

Применение больших языковых моделей в имитационном моделировании

А.А. Вакушин, Б.И. Клебанов

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, Екатеринбург

Аннотация: Современный цикл создания имитационных моделей не обходится без аналитиков, моделистов, разработчиков и специалистов из разных областей. Существует множество известных инструментов для упрощения имитационного моделирования, в дополнение к ним предлагается использовать большие языковые модели (БЯМ), состоящие из нейронных сетей. В качестве примера в статье была рассмотрена модель GPT-4. Такие модели могут снизить затраты, будь то финансовые или временные, при создании имитационных моделей. Были приведены примеры использования GPT-4, по результатам которых делается предположение о том, что с помощью БЯМ можно заменить или существенно снизить трудоемкость использования труда большого количества специалистов и даже пропустить этап формализации. Проведена работа по сравнению процессов создания моделей и проведения экспериментов при использовании разных инструментов ИМ и результаты оформлены в сравнительную таблицу. Сравнение сделано по основным критериям ИМ. Эксперименты с GPT-4 успешно показали, что создание имитационных моделей с помощью БЯМ заметно ускоряется и имеет большую перспективу в данной области.

Ключевые слова: имитационное моделирование, большая языковая модель, нейронная сеть, GPT-4, среда моделирования, математическая модель.

Введение

В имитационном моделировании (далее ИМ) при проектировании сложных (особенно междисциплинарных) систем принимают участие множество специалистов разных профилей. Это сотрудники, которые вносят свой вклад в разные аспекты моделирования и обеспечивают полноценное представление о системе [1]. Среди них исследователи (постановщики задач), специалисты в конкретных предметных отраслях, аналитики, математики, моделисты, программисты.

Исследователь ставит задачу, аналитики уточняют требования к модели, взаимодействуя со специалистами в конкретных предметных отраслях. Математики на основе требований определяют математическую модель системы, далее моделисты, используя формальное описание, определяют модель на языке ИМ и проводят необходимые эксперименты.

Результаты экспериментов передаются исследователю, который может уточнить задачу или решить, что полученное решение не соответствует его представлениям и потребовать переделать модель. И в том и другом случае цикл разработки модели повторяется.

Еще более усложняется ситуация, когда для разработки модели не используются специальные средства моделирования и она пишется на языке программирования.

LLM как современный инструмент моделиста

Для упрощения процесса моделирования есть множество инструментов, таких, как системы дискретно-событийного моделирования (например, SimPy или Arena), среды моделирования (например, AnyLogic или Simulink), агентно-ориентированные платформы (например, NetLogo или Repast), системы системной динамики (например, Vensim или Stella), и другие. Каждый из этих инструментов предлагает свои особенности и возможности для формализации и моделирования в различных областях и сценариях.

В связи с взрывным ростом популярности нейросетей и больших языковых моделей (далее БЯМ), в частности, появляется большое количество научных работ с применением данных технологий в разных научных областях: разработка программного обеспечения [2, 3], сельское хозяйство [4], экономика [5], медицина [6], кибербезопасность [7].

В данной статье в качестве дополнительного инструмента предлагается использовать мультимодальную БЯМ GPT-4 компании OpenAI, Inc (USA, Delaware). GPT понимает естественные языки и может генерировать ответ как на естественном, так и формализованном языке. Также GPT содержит информацию из разных дисциплин, например, психологии, физики и экономики, что позволяет решать междисциплинарные задачи. GPT может помогать при анализе данных – обрабатывать, структурировать, создавать

визуализацию. Следует отметить, что GPT является еще и средством интеграции (например, через плагины) знаний в различных предметных отраслях, представленных в специализированных пакетах программ.

Ниже на примерах показано, что GPT может заменить или существенно снизить трудоемкость использования математиков, междисциплинарных специалистов и моделистов, и программистов при разработке ИМ.

Вот простой пример диалога (таблица №1), из которого следует, что GPT на основе простого текстового запроса, касающегося физических законов, минуя этапы формализации и разработки модели, самостоятельно проводит моделирование и визуализирует результат.

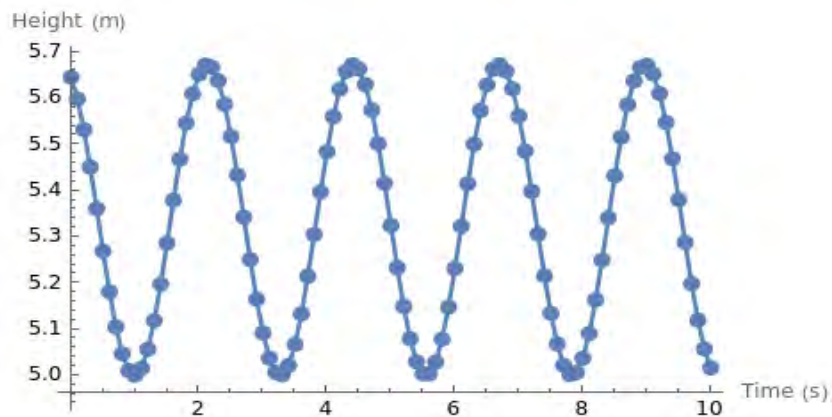


Рис. 1. – График изменения высоты шара (визуализация GPT)

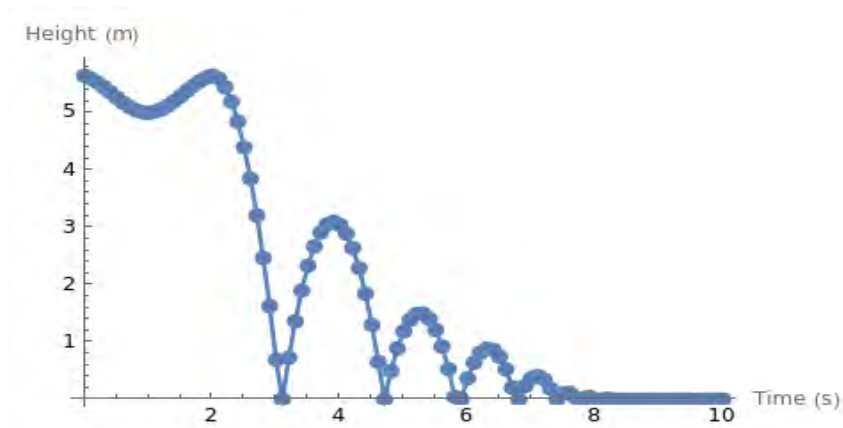


Рис. 2. – График изменения высоты шара после отрыва от стержня (визуализация GPT)

Таблица № 1

Пример разработки модели и получение результата без формализации

Исследователь	GPT-4
Постановка задачи: Шар весом 1 кг висит на высоте 5 метров. Шар прикреплен к стержню, шарнирно закрепленному на высоте 10 метров над землей. Сначала стержень с шаром отведен от вертикального положения на 30 градусов, затем отпущен. Построй график изменения высоты шара. Весом стержня и трениями пренебрегаем	Вот график (рис. 1), показывающий изменение высоты шара весом 1 кг, который висит на высоте 5 метров и прикреплен к стержню, шарнирно закрепленному на высоте 10 метров над землей. Сначала стержень с шаром отведен от вертикального положения на 30 градусов, затем отпущен. Весом стержня и трением пренебрегаем.
Хорошо. Добавим в постановку этой задачи событие: в момент времени 2 сек. шар отрывается от стержня и падает на идеально твердую землю	Вот график (рис. 2), показывающий изменение высоты шара весом 1 кг, который висит на высоте 5 метров и прикреплен к стержню, шарнирно закрепленному на высоте 10 метров над землей. Сначала стержень с шаром отведен от вертикального положения на 30 градусов, затем отпущен. В момент времени 2 секунды шар отрывается от стержня и падает на идеально-твердую землю. Весом стержня и трением пренебрегаем.

На последней диаграмме видно, что шар, как и ожидалось, после отрыва от стержня падает на жесткую поверхность и отскакивает, постепенно уменьшая высоту и период отскоков.

Ниже представлен пример диалога (таблица №2), демонстрирующий результат самостоятельной работы GPT по формализации ранее поставленной задачи.

Пример формализации задачи в диалоге с GPT-4

You (математик)	GPT-4
1	2
<p>Представь математическую модель движения шара в формализме гибридных автоматов.</p>	<p>Модель движения шара в формализме гибридных автоматов включает в себя как непрерывные, так и дискретные компоненты. Гибридные автоматы хорошо подходят для описания систем, в которых происходит взаимодействие между непрерывными процессами (например, движением шара) и дискретными событиями (например, отрывом шара от стержня).</p> <p>Описание модели:</p> <ol style="list-style-type: none"> 1. Непрерывные переменные: <ul style="list-style-type: none"> • h - высота шара относительно земли • v - скорость шара • θ - угол отклонения стержня от вертикали 2. Параметры: <ul style="list-style-type: none"> • l - длина стержня (5 метров) • g - ускорение свободного падения (9.81 м/с^2) • e - коэффициент восстановления скорости при ударе (0.8) 3. Дискретные события: <ul style="list-style-type: none"> • Отрыв шара от стержня (происходит в момент времени $t = 2$ секунды). 4. Динамика системы: <ul style="list-style-type: none"> • Отрыв шара от стержня (происходит в момент • До отрыва (маятниковое движение): <ul style="list-style-type: none"> ◦ $\dot{\theta} = \omega$ ◦ $\dot{\omega} = -lg \sin(\theta)$ ◦ $h = h_0 - l \cos(\theta)$ • После отрыва (свободное падение и отскоки): <ul style="list-style-type: none"> ◦ $\dot{v} = -g$ ◦ $\dot{h} = v$ ◦ При $h \leq 0$: v обновляется как $-e \cdot v$

1	2
	<p>Гибридный автомат:</p> <ol style="list-style-type: none"> 1. Состояния: <ul style="list-style-type: none"> • S_1: Маятниковое движение (до отрыва) • S_2: Свободное падение и отскоки (после отрыва) 2. Переходы <ul style="list-style-type: none"> • Из S_1 в S_2 при $t = 2$ секунды 3. Уравнения в каждом состоянии: <ul style="list-style-type: none"> • В S_1 – используются уравнения маятникового движения. • В S_2 – используются уравнения свободного падения и отскоков. <p>Эта модель представляет собой упрощенное описание реальной системы и может быть дополнена или изменена для учета дополнительных факторов, таких, как сопротивление воздуха или неидеальности в поведении пружины и стержня.</p> <p>Начало формы:</p> <ul style="list-style-type: none"> • $t = \sqrt{\frac{2h}{g}}$ $h_{\text{отскок}} = e^2 \cdot h(e^2)$

Важно, что в зависимости от сложности задачи, GPT либо сам визуализирует результаты моделирования, либо генерирует код, который исследователь должен перенести в среду исполнения и запустить. Следует отметить возможность GPT проверять соответствие полученных результатов постановке задачи. Конечно, все ответы и решения GPT должны контролироваться исследователем.

Был проведён эксперимент создания ИМ мультиагентной системы, ранее представленной формализмом взаимодействующих гибридных автоматов, в ходе которого при взаимодействии с GPT не использовались специфические термины по типу «гибридный автомат», «машина состояний», диффуравнения и т. п. В качестве проверки с помощью GPT

были получены: последовательность действий для известной среды моделирования (далее СМ) непрерывно-дискретных социальных систем – AnyDynamics [8, 9]; исходный код на двух языках программирования (C#, Modelica). Все они были запущены в средах моделирования/разработки. Результат оказался положительным – созданная с помощью GPT модель успешно работала, как было задумано моделистом.

В таблице №3 представлена таблица с нашими оценками сравнения процессов создания имитационных моделей и проведения экспериментов при использовании разных инструментов ИМ, в т. ч. GPT. Оценки представлены в шкале от 1 до 5, где 1 – плохой результат, а 5 – отличный.

Таблица № 3

Сравнительная таблица использования инструментов для ИМ.

	Open Modelica	Open Modelica с GPT	C#	C# с GPT	AnyDynamics	AnyDynamics с GPT	GPT
Скорость создания модели	3	4	1	3	4	5	5
Инструменты для визуального контроля процесса симуляции	4	4	1	3	4	4	4
Функциональность и возможности моделирования	4	5	3	5	4	5	4
Лёгкость использования и разработки с точки зрения моделиста	1	3	1	3	4	4	5
Доступность и стоимость инструмента	5	5	5	5	4	4	5
Наличие готовых компонентов, библиотек, плагинов	4	5	4	4	2	2	3
Требование формализации	да	да	да	да	да	да	нет

Пояснение к критериям сравнения.

Инструменты для визуального контроля процесса симуляции – возможности, которые предоставляет среда моделирования/разработки для отображения данных модели в виде графиков, анимации, диаграмм и т. п.

Лёгкость использования и разработки с точки зрения моделиста – моделисты, как правило, для создания моделей прибегают к помощи программистов и аналитиков данных.

В качестве человека, который создаёт ИМ, возьмём специалиста, у которого нет навыков программирования, знаний прикладной математики, анализа данных, полных знаний предметной области. Для создания модели в Open Modelica требуются технические навыки, из-за этого у моделиста потребуется больше времени — на освоение инструментов и понимание среды моделирования. То же самое можно сказать и про C#, который не предназначен для моделирования, а является инструментом, который позволит сделать всё что угодно, но с требованием к хорошим знаниям этого языка программирования и ценой времени. Однако, порог входа и скорость обучения этим инструментам упрощает GPT ввиду своей простоты в запросах и получения ответов на них. В AnyDynamics порог входа ниже предыдущих инструментов благодаря визуальному конструированию моделей и информативному интерфейсу.

Визуальный контроль присутствует в Open Modelica и включает в себя множество инструментов, однако отсутствуют некоторые возможности (например построение трёхмерных графиков). В AnyDynamics такие инструменты присутствуют, однако при симуляции больших многокомпонентных моделей скорость симуляции падает. А в C# встроенных инструментов нет, однако всегда есть возможность использовать GPT и самостоятельно написать данный функционал.

Если рассматривать ситуацию с точки зрения возможностей и функциональности моделирования, то безусловным лидером является язык программирования C#, т. к. с на нём можно написать инструменты симуляции модели, инструменты визуализации, подсчёта аналитики, графический интерфейс для удобства моделирования и т. д. Единственное, что служит препятствием для реализации – знание языка на недостаточно хорошем уровне и время, потраченное на программирование. AnyDynamics и Open Modelica являются СМ и готовы для работы моделиста из «коробки», а C# является многофункциональным «строительным инструментом». А GPT сглаживает и помогает этот функционал использовать по максимуму.

Open Modelica и C# в большей степени используют запрограммированные команды, а AnyDynamics в большей степени является графической средой. Исходя из того, что мы определили моделиста, как человека без технических знаний, то для него важен низкий порог входа в среду моделирования, однако Open Modelica и C# не имеют такого преимущества. Но снизить этот порог может GPT, моделист в форме запросов на простом языке может получать необходимую информацию.

Немаловажными критериями являются стоимость и доступность. Open Modelica, C# – бесплатны и имеют открытый исходный код, а у AnyDynamics ситуация противоположна – для коммерческой деятельности требуется лицензия и исходный код закрыт. У GPT от OpenAI модель версии ChatGPT-4 по ежемесячной подписке, однако есть бесплатная версия — ChatGPT 3.5-Turbo.

Ещё один важный пункт при моделировании заключается в наличии и разнообразии плагинов, так как зачастую с помощью них можно сделать систему точнее и более функциональной (например, JModelica для Open Modelica служит для оптимизации и анализа динамических систем). Так как Open Modelica очень известный и распространённый инструмент,

пользующийся популярностью у многих моделистов, то для него создано больше количество плагинов и библиотек. Однако про AnyDynamics такого нельзя сказать, так как существуют лишь стандартные библиотеки от авторов данного инструмента, а при поиске в сети Интернет не удаётся что-либо обнаружить. Ситуация с C# аналогичная Open Modelica – известность инструмента и его распространённость, однако недостаток заключается в том, что C# – из «коробки» не язык моделирования, а существующие библиотеки не всегда могут подходить под задачи ИМ.

Заключительный сравнительный критерий – формализация. Для подавляющего количества у всех существующих инструментов ИМ - формализация необходима, так как модель базируется на определённых правилах, ограничениях и взаимодействиях между компонентами (формальных языков, математических уравнений, графов, диаграмм). Но GPT позволяет этот этап обойти. Уже сейчас на естественном языке можно создать простую ИМ.

Теоретически GPT-4 с плагинами может заменять среду ИМ, то есть проводить симуляции математических моделей (например, что будет с агентом через 10 сек.). Так, с плагином «Show Me» диаграммы, процессы модели можно получать прямо из GPT-4, математические расчёты проводить с помощью – Wolfram. На текущий момент компания OpenAI заменила плагины другим похожим функционалом — магазином модифицированных GPT.

Выводы

Исходя из полученных данных, можно сделать вывод о том, что GPT однозначно помогает при создании ИМ. Использование БЯМ на естественном языке сейчас ускоряет создание ИМ, сокращая её технологические этапы — формализацию, программирование и испытание [1], выполняет роль специалистов из сферы разных дисциплин. Есть

предположение, что такие модули могут заменить традиционные СМ, где без использования формализации на естественном языке будут описаны процессы системы.

БЯМ не ограничивается GPT-4, их существует большое множество – как проприетарные, так и свободные, как платные, так и бесплатные. Их количество с каждым месяцем только увеличивается, а их общее качество повышается [10]. И, следовательно, будет расширяться применение БЯМ для создания все более сложных ИМ.

Литература

1. Лычкина Н.Н. Имитационное моделирование экономических процессов: Учеб. пособие. – М.: ИНФРА-М, 2014. – 254 с.
2. Chaozheng Wang, Junhao Hu, Cuiyun Gao, Yu Jin, Tao Xie, Hailiang Huang, Zhenyu Lei, Yuetang Deng Practitioners' Expectations on Code Completion. arXiv e-prints. 2023. arXiv: 2301.03846. 12 p.
3. Jeong C.A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. Advances in Artificial Intelligence and Machine Learning; Research 3 (4). 2023. pp. 1588-1618
4. Jiajun Qing, Xiaoling Deng, Yubin Lan, Zhikai Li GPT-aided diagnosis on agricultural image based on a new light YOLOPC. Computers and Electronics in Agriculture. 2023. №. 213. 108168.
5. Меркулов В.В., Сивко Д.Ю., Дмитриев А.С. Модель прогнозирования фондовых рынков на основе нейронных сетей // Инженерный вестник Дона. 2023. №6. URL: ivdon.ru/ru/magazine/archive/n6y2023/8465.
6. Wei Zhu, Wenfeng Li, Xing Tian, Pengfei Wang, Xiaoling Wang, Jin Chen, Yuanbin Wu, Yuan Ni, Guotong Xie Text2MDT: Extracting Medical Decision Trees from Medical Texts. arXiv e-prints. 2024. arXiv: 2401.02034. 39 p.
7. Семькина Н.А., Садовникова Н.М. Особенности применения методов искусственного интеллекта при решении задачи мониторинга сетевого

трафика с целью обнаружения атак // Инженерный вестник Дона. 2023. №4.
URL: ivdon.ru/ru/magazine/archive/n4y2023/8367.

8. Вакушин А.А., Клебанов Б.И. Сравнительный анализ пакетов MATLAB.Stateflow/Simulink и AnyDynamics для построения имитационных моделей социальных систем // Современные наукоемкие технологии. – 2021. – № 7. – С. 18-23 URL: top-technologies.ru/ru/article/view?id=38748.

9. Клебанов Б.И., Вакушин А.А., Тен Ю.М. Применение пакета AnyDynamics для разработки имитационных моделей социальных систем // Современные наукоемкие технологии. – 2022. – № 4. – С. 46-51; URL: top-technologies.ru/ru/article/view?id=39106.

10. Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, Ji-Rong Wen A survey of large language models. arXiv e-prints. 2023. arXiv: 2303.18223. 124 p.

References

1. Lychkina N.N. Imitatsionnoe modelirovanie ekonomicheskikh protsessov: Ucheb. Posobie [Simulation modeling of economic processes: Textbook.]. M.: INFRA-M. 2014. 254 p.

2. Chaozheng Wang, Junhao Hu, Cuiyun Gao, Yu Jin, Tao Xie, Hailiang Huang, Zhenyu Lei, Yuetang Deng Practitioners' Expectations on Code Completion. arXiv e-prints. 2023. arXiv: 2301.03846. 12 p.

3. Cheonsu Jeong A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. Advances in Artificial Intelligence and Machine Learning; Research 3 (4). 2023. pp. 1588-1618

4. Jiajun Qing, Xiaoling Deng, Yubin Lan, Zhikai Li GPT-aided diagnosis on agricultural image based on a new light YOLOPC. Computers and Electronics in Agriculture. 2023. №. 213. 108168.



5. Merkulov V.V., Sivko D.Yu., Dmitriev A.S. Inzhenernyj vestnik Dona. 2023. №6. URL: ivdon.ru/ru/magazine/archive/n6y2023/8465.
6. Wei Zhu, Wenfeng Li, Xing Tian, Pengfei Wang, Xiaoling Wang, Jin Chen, Yuanbin Wu, Yuan Ni, Guotong Xie Text2MDT: Extracting Medical Decision Trees from Medical Texts. arXiv e-prints. 2024. arXiv: 2401.02034. 39 p.
7. Semykina N.A., Sadovnikova N.M. Inzhenernyj vestnik Dona. 2023. №4. URL: ivdon.ru/ru/magazine/archive/n4y2023/8367.
8. Vakushin A.A., Klebanov B.I. Sovremennye naukoemkie tekhnologii. 2021. № 7. pp. 18-23 URL: top-technologies.ru/ru/article/view?id=38748.
9. Klebanov B.I., Vakushin A.A., Ten Yu.M. Sovremennye naukoemkie tekhnologii [Modern High Technologies]. 2022. № 4. pp. 46-51; URL: top-technologies.ru/ru/article/view?id=39106.
10. Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, Ji-Rong Wen A survey of large language models. arXiv e-prints. 2023. arXiv: 2303.18223. 124 p.

Дата поступления: 20.12.2023

Дата публикации: 29.01.2024