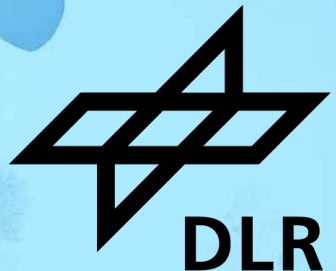


Can ChatGPT, Bard etc. assist with “simple” engineering mathematics?

Some experiments with optimality systems
in PDE-constrained optimization

Fabian Hoppe

German Aerospace Center, Institute for Software Technology, High-Performance Computing, Köln



```
for i in range(1, self.n_clusters):
    distances = ht.spatial.distance.cdist(x, centroids, quadratic_expansion=True)
    D2 = distances.min(axis=1)
    D2.resplit_(axis=None)
    prob = D2 / D2.sum()
    random_position = ht.random.rand().item()
    sample = 0
    sum = 0
    for j in range(len(prob)):
        if sum > random_position:
            break
        sum += prob[j].item()
        sample = j
```

Motivation and Methodology

Motivation



- This talk should be viewed as an invitation to experiment with LLMs: **Choose some interesting task from „your domain“ and try out what happens!**
- LLMs play an important role in software development, e.g., as co-pilots: how good do they „understand“ (better: correctly immitate understanding) some of the underlying objects and structures?
- Curiosity from a trained mathematicians point of view: how good do LLMs perform at a non-trivial task in my (prior) field?
- **Limitations:** small number of experiments for a very specific task, probably outdated as experiments have been done ~July/August 2023

-01.13867v2 [cs.LG] 20 Jul 2023

Mathematical Capabilities of ChatGPT

Simon Frieder^{1,5}, Luca Pinchetti¹, Alexis Chevalier³, Ryan-Rhys Griffiths⁴, Tommaso Salvatori^{2,7}, Thomas Lukasiewicz^{2,1}, Philipp Christian Petersen^{5,6}, and Julius Berner⁵

¹Department of Computer Science, University of Oxford, Oxford, UK
²Institute of Logic and Computation, Vienna University of Technology, Vienna, Austria
³School of Mathematics, Institute for Advanced Study, Princeton, US
⁴Department of Physics, University of Cambridge, Cambridge, UK
⁵Faculty of Mathematics, University of Vienna, Vienna, Austria
⁶Research Network Data Science, University of Vienna, Vienna, Austria
⁷VERSES Research Lab, Los Angeles, CA 90016, USA

Abstract

We investigate the mathematical capabilities of two iterations of ChatGPT (released 9-January-2023 and 30-January-2023) and of GPT-4 by testing them on publicly available datasets, as well as hand-crafted ones, using a novel methodology. In contrast to formal mathematics, where large databases of formal proofs are available (e.g., the Lean Mathematical Library), current datasets of natural-language mathematics, used to benchmark language models, either cover only elementary mathematics or are very small. We address this by publicly releasing two new datasets: GHOSTS and miniGHOSTS. These are the first natural-language datasets curated by working researchers in mathematics that (1) aim to cover graduate-level mathematics, (2) provide a holistic overview of the mathematical capabilities of language models, and (3) distinguish multiple dimensions of mathematical reasoning. These datasets also test whether ChatGPT and GPT-4 can be helpful assistants to professional mathematicians by emulating use cases that arise in the daily professional activities of mathematicians. We benchmark the models on a range of fine-grained performance metrics. For advanced mathematics, this is the most detailed evaluation effort to date. We find that ChatGPT can be used most successfully as a mathematical assistant for querying a mathematical search engine and knowledge base interface. GPT-4 can address undergraduate-level mathematics but fails on graduate-level difficulty. Contrary to reports in the media about GPT-4 and ChatGPT's exam-solving abilities (a potential bias), their overall mathematical performance is well below the level of a graduate student goal is to use ChatGPT to pass a graduate-level math exam, you would be b from your average peer!

3331v3 [cs.CL] 31 Oct 2023

Investigating the Effectiveness of ChatGPT in Mathematical Reasoning and Problem Solving: Evidence from the Vietnamese National High School Graduation Examination

Xuan-Quy Dao
School of Engineering
Eastern International University, VNUHCM City
Binh Duong, Vietnam
quy_dao@eiu.edu.vn

Ngoc-Bich Le
School of Biomedical Engineering
International University, VNUHCM City
Ho Chi Minh City, Vietnam
lbich@biu.edu.vn

ABSTRACT

This study offers a complete analysis of ChatGPT's mathematics abilities in responding to multiple-choice questions for the Vietnamese National High School Graduation Examination (VNSHGE) on a range of subjects and difficulty levels. The dataset included 250 questions divided into four levels: knowledge (K), comprehension (C), application (A), and high application (H), and it included ten themes that covered diverse mathematical concepts. The outcomes demonstrate that ChatGPT's performance varies depending on the difficulty level and subject. It performed best on questions at Level (K), with an accuracy rate of 83%; but, as the difficulty level rose, it scored poorly, with an accuracy rate of 10%. The study has also shown that ChatGPT significantly succeeds in providing responses to questions on subjects including exponential and logarithmic functions, geometric progression, and arithmetic progression. The study found that ChatGPT had difficulty correctly answering questions on topics including derivatives and applications, spatial geometry, and Oxyspatial calculus. Additionally, this study contrasted ChatGPT outcomes with Vietnamese students



Effectiveness of ChatGPT in Coding: A Comparative Analysis of Popular Large Language Models

Carlos Eduardo Andino Coello, Mohammed Nazeem Alimam and Rand Kouatly *

Faculty of Tech and Software Engineering, University of Europe of Applied Sciences, 14469 Potsdam, Germany; carloandinoandino@uea-germany.de (C.E.A.C.); mohammednazeem.alimam@uea-germany.de (M.N.A.)
* Correspondence: rand.kouatly@uea-germany.de

Abstract: This study explores the effectiveness and efficiency of the popular OpenAI model ChatGPT, powered by GPT-3.5 and GPT-4, in programming tasks to understand its impact on programming and potentially software development. To measure the performance of these models, a quantitative approach was employed using the Mostly Basic Python Problems (MBPP) dataset. In addition to the direct assessment of GPT-3.5 and GPT-4, a comparative analysis involving other popular large language models in the AI landscape, notably Google's Bard and Anthropic's Claude, was conducted to measure and compare their proficiency in the same tasks. The results highlight the strengths of ChatGPT models in programming tasks, offering valuable insights for the AI community, specifically for developers and researchers. As the popularity of artificial intelligence increases, this study serves as an early look into the field of AI-assisted programming.

Keywords: artificial intelligence; ChatGPT; GPT-3.5; GPT-4; Python programming; OpenAI; Google's Bard; Anthropic's Claude

ChatGPT is fun, but it is not funny!

Humor is still challenging Large Language Models

Sophie Jentsch¹ and Kristian Kersting^{2,3,4}

¹Institute for Software Technology, German Aerospace Center (DLR), Cologne, Germany

²Computer Science Department, Technical University Darmstadt, Darmstadt, Germany

³Centre for Cognitive Science, TU Darmstadt

⁴Hessian Center for AI (hessian.AI)

sophie.jentsch@dlr.de, kersting@es.tu-darmstadt.de

Abstract

Humor is a central aspect of human communication that has not been solved for artificial agents so far. Large language models (LLMs) are increasingly able to capture implicit and contextual information. Especially, OpenAI's ChatGPT recently gained immense public attention. The GPT3-based model almost seems to communicate on a human level and can even tell jokes. Humor is an essential component of human communication. But is ChatGPT really funny?

We put ChatGPT's sense of humor to the test. In a series of exploratory experiments around jokes, we generate, evaluate, and detect

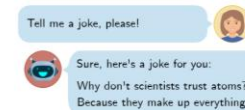


Figure 1: Exemplary illustration of a conversation between a human user and an artificial chatbot. The joke is a true response to the presented prompt by ChatGPT.

The topic of this talk



- **Can LLMs set up the optimality system (KKT system) for optimization problems with PDE-constraints?**
- **Why exactly this task?** (except for the reason that I have good knowledge therein)
 - Non-trivial, quite abstract „symbolic“ task (no calculations!)
 - University level math (usually end of BSc) with applications in Science/Engineering
 - not too much creativity required for the pure result (unlike for reasoning / proofs etc.)

What did I do?

- 9 different optimization problems: 10 times same input prompt (in different „chats“, of course) for each model
- 3 general-purpose LLMs via their chat-interface: Bard, ChatGPT with GPT 3.5, and ChatGPT with GPT 4 (Versions as of July/August 2023)
- Limitation to simple, one-dimensional prototypical model problems:
 - simple (TeX-like) notation
 - reduces recognition of „typical“ problems from papers
 - mathematical reasons (ensure mathematical correctness of “just” doing calculations)
- Pure focus on correct results (no reasoning/proofs etc.)

ChatGPT 4

You
Let a real number $\gamma > 0$ be given. We consider the following optimal control problem:

$$\text{minimize } J(y,u) := \frac{1}{2} (y(1/2)-1)^2 + \frac{\gamma}{2} \int_0^1 |u|^2$$

subject to:

$u \in L^2(0,1)$ and the state equation: $-y'' = u$ on $(0,1)$ with boundary conditions $y(0)=y(1)=0$.

State the optimality system (KKT-system) for this optimal control problem. Only state the optimality system (consisting of the state equation, the adjoint equation (with boundary conditions, of course), and the gradient equation) and do not give any additional explanation.

ChatGPT
To state the optimality system (KKT-system) for the given optimal control problem, we need to include the state equation, the adjoint equation (with boundary conditions), and the gradient equation. Here it is:

- State Equation:**
 $-y'' = u$ on $(0, 1)$
Boundary Conditions:
 $y(0) = y(1) = 0$
- Adjoint Equation:**
Let p be the adjoint state. The adjoint equation is derived by considering the Lagrangian of the problem and differentiating it with respect to y . This yields:
 $-p'' = y(1/2) - 1$ on $(0, 1)$
Boundary Conditions:
 $p(0) = p(1) = 0$
- Gradient Equation:**
The gradient of the Lagrangian with respect to u gives the following condition:



Gemini

Aktuelle Neuigkeiten zum Gemini-Apps Privacy Hub

You
Let a real number $\gamma > 0$ be given. We consider the following optimal control problem: minimize $J(y,u) := \frac{1}{2} (y(1/2)-1)^2 + \frac{\gamma}{2} \int_0^1 |u|^2$ subject to: $u \in L^2(0,1)$ and the state equation: $-y'' = u$ on $(0,1)$ with boundary conditions ...

Vorschläge anzeigen

Absolutely, the optimality system (KKT-system) for the given optimal control problem is:

State equation:
 $-y'' = u$ on $(0,1)$

Boundary conditions:
 $y(0) = 0$
 $y(1) = 0$

Adjoint equation:
 $-p'' = y(1/2)-1$ on $(0,1)$

Boundary conditions:
 $p(0) = \text{free}$
 $p(1) = \text{free}$

Gradient equation:
 $u = -\gamma p'$ a.e. on $(0,1)$

Prompt hier eingeben

Gemini zeigt möglicherweise fehlerhafte Informationen an, z. B. zu Personen. Überprüfe daher die Antworten. [Datenschutz & Gemini-Apps](#)

```
3 # load data set from .h5-file (parallel I/O) on CPU
4 data = ht.load_hdf5('my_data_file.h5', 'my_data_set', split=0, dtype=ht.floa
5
6 # cluster the data set by K-Means
7 clustering = ht.cluster.KMeans(n_clusters=10, init="kmeans++")
8 clustering.fit(data)
9
10 # print the centers found
11 centers = clustering.cluster_centers_
12 print(centers)
```

Some observations...

A very simple model problem



PROMPT:

Let $y_d \in L^2(0,1)$ and a real number $\gamma > 0$ be given. We consider the following optimal control problem:

minimize $J(y,u) := \frac{1}{2} \|y - y_d\|_{L^2(0,1)}^2 + \frac{\gamma}{2} \|u\|_{L^2(0,1)}^2$ subject to:

$u \in L^2(0,1)$ and the state equation: $-y'' = u$ on $(0,1)$ with boundary conditions $y(0)=y(1)=0$.

State the optimality system (KKT-system) for this optimal control problem. Only state the optimality system (consisting of the state equation, the adjoint equation (with boundary conditions, of course), and the gradient equation) and do not give any additional explanation.

Problem:

$$\begin{aligned} \min_{y,u} J(y,u) &:= \frac{1}{2} \|y - y_d\|_{L^2(0,1)}^2 + \frac{\gamma}{2} \|u\|_{L^2(0,1)}^2 \\ \text{s.t.} \quad &u \in L^2(0,1), \\ &-y'' = u \quad \text{on } (0,1), \\ &y(0) = 0, \\ &y(1) = 0. \end{aligned}$$

Results for the very simple model problem



GPT 3.5

The optimality system (KKT-system) for the given optimal control problem is as follows:

1. State Equation:

$$-y''(x) = u(x) \text{ for } x \text{ in } (0,1), \text{ with boundary conditions } y(0) = y(1) = 0.$$

2. Adjoint Equation:

$$-z''(x) = y(x) - y_d(x) \text{ for } x \text{ in } (0,1), \text{ with boundary conditions } z(0) = z(1) = 0.$$

3. Gradient Equation:

$$u(x) = -1/\gamma * z(x) \text{ for } x \text{ in } (0,1).$$

where $y(x)$ is the state variable, $u(x)$ is the control variable, and $z(x)$ is the adjoint variable. The system consists of the state equation, the adjoint equation (with boundary conditions), and the gradient equation for the control variable.

Bard

Sure, the optimality system for the optimal control problem you gave is as follows:

State equation:

$$-y'' = u$$

Adjoint equation:

$$-y'' * \lambda = y - y_d$$

Gradient equation:

$$-2(y - y_d) + 2\gamma u = 0$$

The state equation is the differential equation that the state variable y must satisfy. The adjoint equation is a differential equation that is coupled to the state equation, and it contains the Lagrange multiplier λ . The gradient equation is the equation that must be satisfied by the gradient of the cost functional J .

The boundary conditions for the state equation and the adjoint equation are as follows:

State equation:

$$y(0) = y(1) = 0$$

Adjoint equation:

$$\lambda(0) = \lambda(1) = 0$$

These boundary conditions ensure that the state variable y and the adjoint variable λ are both zero at the endpoints of the interval $(0,1)$.

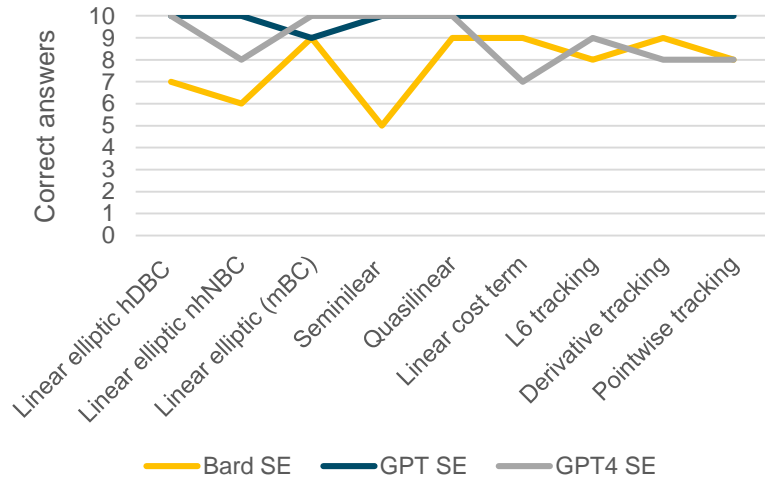
I hope this helps! Let me know if you have any other questions.

| | SE | AE | VI | total |
|---------|----|----|----|-------|
| Bard | 7 | 1 | 0 | 0 |
| GPT 3.5 | 10 | 10 | 7 | 7 |
| GPT 4 | 10 | 10 | 1 | 1 |

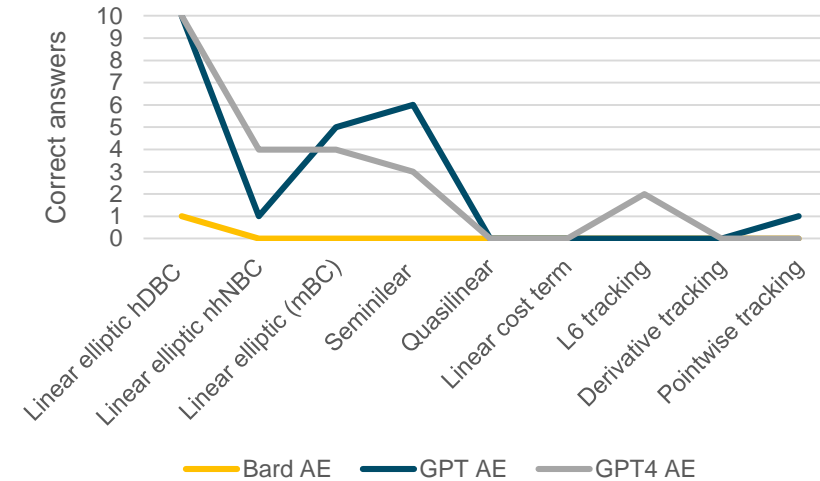
Results for all model problems



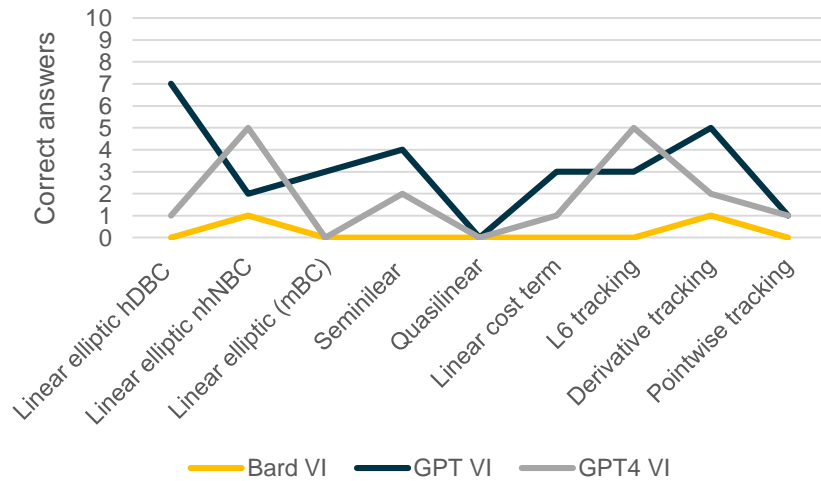
State equation (SE)



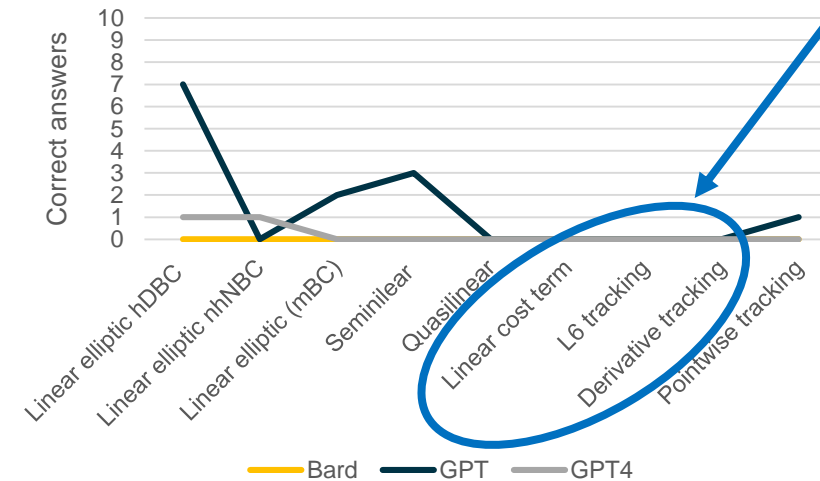
Adjoint equation (AE)



Variational inequality (VI)



Complete System (SE + AE + VI)



for comparison:
certainly **much worse**
than the average of
students in an
introductory lecture on
this topic

A surprising answer...



Problem:

$$\begin{aligned} \min_{y,u} J(y, u) &:= \frac{1}{2} \left(y\left(\frac{1}{2}\right) - 1 \right)^2 + \frac{\gamma}{2} \|u\|_{L^2(0,1)}^2 \\ \text{s.t.} \quad u &\in L^2(0, 1), \\ -y'' &= u \quad \text{on } (0, 1), \\ y(0) &= 0, \\ y(1) &= 0. \end{aligned}$$

GPT
3.5

The optimality system (KKT-system) for the given optimal control problem is as follows:

1. State Equation:

$(-y'' = u)$ on $((0,1))$ with boundary conditions $(y(0)=y(1)=0)$.

2. Adjoint Equation:

$(-z'' = -(y(1/2) - 1)\delta_{1/2})$ on $((0,1))$ with boundary conditions $(z(0)=z(1)=0)$.

3. Gradient Equation:

$(\gamma u = z)$.

Note: In the adjoint equation, $(\delta_{1/2})$ is the Dirac delta function centered at $(1/2)$.

3x by Bard (3x wrong)

2x by GPT3.5 (1x correct)

„Diract Delta Function“ → Bias?



```
for i in range(1, self.n_clusters):  
    distances = ht.spatial.distance.cdist(x, centroids, quadratic_expansion=True)  
    D2 = distances.min(axis=1)  
    D2.resplit_(axis=None)  
    prob = D2 / D2.sum()  
    random_position = ht.random.rand().item()  
    sample = 0  
    sum = 0  
    for j in range(len(prob)):  
        if sum > random_position:  
            break  
        sum += prob[j].item()  
    sample = j
```

Summary / Conclusion

Summary / Conclusion / Additional thoughts



- Usually: the more complicated the problem, the more worse the answers
- All answer „look correct“ for a non-mathematician...
- ...only few errors occur that are straight-forward to recognize for a non-expert mathematician (e.g.: missing conditions, addition of function and real numbers...)
- In many cases „typical“ errors (also across different problems) instead of completely structure-less hallucination
- My impression: GPT 3.5 better than GPT4 much better than Bard
- Often unwanted explanations, mostly for Bard and GPT4 (finetuned for kind conversation!?)
- **Surprising answers:** Dirac measure, y_d is recognized as „desired state“, H^2 -regularity of solutions, correct Lagrange functional is set up and used, typical formulations „*The exact form of these boundary conditions can be quite complex and is not given here.*“ (GPT4)
- **Limitations:** „simple“ stationary problems without additional constraints, small number of samples, limited statistical evaluation, pure math and no coding (in which LLMs could be better), no interaction (one shot Q&A), general-purpose LLMs instead of finetuned models

```
sz = chunk_map[en, axis]
if arr0.comm.rank == en:
    lcl_slice = [slice(None)] * arr0.ndim
    lcl_slice[axis] = slice(ttl, ttl + sz)
    t_arr0 = t_arr0[lcl_slice].clone()
    ttl += sz.item()
```

```
if len(t_arr0.shape) < len(t_arr1.shape):
    t_arr0.unsqueeze_(axis)
```

```
if s1 is None:
```

```
    arb_slice = [None] * len(arr0.shape)
```

```
    arb_slice[axis] = slice(0, arr0.shape[axis])
    chunk_map[arb_slice] -= lshape_map[tuple([0]
```

```
# get the desired data in arr1 on each node with a
```

```
if arr1.comm.rank == arr1.comm.size - 1:
```

Thank you for your attention!