

フュージョンエネルギーにおけるバーチャルラボラトリとAI×データ戦略
～ムーンショット目標10デジタルシステムプロジェクト連続ワークショップ (1)～

生成AIを活用したプログラム開発

2025/3/3 (月) 14:55-15:15

東京大学 物性研究所 吉見 一慶



本日本話しする内容

- **自己紹介と講演の背景**
- 生成AIのソフトウェア開発への応用
- 生成AIのソフトウェア普及活動支援への応用
- まとめ

ソフトウェア開発・高度化プロジェクトの紹介

よりユーザーフレンドリーな
スーパーコンピュータの提供を目指して

毎年12月に公募 (2015年～)

- 1,2件の課題を選定
 - OSSとして年度末に公開
- 年間スケジュールイメージ



PASUMS

Project for advancement of
software usability in materials science

2023年度HPCIソフトウェア賞
普及部門賞の最優秀賞受賞！

主要メンバー (2025/2/27現在)



吉見 一慶
(2015/4～)



本山 裕一
(2018/4～)



青山 龍美
(2022/5～)

April



Kick-off meeting

May ~ Feb.



Software development



GitHub



Regular meeting/month

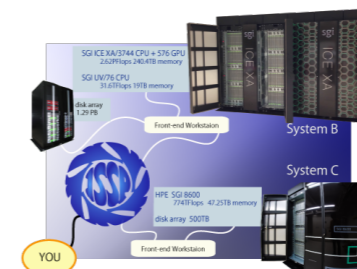
March ~



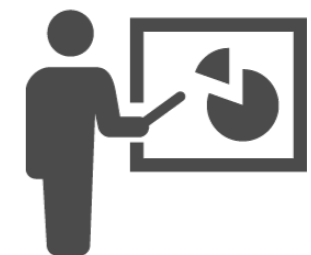
Website

MateriApps

Portal site



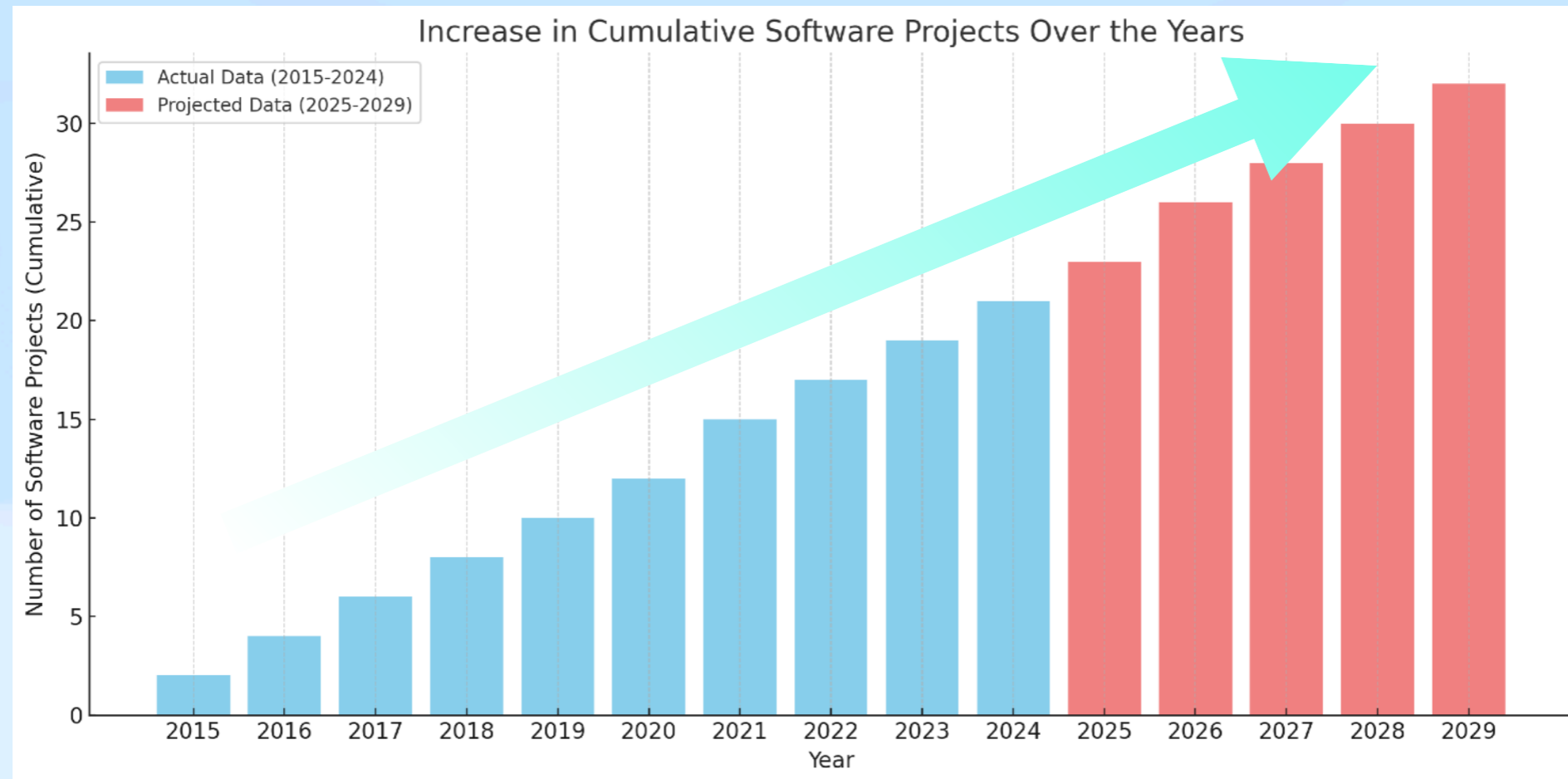
Preinstall



Lecture

生成AIを用いたソフトウェア開発の取り組み背景

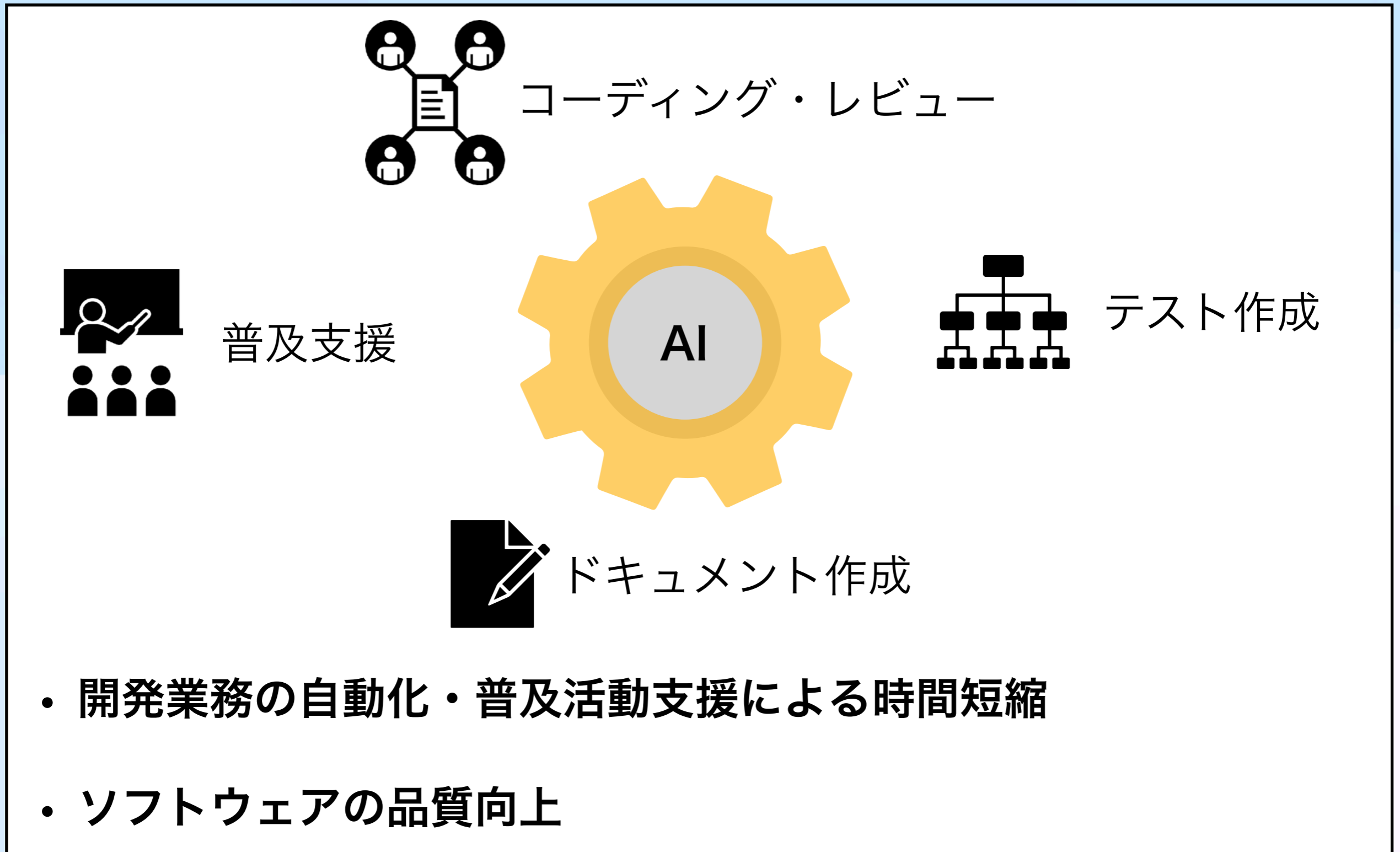
保守するソフトウェア・機能の増加



- コミュニティコードとして管理を目指すが…
 - 個々のソフトウェアのメンテナンスと質の管理が重要
 - 開発プロセスの効率化が求められる

生成AIを用いたソフトウェア開発の取り組み背景

生成AIによる業務改善の可能性



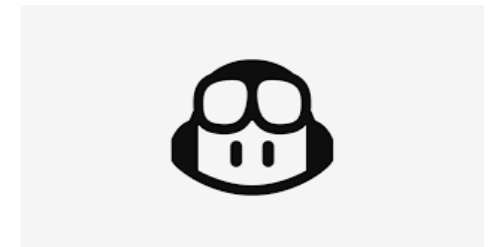
- 開発業務の自動化・普及活動支援による時間短縮
- ソフトウェアの品質向上

本日本話しする内容

- 自己紹介と講演の背景
- 生成AIのソフトウェア開発への応用
- 生成AIのソフトウェア普及活動支援への応用
- まとめ

AIを活用したコーディング支援の利点

- 自動コード補完による開発スピードの向上
(例: GitHub Copilot)
- リアルタイムのエラー検出と修正提案
- ドキュメント自動生成やテストコードの生成
- テスト生成の導入によるバグの早期発見



例) GitHub Copilotを用いた自動コード補完

A screenshot of a code editor window titled "test.py". The editor shows a single line with the number "1". In the top right corner, there is a green checkmark icon, indicating that the code is valid or that an AI suggestion has been accepted.A screenshot of a code editor window titled "test.py" showing a Python function. The code is as follows:

```
1 def is_prime(n):  
2     if n < 2:  
3         return False  
4     for i in range(2, n):  
5         if n % i == 0:  
6             return False  
7     return True
```

A lightbulb icon is positioned to the left of line 3, indicating an AI-generated suggestion or a tip.

GitHub Copilotの提案生成方法

1. オープンソースから学習したパターンを活用

- **学習元:** 大量のオープンソースコードを解析
- **利点:** 一般的なコーディングパターンやベストプラクティスを提案
- **結果:** 効率的で読みやすいコードを自動生成



<https://github.com/features/copilot>

2. 現在編集集中のファイルの内容を解析

- **解析対象:** コード本体とコメントを含むファイルの内容
- **文脈に基づく提案:** 直前のコードやコメントに即したコードを提案



3. 関連ファイルとプロジェクト全体の設定を考慮

- **解析範囲:** 他の開いているタブ、関連ファイル、プロジェクトフォルダの設定ファイル
- **使う情報:** コード内容、設定ファイル（例:settings.jsonなど）
- **結果:** 他ファイルのコード、設定情報、フォルダ構造を基に、プロジェクト全体に即した提案を行う。

ChatGPTとは？

- ChatGPTとは？



<https://chatgpt.com/>

- OpenAIによって開発されたAI言語モデル。
 - 膨大なテキストデータを用いてトレーニングされている。
 - ✓ 学習データ量：GPT-4 10兆トークン、GPT-3.5 約3410億トークン
 - ✓ パラメータ数：GPT-4 約100兆個、GPT-3.5 約1750億個)
- (注) GPT-4は正式な数値は発表していないので巷で予想されている値

- ChatGPTができること

- 人間のようなテキスト会話の実行、質問への回答、説明の提供など
(最近では、画像・ファイル・音声入力などにも対応)。

- 使用例

- 教育、カスタマーサービス、コンテンツ作成など。

GitHub CopilotとChatGPTのコード開発に対する違い



Feature	GitHub Copilot	ChatGPT
Integration	IDEに直接統合	チャットを通じて使用
リアルタイム提案	あり	なし
自然言語理解	コード補完を主眼	高い
コンテキスト認識	高い(現在編集集中のファイル内)	入力内容に応じて変化
使用例	コード補完、リファクタリングなど	コード生成、トラブルシューティングなど

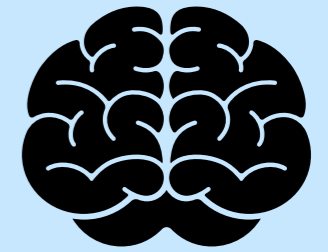
**GitHub Copilotは、IDE内でのコード補完や自動入力をスピーディに行いたい場合に強み。
ChatGPTは、深い理解を要するタスクや、詳細なコード生成、問題解決に強み。**

本日本話しする内容

- 自己紹介と講演の背景
- 生成AIのソフトウェア開発への応用
- **生成AIのソフトウェア普及活動支援への応用**
- まとめ

My GPTsとは？

Plus planで利用可能



- **主な機能**

- プログラミングなしで独自のGPTモデルを作成

- **使用方法**

- **ダッシュボード**：ユーザーフレンドリーなインターフェースで簡単操作
- **API連携**：他のアプリケーションとの統合が可能

- **主な特徴**

- **柔軟性**：さまざまなタスクに適応可能
- **パーソナライズ**：ユーザーのニーズに最適化された設定

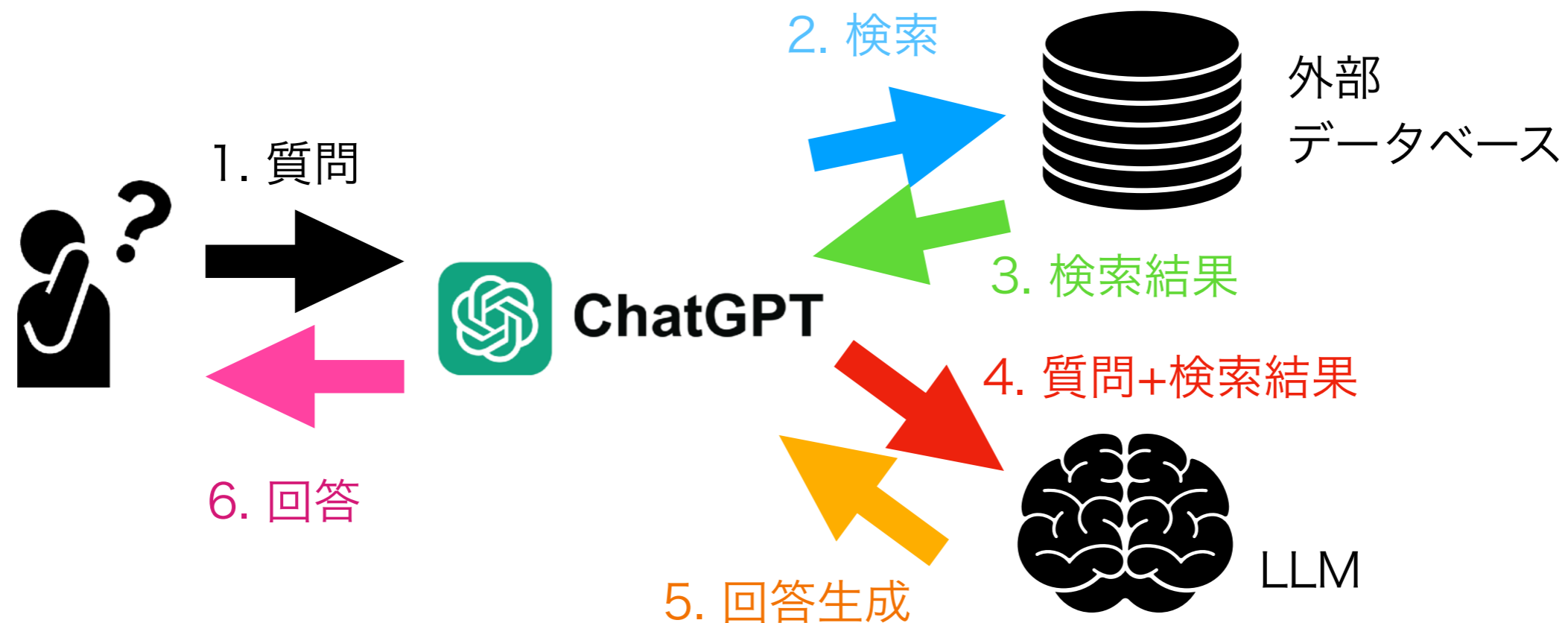


MyGPT作成方法イメージ

RAG (Retrieval-Augmented Generation)

- 情報検索 (Retrieval) + 生成 (Generation) を組み合わせたハイブリッドなAI手法。大規模言語モデル (LLM) に、外部データベースの情報を参照させたいうで回答を生成させるアプローチ。

大まかには、外部データを参照してプロンプトをアップデートするイメージ。



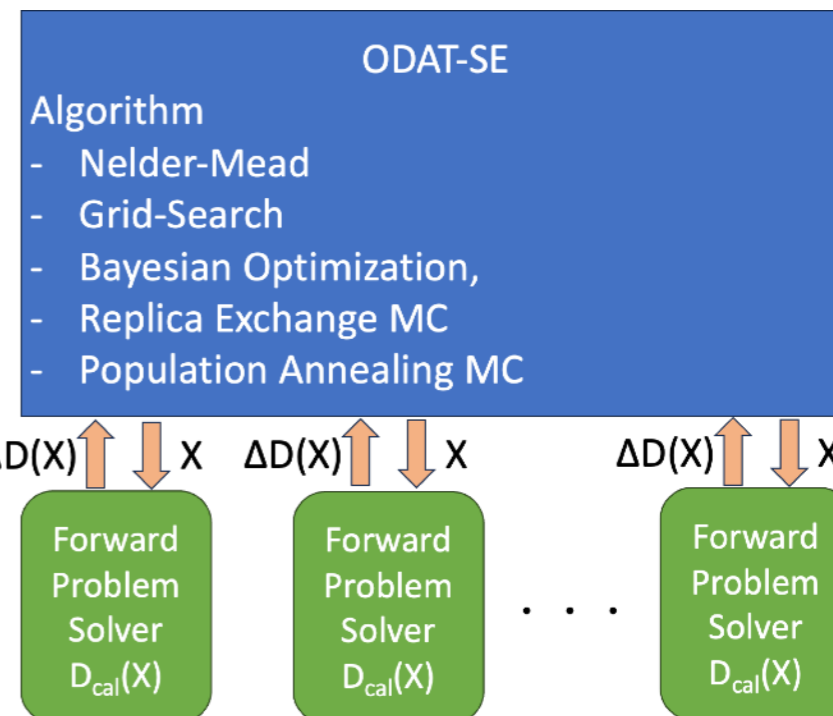
My GPTs 応用例



ODAT-SE-Template Expert (1)

Open Data Analysis Tool for Science and Engineering (ODAT-SE)

順問題ソルバーと逆問題解析アルゴリズムを組み合わせて最適化問題を解く汎用のデータ解析プラットフォーム



URL: <https://github.com/issp-center-dev/ODAT-SE>

主な開発者: Y. Motoyama, K. Yoshimi, T. Aoyama, T. Hoshi

ライセンス: Mozilla Public License v2.0

2024年度ソフトウェア開発・高度化プロジェクトで開発支援中

My GPTs 応用例

ODAT-SE-Template Expert (2)



- odat-se-template マニュアルを事前知識として与えたmyGPT

- <https://chatgpt.com/g/g-6744675c4ee8819185d4eb66782c2016-odat-se-template-tutorial-expert>

- いくつか例題を作って試してみることも可能。
(注) 間違える場合も多いので、あくまでも参考となるベースを作る程度。

ODAT-SE-Template Tutorial Expert



ODAT-SE-Template Tutorial Expert

By KAZUYOSHI YOSHIMI

ODAT-SEへの順問題ソルバー導入に関する日本語チュートリアル専門家。

ODAT-SEのインストール方法を教えてください。

ODAT-SEでモデルを設定する手順を教えてください。

ODAT-SEのデータ解析で使用する主要なコマンドは？

ODAT-SEのエラーを解決する方法を教えてください。

Message ODAT-SE-Template Tutorial Expert

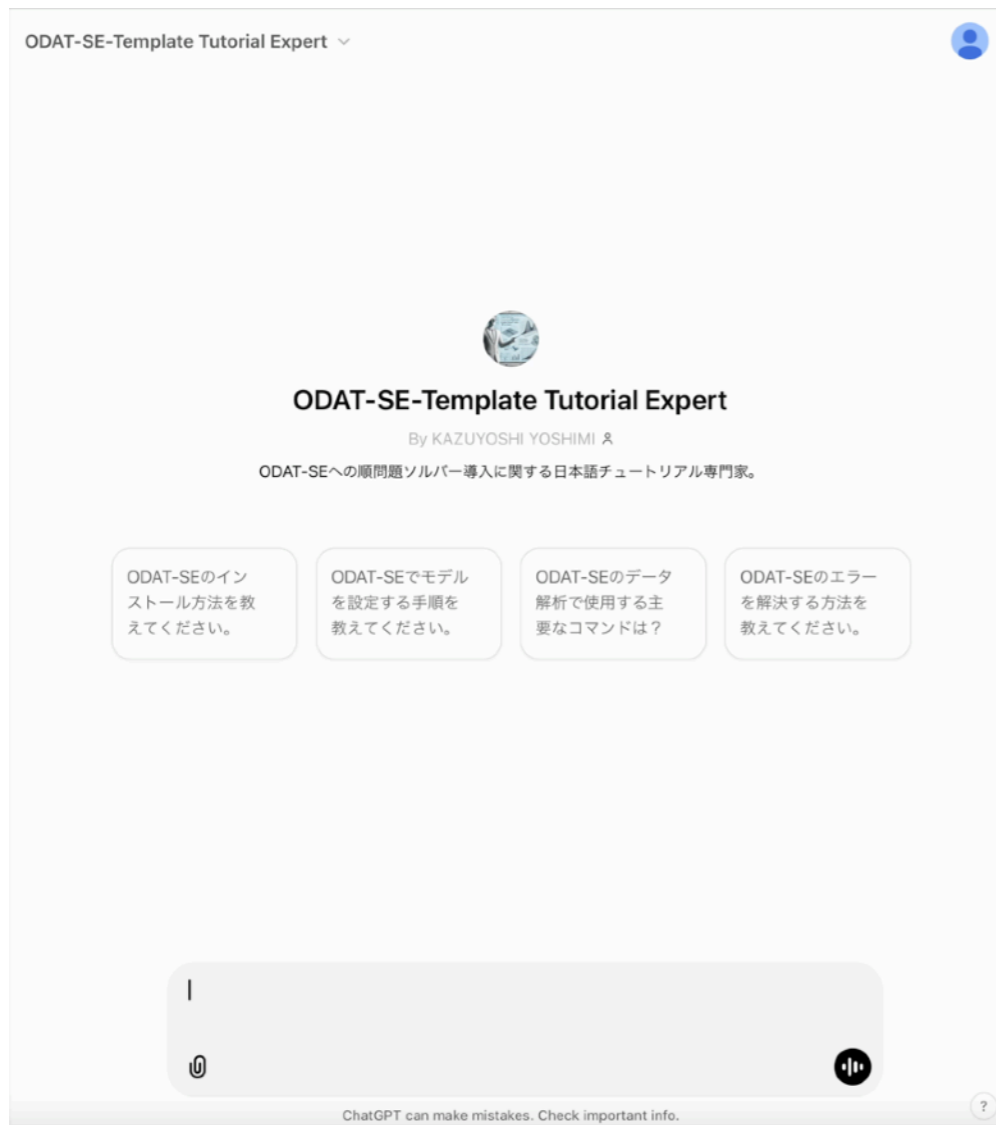


ChatGPT can make mistakes. Check important info.



My GPTs 応用例

ODAT-SE-Template Expert (3)



チュートリアルに従って実行した際に、エラーが出た場合には、そのエラーをメッセージに貼り付けると解決方法を提案してくれる。

実際、今の場合、input.tomlに
[solver]
type = "FourierSolver"
を追加していないため、エラーが出たが、これを追加すると実行できた。

CCMSハンズオン講習会 2024/12/2開催でテスト運用

<https://ccms.issp.u-tokyo.ac.jp/event/7002>

未来予想図(1) AIがコードを読むから教えるへ

AIチュータの進化と自動コード解説・最適な呼び出し方の提案

📖 AIチュータの役割の変化

- AIがコードの動作を解析し、自動で解説。
- 適切なAPIや関数の利用方法を提示。
- 例: 「このコードは何をしているのか?」 → AIが解説。

⚠️ 取り組むべき課題

- AIによるコード解析を活かすために、開発者が**どのようにコードを書けばAIが理解しやすいかを**考慮する必要がある。
- AIチュータを活用するために、開発者が**適切な質問をし、適切な出力を得るスキル**を身につける必要がある。
- **AIによる解説が正しいかどうかを判断し、適切に修正・補完できる知識**が求められる。

📖 ユーザーがコードを入力

🤖 AIチュータが解析

📝 コードの動作解説
& 最適なAPIの提示

🚀 コーディングを加速

未来予想図 (2) AIによるコード組み合わせ解析の進化

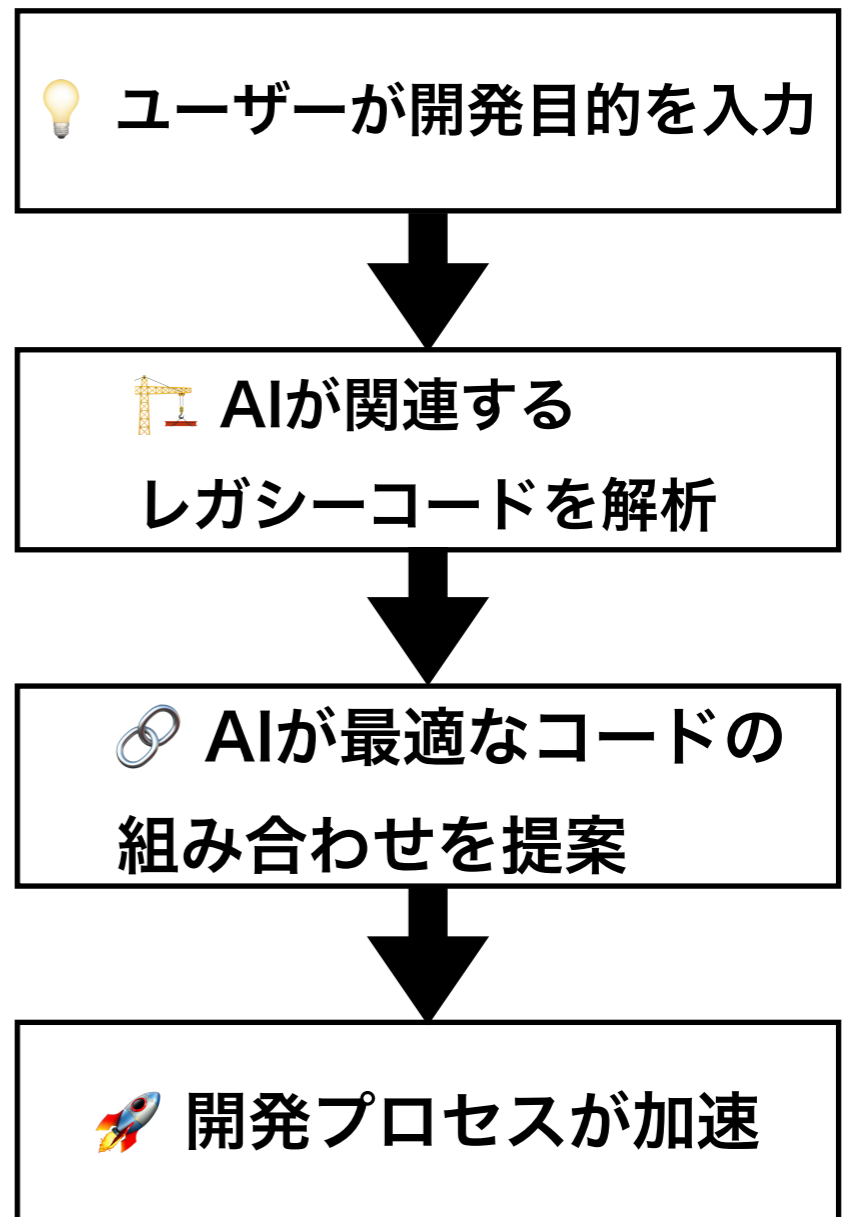
AIが複数のレガシーコードを統合し、最適なシステムを提案

🔗 モジュールの関連性をAIが解析

- レガシーコードのモジュール間の関係を理解し、再構成。
- 自動で「推奨モジュール」を選択し、組み合わせを提案。
- 例: 「〇〇の機能を実装したい」 → AIが関連コードを検索し、最適なモジュールを提示。

⚠️ 取り組むべき課題

- AIが推奨するコードの組み合わせが適切かどうかを評価できる能力を身につける必要がある。
- 開発者がAIに対してどのようなコードを提供するかによって、解析結果が変わるため、入力コードの整理が重要になる。
- レガシーコードのモジュール化を進め、AIが解析しやすい環境を整備する必要がある。

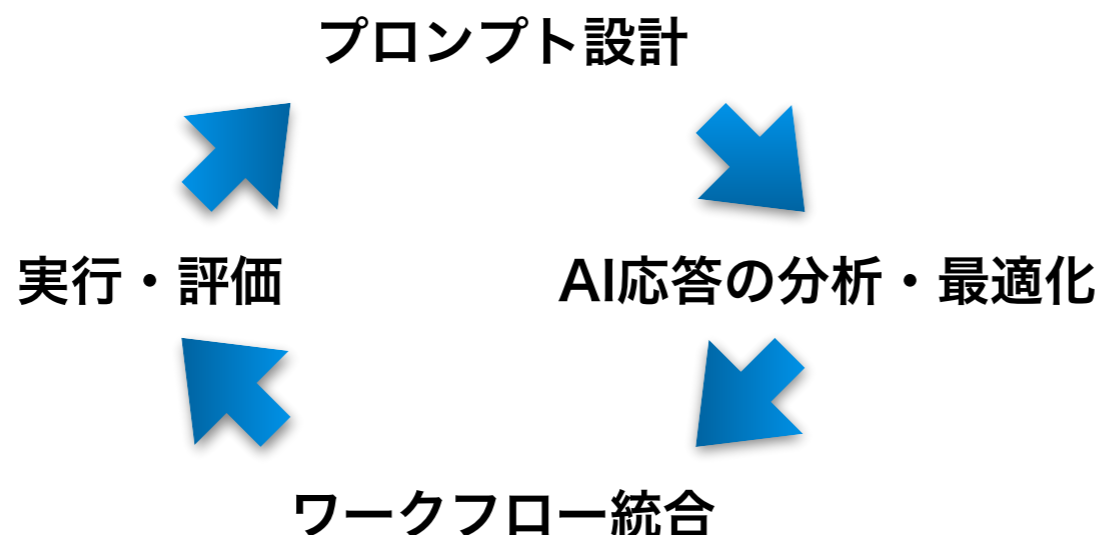


生成AIを使う vs. 使いこなす

単なる情報取得ツールとしての利用から、研究・開発フローに統合する高度活用へ

項目	熟練者（高度活用者）	利用者（基本的な活用）
プロンプト設計	目的に応じて最適化・調整	簡単な質問を入力するのみ
AIの限界理解	誤情報を識別・補正	出力をそのまま信用
ワークフロー統合	業務・研究に組み込む	スポット利用に留まる
カスタマイズ能力	API・スクリプト活用	既存UIのみ利用
継続的学習	最新技術を学習・適用	現在の機能に依存

熟練者のワークフロー (サイクル型)



利用者のワークフロー



AIMHack2024



再委託機関として企画・運営実施

ハッカソンの概要

- 日時：2024年7月24日～26日
- 場所：御殿場高原 時之栖
- 参加者：26名
- 目的：ChatGPTなどの生成AIを利用することで、材料科学のソフトウェア・データベースの効率的な活用方法を学ぶことを主な目的とする
 - 参加者がソフトウェアを選んだ上で、ChatGPTを活用しながら、基礎操作、インストール、実行までを実践する。また、ChatGPTの回答を参考に内容の真偽を確認しながら、使用方法を簡潔にまとめたレビューを作成する。
 - ハッカソンで得られたノウハウを共有し、生成AIの活用方法を向上させる



パンフレット表紙
(ChatGPTにより作成)

MateriAI2024

社会連携研究部門
の活動として企画・運営実施

計算物質科学分野におけるAI技術の活用

Event

2025 - MON 2025 - WED
03/03 ~ 03/05 13:00 ~ 15:00 受付中

MateriAI 2024 ~ 計算物質科学分野におけるAI技術の活用 [Utilization of AI technology in the field of computational materials science]

御殿場高原 時之栖/Gotemba Kogen Tokinosumika
静岡県御殿場市神山719/719 Kamiyama, Gotemba, Shizuoka

参加申し込み >



- 近年、人工知能（AI）は多分野で急速な進歩を遂げており、計算物質科学の領域でも、新材料設計や物性予測などでその有用性が確認されています。今回のハッカソンでは、こうしたトレンドに着目し、既存の物質科学関連データから新たな法則やパターンを抽出する「発見型」アプローチを取り入れます。
- 各チームでのプロジェクト作業を通じて、実践的なスキルとチームワークの経験を積むこともできます。

<https://ccms.issp.u-tokyo.ac.jp/event/7033>

次年度以降も開催予定！

まとめ

生成AIを活用した計算物質科学ソフトウェア開発の概要と今後の展望

- **ソフトウェア開発の背景と課題**

- PASUMSによるソフトウェア開発は進む一方、普及や保守管理に時間と労力がかかることが懸念される。

- **生成AIの応用:**

- 自動化ツール (例: GitHub Copilot) を活用し、コーディング・レビュー・ドキュメント作成の効率化を図る。チュートリアル記事やレビュー記事作成の支援 (例: MyGPTの活用) → 開発者以外の研究者や学生にもツールを使いやすくするための支援活動に利用できる可能性。
- AIを使う際の注意点としては、最終確認の重要性及び意図の明確化が挙げられる。

- **生成AIを活用したさらなる効率化:**

- ソフトウェアの質を向上させると同時に、普及とメンテナンスの両立を図る。
- 教育や研究者のコミュニティとの連携を強化し、効果的なツールを開発・展開する。

Appendix

生成AIを熟練活用するための5つのスキルセット

基礎知識から応用技術まで、熟練者に求められるスキル例

AIの基礎知識

- LLMの仕組みと学習データの影響を理解
- AIの強みと限界、誤りを見抜く力を養う

高度なプロンプト設計

- 質の高い出力を得るためのプロンプトエンジニアリング
- 「役割指定」「分割質問」「出力フォーマット指定」などのテクニック活用

ワークフロー統合

- 生成AIを日常業務や研究プロセスに組み込み、反復可能なフローを構築
- 自動化・バッチ処理の導入

カスタマイズ・拡張

- API・スクリプトを活用し、AIの出力を高度に制御
- 独自のデータを活用したAIの最適化（ファインチューニング）

継続的な学習と実践

- 最新のAI技術をキャッチアップし、実用レベルで応用
- 研究会・ハッカソンなどでスキルを磨く