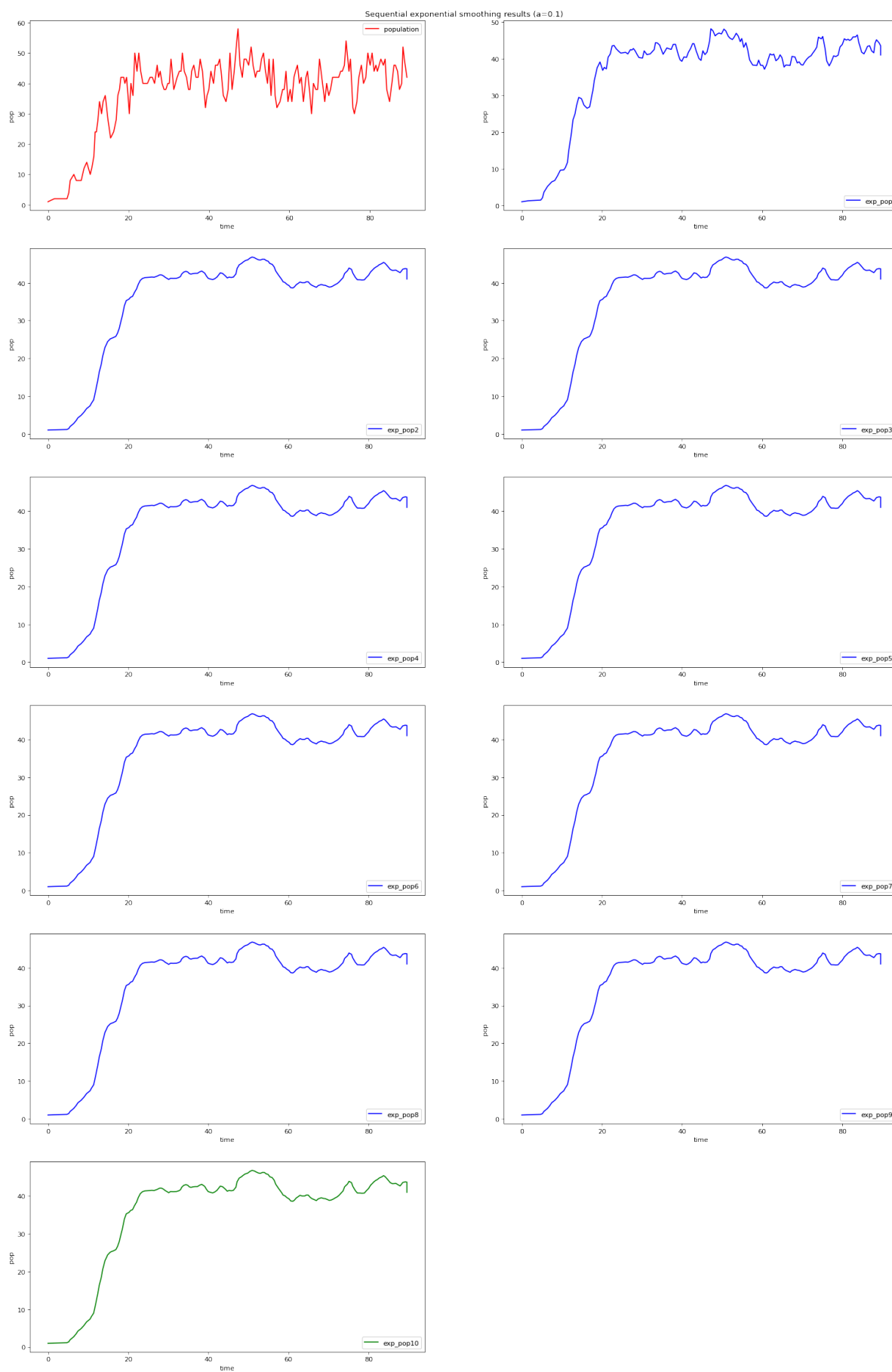
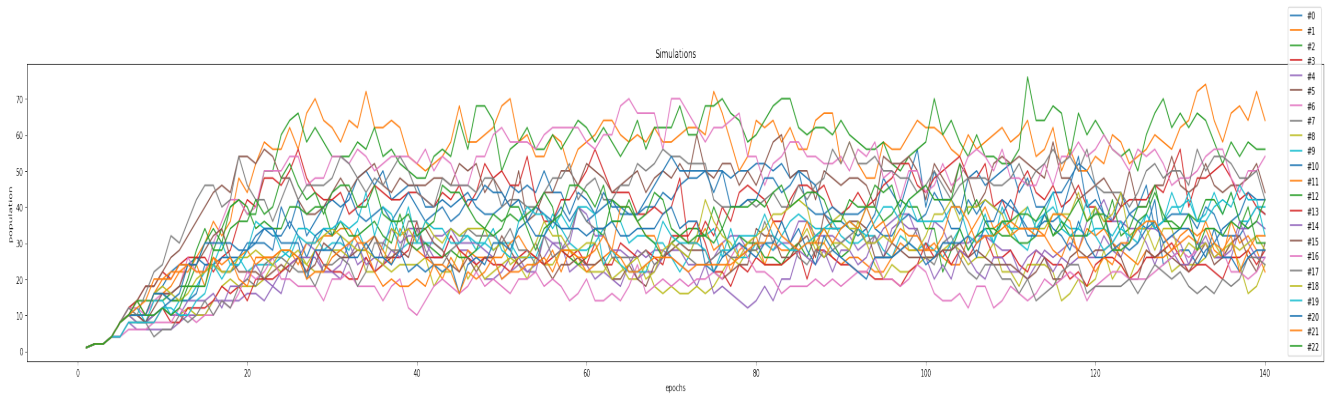
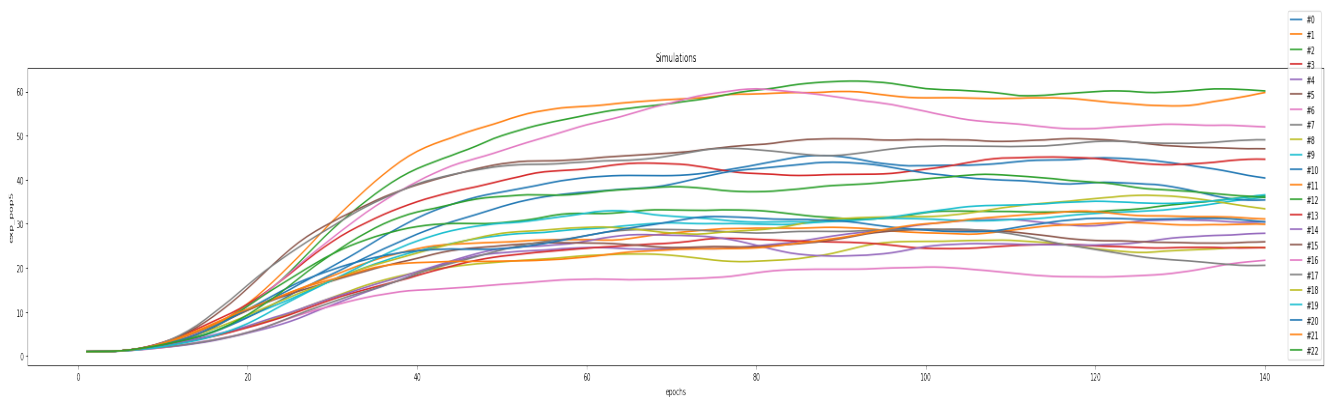


Рис. 2.3: Графики применения последовательного экспоненциального сглаживания

3 Поиск точки останова



(a) Без сглаживания



(b) Со сглаживанием (эксп, $n = 5$)

Рис. 3.1: Графики из датасета для поиска параметров выхода на плато

По выше представленным графикам заметим, что их можно разделить на три части:

- Участок подъёма
- Колено (перегиб)
- Само плато

Алгоритм нахождения плато в реальном времени заключается в сравнение дисперсий соседних участков графика. В результате два участка, разница дисперсий у которых максимальна – искомая точка останова.

Очевидно что в реальном времени, сравнивать все со всем – не оптимальная опция, поэтому считаем плато найденным, если дисперсия двух участков стала больше некоторого порогового значения.

Подбор этого значения, а также значения движущегося окна – еще одна стоящая передо мной задача. С последним легче: оно не может быть слишком маленькими, так как в этом случае сильный шум, не устраненный по каким-либо причинам сглаживаниями, не

даст алгоритму дойти до плеча, а слишком большое отнимает слишком много ресурсов на построение графика, для которого выход на плато давно произошел. На основе исследований, иллюстрируемых 3.2 и вышеописанным фактом, что в трехмерном случае сложно достичь задаваемого большого количества эпох без того, чтобы хотя бы некоторые из них не превышали лимит времени симуляций, предлагается использовать окна размером 15 и 45 эпох.

Проверка окон каждого размера происходит независимо, то есть симуляции будут останавливаться при достижении порога на любом из них

3.0.1 Подбор размера окна

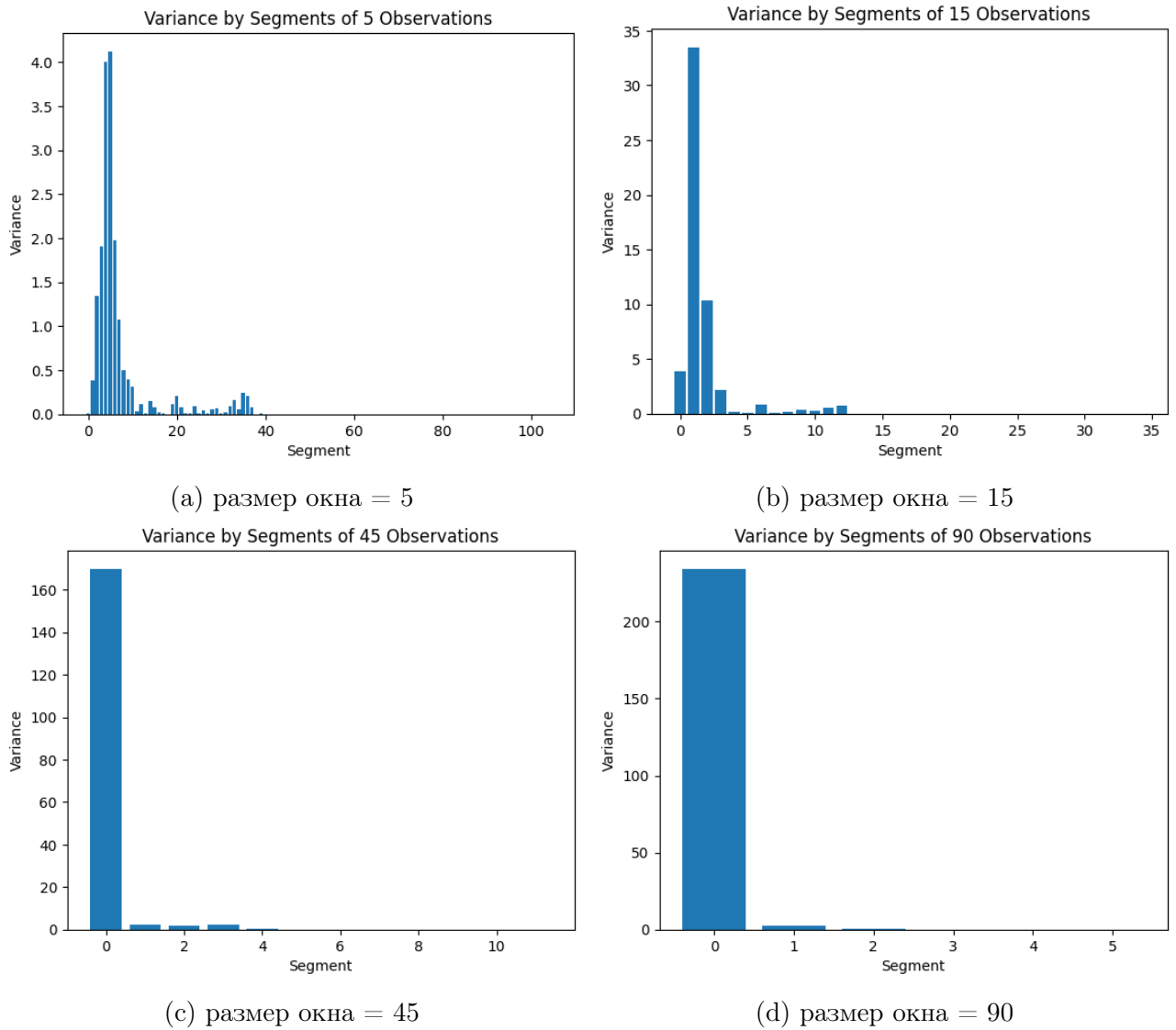


Рис. 3.2: Дисперсия на участках графика, разбитых на окна различных размеров

3.0.2 Подбор порогового значения

Подбор порога же сложнее. Он зависит от типа симуляций, и коллегами, занимающимися одновидовыми симуляциями было получено, что это значение порядка 20.

Нужно рассмотреть первый и второй максимумы дисперсий на окнах и максимизировать вероятность того, что пороговое значение окажется между ними.

В моем случае, дисперсия первого и второго максимумов описывается в [3.1](#):

Таблица 3.1: Описание распределения дисперсии максимумов для окон подобранных выше размеров

	1ый максимум	2ой максимум
count	46	43
mean	2159.453173	1413.528082
std	3020.472232	2411.808612
min	15.842793	7.097566
25%	43.031232	17.108556
50%	115.911026	42.903450
75%	4621.762862	1487.646270
max	8421.269635	7413.687056

Но также заметим медианные значения:

$$median_{max} = 115.911026$$

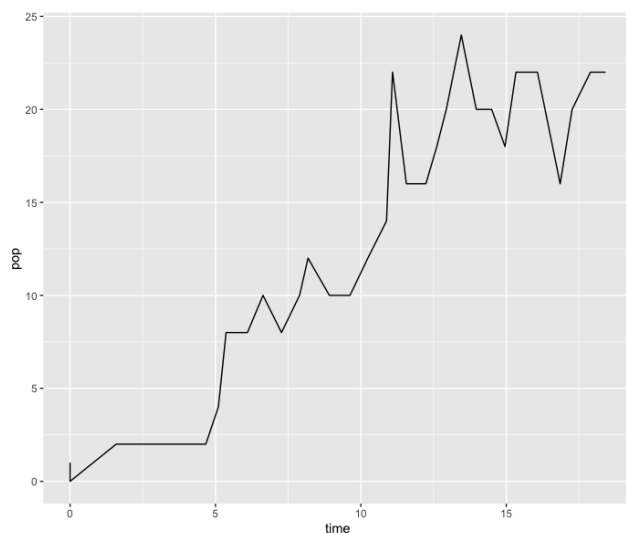
$$median_{max2} = 42.90345$$

То есть для большинства случаев при пороговом значении > 42 первый и второй максимум различаются примерно в 2 раза, то есть они не могут быть перепутаны.

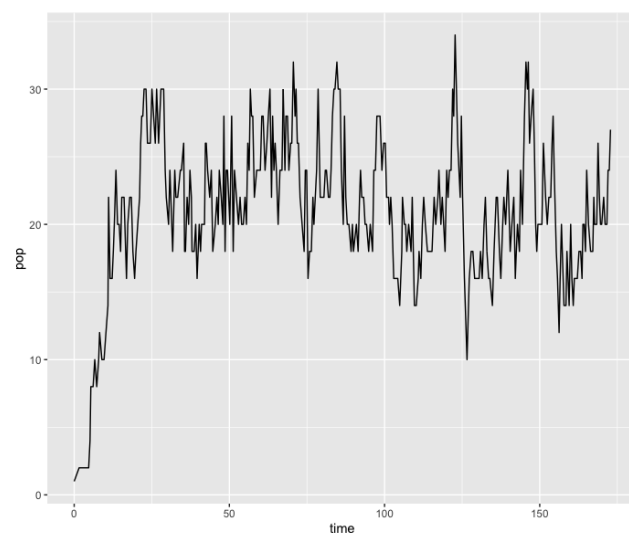
По этим соображениям, как пороговое значение 3d симуляций по умолчанию я выбрала **45**.

3.0.3 Результаты подбора параметров

Для демонстрации эффективности подобранных параметров, я запустила парные симуляции с полным проходом эпох и с остановкой при нахождении плато:



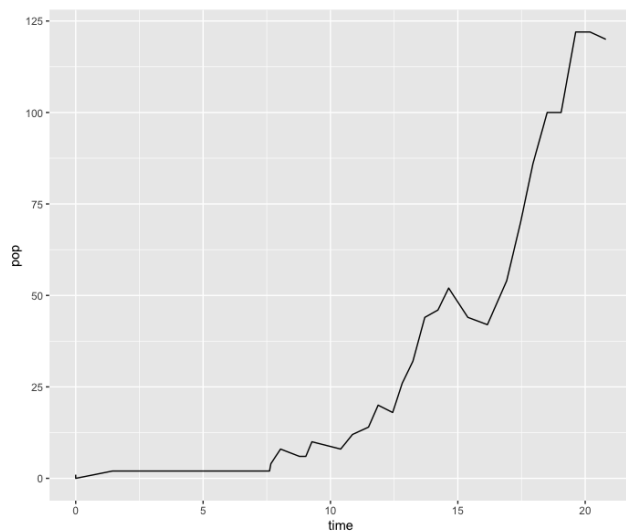
(a) Остановка при выходе на плато



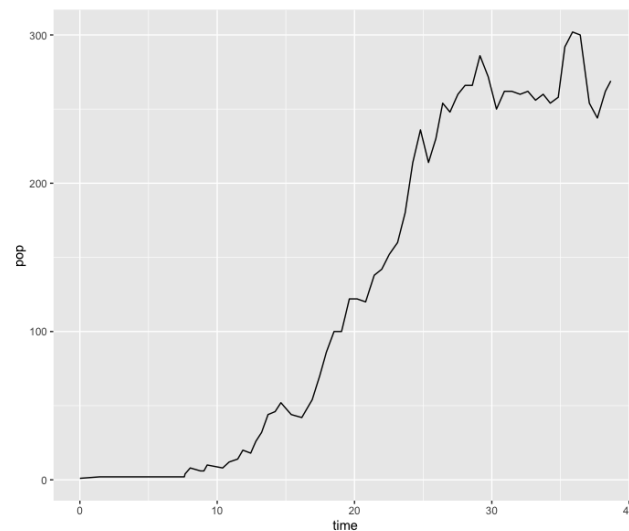
(b) Полный проход

Рис. 3.3: Парные графики, запущенные с параметрами:
 $\text{area_length_x} = \text{area_length_y} = \text{area_length_z} = 10$,
 $b = 1$, $d = 0$, $dd = 50$, $\text{initial_pop} = 1$, $\text{sd_d} = \text{sd_d} = 1$, $\text{death_r} = 3$
 Количество эпох: 1000

Демонстрация минусов сравнительно маленьких размеров окон:



(a) Остановка при выходе на плато



(b) Полный проход

Рис. 3.4: Парные графики, запущенные с параметрами:
 $\text{area_length_x} = \text{area_length_y} = \text{area_length_z} = 15$,
 $b = 0.9$, $d = 0.2$, $dd = 10$, $\text{initial_pop} = 1$, $\text{sd_d} = \text{sd_d} = 0.8$, $\text{death_r} = 2.4$
 Количество эпох: 1000

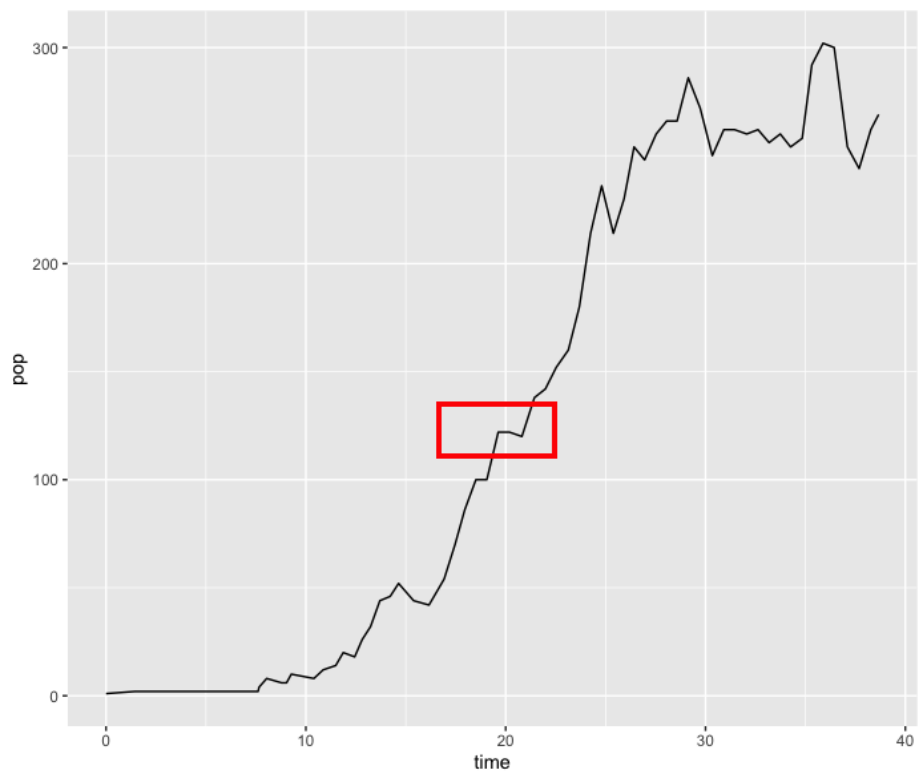


Рис. 3.5: Место, которое алгоритм принял за плато на [3.4](#)

Получается, подобранный порог эффективен, только для гладких на этапе подъема графиках из-за особенностей 3d симуляций.

Выводы

В ходе работы над курсовым проектом мною были изучены различные модели для описания биологических сообществ, в частности Дикмана-Лоу.

Были сделаны доклады на связанные темы, которые можно найти в моем репозитории, а также проведены 1d, 2d и 3d симуляции. Первые для ознакомления, вторые – для коллег занимающихся численным методом, третьи – для нахождения сглаживаний и точки останова.

При выполнении последней задачи главной трудностью, с которой я столкнулась, была моя невозможность наблюдать рост популяции при большом количестве эпох или на большом пространстве, так как симуляции такого вида происходили чрезвычайно медленно и не давали мне выхода на плато.

Поэтому в данной работе рассматриваются относительно легкие 3d симуляции, но в будущем результаты, полученные здесь могут быть адаптированы и для более тяжелых случаев.

Список литературы

- [1] Ulf Dieckmann и Richard Law. “Moment Approximations of Individual-based Models”. В: (2000).
- [2] Serik Sagitov и Altynay Shaimerdenova. “Extinction times for a birth-death process with weak competition”. В: *arXiv:1209.5182v3 [math.PR]* 53 (2015).
- [3] Е.Г. Галкин, В.К. Зеленков и А.А. Никитин. “Компьютерные симуляции и численные методы в двухвидовой модели пространственных сообществ”. В: *International Journal of Open Information Technologies* 7 (2019).
- [4] Соколова Диана. *CourseProject2022BiologicalMath*. URL: <https://github.com/sokanaid/CourseProject2022BiologicalMath> (дата обр. 2023).
- [5] Галкин Егор. *Rcpp interface for c++ based poisson simulator*. URL: <https://github.com/YegorGalkin/RcppSim> (дата обр. 01.06.2021).
- [6] Куркин Максим Леонидович. “Оптимизация параметров численного метода для модели экологических сообществ [Выпускная квалификационная работа]”. В: (2021).
- [7] Ксения Мхайлова. “Исследование биологической модели стационарных сообществ [Курсовая работа]”. В: (2022).

Приложения

- Ссылка на github с материалами проекта:

<https://github.com/le-beda/Coursework2023>

- Ссылка на Google Disk с результатами симуляций:

https://drive.google.com/drive/folders/1EJWM4QVlzch8IK-I6NC5XLUJFA77pxE6?usp=share_link