

LLMを用いた鋼プレートガーダー橋 BIMモデルの自動生成手法の考察

阿部 大樹¹

¹東京大学工学部社会基盤学科（〒113-8656 東京都文京区本郷七丁目 3-1）

E-mail: hiroki-abe510@g.ecc.u-tokyo.ac.jp

インフラ老朽化対策と BIM/CIM 推進が進む中、橋梁 3D モデルの作成工数が普及の障壁となっている。本研究では、大規模言語モデル（LLM）を用いて鋼プレートガーダー橋（RC 床版）の BIM モデルを自動生成するシステムを提案した。本システムの特徴は、(1)RAG（検索拡張生成）により道路橋示方書等から設計ルールを検索し、DesignRule として構造化抽出することで、部材寸法等の設計パラメータ初期値を自律的に決定し、従来の経験則のハードコードに依存せず設計根拠を明示する点、(2) 曲げ・せん断・たわみ等の照査計算を決定論的処理で実行し LLM の計算誤りを排除する点、(3) 照査不合格時には LLM が修正案を生成し合格まで繰り返す自動修正ループを備える点、(4) 橋長・幅員の 2 パラメータのみから IFC 形式の 3D モデルまで一貫出力する点である。橋長 20～70m、幅員 8～24m の 32 条件に対して RAG 有無の 2 水準で計 192 件の設計生成実験を行った。RAG ありでは初回合格率 37.5%、修正ループ適用後の収束率 85.4% を達成し、RAG なしの初回合格率 0%、収束率 75.0% と比較して、本実験条件の全体傾向として RAG の有効性を確認した。RAG により桁高/支間比のルールが参照され、ウェブ高が平均 13.6% 増加することで、特にたわみ照査の初回合格率が 44.8 ポイント向上した。一方、広幅員条件（24m）では桁本数選択が修正不可のため収束率が低下する課題が残った。本システムは概略設計レベルの簡易照査（曲げ/せん断/たわみ/床版厚/幅厚比/横桁配置）を満たす断面案及び BIM モデルを自動生成できることを実証した。今後は対傾構・横構の追加、座屈・疲労照査の厳密化、桁本数を含む設計変数の修正機構導入が課題である。

Key Words: BIM, LLM, RAG, Steel Plate Girder Bridge, IFC, Structural Design Automation

1. はじめに

我が国では、高度経済成長期に建設された橋梁・トンネル等の社会インフラが老朽化を迎えつつある。建設後 50 年を超える道路橋の割合は年々増加しており¹⁾、インフラの維持管理および更新の重要性がますます高まっている。しかし、土木建設業界は他産業と比較して生産性向上が遅れており、増大する維持管理需要に対して慢性的な人手不足が深刻な課題となっている。こうした状況を受け、国土交通省は i-Construction²⁾ をはじめとする施策を通じて ICT 活用による生産性向上を推進しており、設計・施工・維持管理の各段階においてデジタル技術を活用した業務効率化が強く求められている。

このような生産性向上の取り組みの中核として、BIM/CIM (Building/Construction Information Modeling) の導入が推進されている³⁾。BIM/CIM は 3 次元モデルに属性情報を付与し、設計から維持管理までの一貫したデータ活用を可能とする技術であり、国土交

通省は 2023 年度より小規模工事を除く直轄工事において原則適用を開始した。しかし、3D モデルの作成には依然として多大な労力と時間を要し、特に橋梁のような複雑な構造物では手作業によるモデリング工数が大きな負担となっている。この工数負担が BIM/CIM の普及を妨げる一因となっており、モデリングの自動化・省力化が喫緊の課題である。

こうした課題に対し、近年急速に発展している大規模言語モデル（LLM）の活用が有望視されている。LLM は自然言語による指示から構造化されたデータを生成する能力を持ち、設計業務の自動化への適用可能性が期待されている。そこで本研究では、LLM を用いて自然言語ベースで BIM モデルを構築できるエージェント型システムを提案する。なお、本研究は LLM による BIM モデル生成の基礎的な技術開発を範囲とするため、パラメトリックに作成のしやすい鋼プレートガーダー橋（RC 床版）を対象を限定する。本システムの特徴は、(1) 最小限の入力（橋長・幅員の 2 パラメータ）から断面設計を自動生成する

点、(2)RAG (Retrieval-Augmented Generation: 検索拡張生成) により道路橋示方書等の設計知識を参照し設計根拠を明示する点、(3) 決定論的な照査計算と LLM による自動修正ループにより設計品質を担保する点、(4)IFC 形式での出力により BIM 環境との連携を実現する点である。

本論文の構成は以下の通りである。2 章では背景および関連研究を述べ、3 章で提案システムの詳細を説明する。4 章では評価実験の設定と結果を示し、5 章で考察を行う。6 章で本研究のまとめと今後の課題を述べる。

2. 背景および関連研究

(1) BIM/CIM と橋梁モデリングの現状

国土交通省は 2023 年度より小規模工事を除く直轄工事において BIM/CIM (Building/Construction Information Modeling) の原則適用を開始した³⁾。BIM/CIM は設計・施工・維持管理の各段階において 3 次元モデルと属性情報を一貫して活用する技術であり、建設業の生産性向上に大きく寄与することが期待されている。

BIM/CIM モデルのデータ交換標準としては IFC (Industry Foundation Classes) が広く採用されている。IFC は buildingSMART International が策定するオープン標準であり、ベンダー非依存の相互運用性を提供する。特に IFC4x3 ではインフラ (橋梁・道路・鉄道等) への対応が強化され、橋梁 BIM モデルの標準フォーマットとしての利用が進んでいる⁴⁾。

一方で、BIM/CIM の普及には課題も残されている。国土交通省が実施したアンケート調査⁵⁾ によれば、BIM/CIM の導入により「非効率になった」と回答された項目の最多が「モデル作成に手間 (時間、費用、人) がかかる」であり、回答者の 33%がこの課題を指摘している。特に橋梁のような複雑な構造物では、3 次元モデルの作成に多大な工数を要することが BIM/CIM 普及の障壁となっている。本研究はこのモデリング工数の課題に対し、LLM による自動化アプローチを提案するものである。

(2) LLM の建設・土木分野への応用

近年、大規模言語モデル (LLM) の建設・土木分野への応用に関する研究が活発化している。ここでは、LLM と BIM モデルの連携に関連する先行研究を整理する。Nithyanantham ら⁶⁾ は、LLM で IFC データを MCP (Model Context Protocol) 経由で直接操作するフレームワーク「MCP4IFC」を提案した。このフ

レームワークにより、自然言語による指示から BIM モデルの作成・編集が可能となるが、対象は壁・部屋など建築向けの汎用的な幾何形状操作が中心である。そのため、橋梁設計において不可欠な構造計算や工学的制約に基づいたモデル生成については考慮されていない。Yang ら⁷⁾ は、LLM を用いて設計仕様書から構造化ルールを抽出し、IFC モデルの適合性をチェックする手法を提案した。仕様書からのルール抽出精度は 79.5%、IFC モデルへの適合性チェック精度は 84.4%を達成している。ただし、この研究は既存モデルに対する仕様書テキストとの形式的な適合性チェックを主眼としており、応力照査などの数値計算を含む設計プロセスの自動化や、モデルそのものをゼロから生成するアプローチとは異なる。山本ら⁸⁾ は、GPT-4o を用いて直方体や円柱などの単純形状の IFC ファイルを生成する手法を試みている。しかし、IFC ファイルは例えば JSON、csv のような世に広く流通している形式ではなく、LLM の学習データに占める割合が小さいことは容易に推察される。実際、この研究ではスキーマ違反などの構文エラーが多発し、単純な形状でさえ生成成功率が 30~60%程度に留まることが課題として報告されている。これに対し本研究では、LLM の出力を厳密に型定義された構造化データ (JSON/Pydantic) として受け取り、それを python プログラムで IFC へ変換する手法をとることで、BIM データとしての整合性を保証している。

LLM による自動生成というと、IFC 等の形状記述を直接生成する「形状生成」に注目が集まりやすい。しかし、橋梁の概略設計では、示方書や既往文献に基づく経験則・設計ルールを参照し、桁高や板厚などの設計パラメータの初期値を設定した上で、照査により断面を収束させる作業が大きな比重を占める。従来の設計自動化では、例えば桁高を $h \approx L/20$ とするようなルールを設計者がハードコードする必要があった。これに対し本研究では、RAG により文献から設計ルールを動的に取得し、LLM がそれらを解釈して設計パラメータの初期値を自律的に決定する点 (DesignRule 抽出) を重視する。

(3) RAG (検索拡張生成) による専門知識活用

箱石ら⁹⁾ は、汎用的な事前学習済みモデルである BERT が土木分野の専門文書では精度が低下することを示している。GPT や Gemini などの基盤モデルが大幅に進化した現在でも、引き続き LLM の性質としての専門分野での精度低下には対応が必要であるが、LLM の専門知識対応には大きく分けて fine-tuning と RAG の 2 つのアプローチが存在する。

fine-tuning は、事前学習済みモデルを特定ドメイン

表-1: 先行研究と本研究の機能的・技術的比較

先行研究	アプローチと課題	本研究の独自性・解決策
MCP4IFC ⁶⁾	幾何形状の直接操作（建築汎用）	橋梁に特化し、RAGと推論に基づく 自律的な設計パラメータ決定
Yang ら ⁷⁾	道路橋の BIM モデルに対して、テキストルールとの照合による形式的な適合性チェック	力学計算による 工学的照査 に加え、不適合を解消する 自動修正ループ
山本ら ⁸⁾	LLM による IFC テキスト直接生成（単純形状のみ）	構文エラーを防ぐため JSON でパラメータセットを生成し、python で IFC へ変換 することで実構造物の BIM を堅牢に生成

のデータで追加学習させ、専門知識を内部パラメータに埋め込む手法である。しかし、基盤モデルの性能が急速に進化していく 2026 年 1 月現在において、最新モデルが必ずしも fine-tuning に対応しているとは限らない。例えば、OpenAI の gpt-5.1 は fine-tuning に対応していない¹⁴⁾。さらに、fine-tuning には大量の学習データと計算リソースを必要とし、知識の更新時には再学習が必要となることも課題として大きい。

一方、RAG (Retrieval-Augmented Generation: 検索拡張生成) は、外部知識ベースから関連情報を検索し、その情報を基にテキストを生成する手法である¹⁰⁾。RAG は以下の手順で動作する：(1) 文書を適当な単位（チャンク）に分割し、各チャンクの埋め込みベクトルを事前に計算してインデックスを構築する、(2) 質問が与えられたとき、質問の埋め込みベクトルと類似度の高いチャンクを検索する、(3) 検索されたチャンクをコンテキストとして LLM に入力し、回答を生成する。あくまで既存モデルに外部のコンテキストを付与して生成させる手法であるため、API 経由で最先端のモデルを使用できることに加え、インデックス構築のみで対応可能であり、知識の更新が文書の差し替えで容易に行えることも利点として大きい。

さらに、Soudani ら¹¹⁾によると、低頻度かつ専門的な知識に対しては RAG が fine-tuning を大幅に上回る性能を示す。道路橋示方書のような専門的な設計規準は学習データ中の出現頻度が低いため、RAG によるアプローチが適していると考えられる。

以上を踏まえ、本研究では RAG を採用することで、(1) LLM のハルシネーション（事実と異なる出力）を設計規準の参照により抑制する、(2) どの文献のどのページを根拠としたかを記録し設計根拠のトレーサビリティを確保する、という 2 つの効果を期待している。

(4) 本研究の位置付け

a) 先行研究との差分

先行研究と本研究の関係を表 1 に整理する。

b) 本研究の新規性

本研究の新規性は以下の 3 点である：

- DesignRule 抽出と自律的な初期値決定**：道路橋示方書等から桁高/支間比（例： $h \approx L/20$ ）のような設計ルールを検索し、LLM がそれらを解釈して設計パラメータ（部材寸法等）の初期値を自律的に決定する。ルールと参考文献（ページ）を併せて記録することで、従来のハードコードに依存しない設計根拠の明示を実現する。
- 決定論照査と LLM 修正のハイブリッド**：近年の研究では、LLM は数学問題で高い正答率を示す一方、計算ミス（calculation error）が主要かつ難しい誤りとして残存することが報告されている¹²⁾。本システムでは、照査計算（曲げ・せん断・たわみ・床版厚・腹板幅厚比・横桁配置）は数式ベースの決定論的处理で確実に実行し、修正提案のみ LLM が生成する。これにより、工学的妥当性を担保しつつ LLM の柔軟な修正提案能力を活用する。
- IFC 出力までの一貫パイプライン**：橋長・幅員の 2 パラメータ入力から、設計生成 → 照査・修正ループ → IFC 出力までを自動化する。BIM/CIM 環境への直接連携を実現し、モデリング工数を大幅に削減する。

3. 提案システム

(1) システム概要

本研究で提案するシステムは、橋長 L [m] と幅員 B [m] の 2 パラメータのみを入力として受け取り、鋼プレートガーダー橋（RC 床版）の断面設計を自動生成し、IFC 形式の 3D モデルとして出力するシステムである。本システムの処理は、RAG（検索拡張生成）、Designer（設計生成）、Judge（照査・修正提案）、IFC 変換の 4 つのサブシステムで構成される。

システムの全体的な処理フローを図 1 に示す。ま

ず Designer が RAG を用いて道路橋示方書等から関連条文を検索し、それらを参照しながら主桁本数、桁高、板厚、床版厚などの断面寸法を決定する。次に Judge が生成された設計に対して曲げ・せん断・たわみなどの照査計算を実行し、不合格の場合は LLM が修正提案 (PatchPlan) を生成する。この照査と修正のループを合格するまで繰り返し、最終的な設計を IFC 形式で出力する。

本システムの主な技術的特徴は以下の 5 点である。第一に、OpenAI API の Structured Output 機能¹³⁾ を活用することで、LLM の出力を Pydantic¹⁵⁾ スキーマで厳密に定義された構造化データとして受け取り、型安全性を確保している。これにより、LLM からの応答を常に IFC 変換しやすい JSON 形式で受け取ることが可能となる。第二に、設計要素ごとに異なるクエリで RAG 検索を実行するマルチクエリ RAG により、幅広い設計知識を効率的に参照している。第三に、照査計算は数式ベースの決定論的处理で実行し、修正提案のみ LLM が生成するハイブリッドアプローチにより、工学的妥当性と柔軟な修正提案能力を両立している。第四に、LLM が複数の修正案を生成し、各案をシミュレーション評価して最良案を選択する複数候補方式により、効率的な設計改善を実現している。第五に、設計 JSON から中間形式 (Senkei JSON) を経て ifcopenshell²²⁾ により IFC に変換する 2 段階パイプラインにより、BIM 環境との連携を実現している。

本システムの対象は単純桁構造の鋼プレートガーダー橋 (RC 床版合成桁) であり、橋長 20m~70m、幅員 8m~24m 程度を想定している。モデル化する部材は RC 床版、主桁 (I 形断面)、横桁の 3 部材である。鋼プレートガーダー橋としては本来、対傾構や横構も主要な構成部材であるが、本研究では構造を単純化して扱うためこれらは意図的にモデル化の対象外とした。また、現時点では断面は全長一定 (支点部・中央部の変化なし) とし、照査はあくまで概略設計レベル (曲げ・せん断・たわみ・床版厚・腹板幅厚比・横桁配置) に限定している。座屈・疲労等の詳細照査などは今後の課題である。

(2) RAG (検索拡張生成) サブシステム

RAG サブシステムは、道路橋示方書や鋼橋設計の教科書から設計に関連する条文・解説を検索し、Designer の LLM プロンプトに参考文献として提供する機能を担う。本サブシステムは、(1)PDF からのテキスト抽出、(2)テキストのチャンク化、(3)埋め込みベクトル生成、(4)ベクトル検索の 4 つの処理で構成される。

表-2: RAG 知識ベースの対象ドキュメント

文書名	主な内容
第一章 概論	鋼橋の基本概念、設計の考え方
第四章 鋼橋の設計法	荷重・応力計算、許容応力度
第六章 床版	RC 床版の厚さ算定式、最小床版厚
第七章 プレートガーダー橋	主桁断面設計、桁高の目安
道路橋示方書 鋼橋・鋼部材編	設計規準、材料規格

a) 対象ドキュメントとテキスト抽出

本システムでは、鋼プレートガーダー橋の設計に必要な知識を網羅するため、道路橋示方書¹⁶⁾ および「鋼橋設計の基礎」¹⁷⁾ から抽出した 5 種類の PDF ドキュメントを RAG の知識ベースとして使用している。具体的な対象文書を表 2 に示す。

PDF からのテキスト抽出には pdfplumber¹⁸⁾ ライブラリを使用している。pdfplumber は PDF の内部構造を解析してテキストを抽出する Python ライブラリであり、特に表構造の保持に優れている。示方書のような表形式で設計値が示される文書の処理に適しており、表中の数値と周囲のテキストの対応関係を維持したまま抽出できる。元 PDF の例を図 2a に、pdfplumber による抽出結果を図 2b に示す。図 2a では鋼材の物理定数が表形式で記載されており、表中の数値とその見出しの対応関係を正確に抽出することが求められる。図 2b に示すように、pdfplumber による抽出結果では表構造が保持されており、「鋼及び鋳鋼のヤング係数」「 $2.00 \times 10^5 \text{ N/mm}^2$ 」のように、項目名と数値の対応関係が維持されたままテキスト化されている。この特性により、後続のチャンク化・RAG 検索において設計値を正確に参照することが可能となる。なお、一般的に PDF からのテキスト抽出は、フォント埋め込みやレイアウトの複雑さにより困難を伴う場合が多い。pdfplumber についても例外ではなく、図 3a に示す数式を含むページでは、図 3b のようにギリシャ文字や微分記号が正しく抽出されず文字化けが生じるケースがある。このような抽出失敗は、元 PDF におけるフォントの埋め込み方式に起因しており、本システムではこうした不完全な抽出結果もそのままチャンク化・インデックス化しているため、数式を含む設計根拠の検索精度には限界がある点に注意が必要である。なお、本研究の範囲においては、ギリシャ文字を含む複雑な数式を必要としないことが多かったため、この限界を許容している。検索結果で実際に使いたい式がうまく抽出で

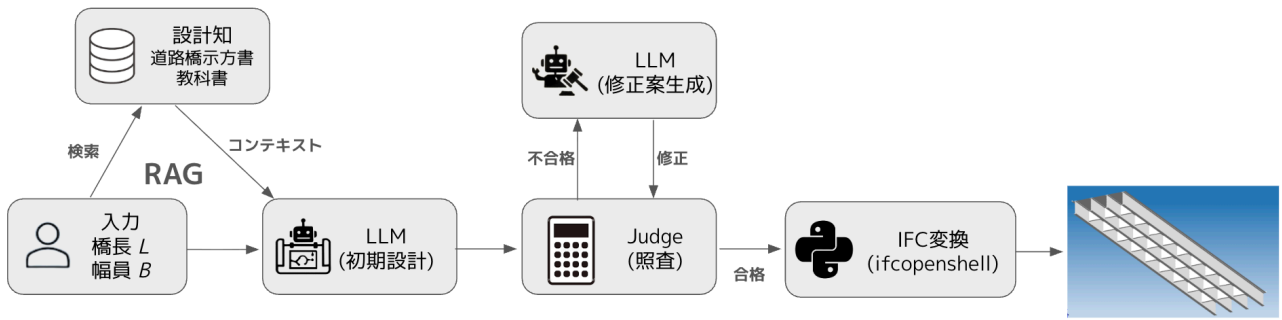


図-1: 提案システムの全体フロー

4.2.2 鋼材の物理定数

- (1) I 編の表-9.1.1 に示す鋼材に関する定数の特性値は表-4.2.1 の値とする。

表-4.2.1 鋼材に関する定数	
鋼 種	定 数
鋼及び鉄鋼のヤング係数	$2.00 \times 10^5 \text{ N/mm}^2$
PC鋼線のヤング係数	$2.00 \times 10^5 \text{ N/mm}^2$
PC鋼より線のヤング係数	$1.95 \times 10^5 \text{ N/mm}^2$
PC鋼棒のヤング係数	$2.00 \times 10^5 \text{ N/mm}^2$
鉄線のヤング係数	$1.00 \times 10^5 \text{ N/mm}^2$
鋼のせん断弾性係数	$7.70 \times 10^4 \text{ N/mm}^2$
鋼及び鉄鋼のポアソン比	0.30
鉄線のポアソン比	0.25

- (2) プレストレスの減少量を算出する場合のPC鋼材の見かけのリラクセーション率は、コンクリートのクリープ、乾燥収縮等の影響を考慮し、その値の信頼性が確保される範囲において適切に定める。ただし、PC鋼材の見かけのリラクセーション率とは、PC鋼材が一定のひずみを保持した状態で、PC鋼材の応力が時間の経過とともに減少する影響と、コンクリートが乾燥収縮、クリープ等により収縮する影響とを考慮して定めるPC鋼材引張力の減少量を、最初に与えたPC鋼材引張力に対する百分率で表した値とする。
- (3) PC鋼材の見かけのリラクセーション率は、表-4.2.2 の値を標準とする。ただし、高温の影響を受ける場合は、蒸気養生を行う場合又は部材上縁に配置されたPC鋼材の純かぶり高が50mm未満で加熱混合型アスファルト舗装を行う場合とする。

(a) 元 PDF (道路橋示方書 鋼橋・鋼部材編 4.2.2 節)

4.2.2 鋼材の物理定数

- (1) I編の表-9.1.1に示す鋼材に関する定数の特性値は表-4.2.1の値とする。

表-4.2.1 鋼材に関する定数

鋼 種	定 数
鋼及び鉄鋼のヤング係数	$2.00 \times 10^5 \text{ N/mm}^2$
PC鋼線のヤング係数	$2.00 \times 10^5 \text{ N/mm}^2$
PC鋼より線のヤング係数	$1.95 \times 10^5 \text{ N/mm}^2$
PC鋼棒のヤング係数	$2.00 \times 10^5 \text{ N/mm}^2$
鉄線のヤング係数	$1.00 \times 10^5 \text{ N/mm}^2$
鋼のせん断弾性係数	$7.70 \times 10^4 \text{ N/mm}^2$
鋼及び鉄鋼のポアソン比	0.30
鉄線のポアソン比	0.25

- (2) プレストレスの減少量を算出する場合のPC鋼材の見かけのリラクセーション率は、コンクリートのクリープ、乾燥収縮等の影響を考慮し、その値の信頼性が確保される範囲において適切に定める。ただし、PC鋼材の見かけのリラクセーション率とは、PC鋼材が一定のひずみを保持した状態で、PC鋼材の応力が時間の経過とともに減少する影響と、コンクリートが乾燥収縮、クリープ等により収縮する影響とを考慮して定めるPC鋼材引張力の減少量を、最初に与えたPC鋼材引張力に対する百分率で表した値とする。
- (3) PC鋼材の見かけのリラクセーション率は、表-4.2.2 の値を標準とする。ただし、高温の影響を受ける場合は、蒸気養生を行う場合又は部材上縁に配置されたPC鋼材の純かぶり高が50mm未満で加熱混合型アスファルト舗装を行う場合とする。

(b) pdfplumber による抽出結果 ((a) と同一ページ)

図-2: 表構造を含む PDF の抽出成功例

きていない場合には、テキストを直接 LLM に渡さずに、該当部分の PDF をマルチモーダルなモデルに渡すことである程度の改善が予想されるが、本研究ではそのような手法は採用していない。

抽出処理では、PDF の各ページを順次処理し、ページ終了ごとに「[Page N]」形式のマーカーを挿入することで、後続のチャンク化処理でページ情報を保持できるようにしている。これにより、設計値の根拠となる文献のページ番号を追跡可能としている。

次に、式 (6.1)～(6.2) に示した断面力は、図 6.3 (b) に示した外力 $pdx dy$ と釣り合わなければならない。そこで、鉛直方向の力のつり合い式 $\Sigma V=0$ を考えると、次式が得られる。

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_z}{\partial y} = -p \quad (6.3)$$

また、yz 平面におけるモーメントのつり合い式 $\Sigma M=0$ を考えると、

$$\frac{\partial m_x}{\partial x} + \frac{\partial m_{yz}}{\partial y} = q, \quad (6.4)$$

になる。xz 平面に関しても、同様に、

$$\frac{\partial m_x}{\partial y} + \frac{\partial m_{yz}}{\partial x} = -q, \quad (6.5)$$

が得られる。

したがって、式 (6.4) を x について、また式 (6.5) を y について 1 回微分し、それらを式 (6.3) に代入すると、次のようにも表現できる。

(a) 元 PDF (鋼橋設計の基礎 第六章)

次に、式 (6.1)～(6.2) に示した断面力は、図 6.3 (b) に示した外力 $pdx dy$ と釣り合わなければならない。そこで、鉛直方向の力のつり合い式 $\Sigma V=0$ を考えると、次式が得られる。

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_z}{\partial y} = -p \quad (6.3)$$

また、yz 平面におけるモーメントのつり合い式 $\Sigma M=0$ を考えると、

$$\frac{\partial m_x}{\partial x} + \frac{\partial m_{yz}}{\partial y} = q \quad (6.4)$$

になる。

xz 平面に関しても、同様に、

$$\frac{\partial m_x}{\partial y} + \frac{\partial m_{yz}}{\partial x} = -q \quad (6.5)$$

が得られる。

したがって、式 (6.4) を x について、また式 (6.5) を y について 1 回微分し、それらを式 (6.3) に代入すると、次のようにも表現できる。

(b) pdfplumber による抽出結果 ((a) と同一ページ)

図-3: 数式を含む PDF の抽出失敗例

b) チャンク化とインデックス構築

テキストのチャンク化は 2 段階で実施される。第一段階では、挿入されたページマーカーに基づいてテキストをページ単位で分割する。第二段階では、各ページのテキストを最大 800 文字ごとに分割する。この文字数は、LLM のコンテキスト長と検索精度のバランスを考慮して設定した値である。長すぎるチャンクは検索精度を低下させ、短すぎるチャンクは文脈情報を失うため、予備実験により 800 文字を採用した。

各チャンクには Pydantic モデル (IndexChunk) で定義されたメタデータが付与される。メタデータには、一意な UUID (id)、元 PDF のファイル名 (source)、章名 (section)、ページ番号 (page, 0 始まり)、チャンク本文 (text) が含まれる。これらのメタデータにより、設計根拠のトレーサビリティを確保している。

埋め込みベクトルの生成には OpenAI Embeddings API¹⁹⁾ の text-embedding-3-small モデルを使用して

いる。

生成された埋め込みベクトルは L2 正規化を施した上で、NumPy²⁰⁾ 配列としてインデックスファイル (embeddings.npy) に保存される。正規化は式 (1) により行う。

$$\mathbf{v}_{\text{norm}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2 + \varepsilon} \quad (1)$$

ここで $\varepsilon = 10^{-8}$ は数値安定性のための微小値である。正規化により、ベクトル間の内積がコサイン類似度と等価になり、検索時の計算を効率化できる。

インデックスは 2 つのファイルで構成される。meta.jsonl は JSON Lines 形式でチャンクのメタデータを格納し、embeddings.npy は $(N_{\text{chunks}}, 1536)$ の形状を持つ NumPy 配列として埋め込み行列を格納する。

c) ベクトル検索

検索時には、クエリ文字列を同じ埋め込みモデルでベクトル化し、保存済みの埋め込み行列との内積を計算する。正規化済みのため、内積はコサイン類似度と等価である。類似度スコア s_i は式 (2) で計算される。

$$s_i = \mathbf{q}_{\text{norm}}^T \mathbf{v}_i \quad (2)$$

ここで \mathbf{q}_{norm} は正規化されたクエリベクトル、 \mathbf{v}_i は正規化済みの i 番目のチャンク埋め込みである。

上位 k 件 (デフォルト $k = 5$) の抽出には NumPy の `argpartition` 関数を使用している。argpartition は $O(N)$ の計算量で k 番目に大きい要素の位置を特定できるため、全チャンクのソート ($O(N \log N)$) を回避して計算効率を確保している。

検索結果は、Pydantic モデル (SearchResult) のリストとして返される。各 SearchResult には、マッチしたチャンク (IndexChunk) と類似度スコア (score) が含まれる。

インデックスは初回ロード後にモジュールレベルのグローバル変数としてメモリ上にキャッシュされ、同一セッション内での複数回の検索において再ロードを回避している。これにより、Designer のマルチクエリ検索 (後述) においても高速な応答を実現している。

(3) Designer (設計生成) サブシステム

Designer サブシステムは、橋長 L と幅員 B を入力として受け取り、RAG で取得した参考文献を基に LLM を用いて鋼プレートガーダー橋の断面設計を自動生成する。本サブシステムの特徴は、(1) 設計要素ごとに異なるクエリで RAG 検索を実行するマルチクエリ RAG、(2) Pydantic スキーマによる型安全な構造化出力、(3) 適用した設計ルールの明示的抽出の 3 点である。

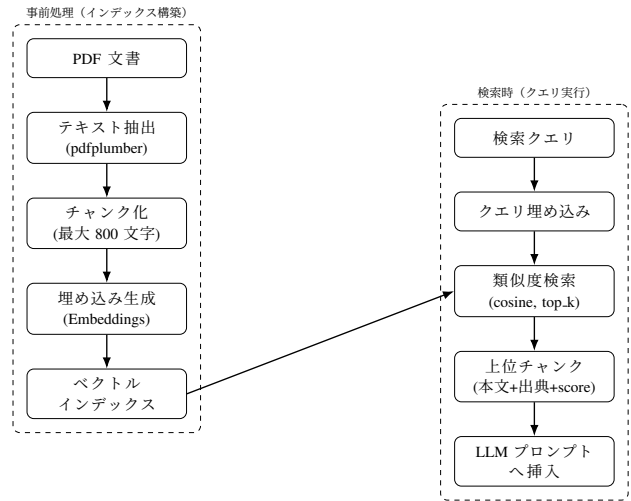


図-4: RAG の検索パイプライン

表-3: BridgeDesign モデルの階層構造

階層 1	階層 2/フィールド	説明
Dimensions	bridge.length	橋長 [mm]
	total.width	全幅 [mm]
	num.girders	主桁本数
	girder.spacing	主桁間隔 [mm]
	panel.length	パネル長 [mm]
	num.panels	パネル数
Sections	girder.standard crossbeam.standard	主桁 I 形断面 横桁 I 形断面
Components	deck.thickness	床版厚 [mm]

a) 入出力仕様

Designer への入力は DesignerInput モデルで定義され、橋長 (bridge.length_m [m]) と幅員 (total.width_m [m]) の 2 パラメータのみを含む。この最小限の入力から、LLM が RAG コンテキストを参照しながら、主桁本数、桁間隔、桁高、板厚、床版厚などの全断面寸法を自動決定する。

出力は BridgeDesign モデルで定義される構造化データである。BridgeDesign の階層構造を表 3 に示す。

主桁断面 (GirderSection) は、腹板高さ、腹板厚、上フランジ幅・厚、下フランジ幅・厚の 6 パラメータで定義される。横桁断面 (CrossbeamSection) は、桁高、腹板厚、フランジ幅・厚の 4 パラメータで定義される。すべての寸法は mm 単位で記録され、Pydantic によるバリデーションにより数値範囲や型の整合性が保証される。

LLM からの直接出力は DesignerOutput モデルで受け取る。このモデルには、BridgeDesign に加えて、設計プロセスの思考・判断根拠を記述した reasoning、適用した設計ルールの一覧である rules (DesignRule のリスト)、および部材間の依存関係を表す depen-

表-4: マルチクエリ RAG の検索クエリ

カテゴリ	クエリ内容 (例)
寸法関連	鋼プレートガーダー橋 橋長 L_m 幅員 B_m 桁配置 主桁本数
主桁配置	並列 I 桁 主桁間隔 幅員と主桁本数の関係
主桁断面	橋長 L_m 主桁断面 桁高 腹板厚さ フランジ
RC 床版	RC 床版合成桁 床版厚さ 最小床版厚
横桁	横桁 対傾構 横構 設計

dependency_rules (DependencyRule のリスト) が含まれる。これにより、生成された設計値だけでなく、その根拠も記録される。

b) マルチクエリ RAG

Designer は設計生成時に、設計要素ごとに異なる 5 種類のクエリで RAG 検索を実行する。表 4 に各クエリの詳細を示す。

各クエリで上位 5 件のチャンクを取得するため、合計最大 25 チャンクの参考文献がプロンプトに含まれる。プロンプトでは、これらのチャンクに rank 番号 (1~25) を付与し、LLM が設計値を決定する際にどのチャンクを根拠としたかを source_hit_ranks として記録させる。マルチクエリ RAG の処理フローを図 5 に示す。

図 6 に、実際の RAG 検索ヒット結果の例を示す。各ヒットには rank 番号、コサイン類似度スコア (score)、出典ドキュメント名 (source)、ページ番号 (page)、および抽出されたテキスト (text) が含まれる。この例では、「鋼橋設計の基礎_第七章 プレートガーダー橋.pdf」や「鋼橋設計の基礎_第一章 概論.pdf」など複数の文献から、プレートガーダー橋の設計に関連するチャンクが類似度順に取得されている。Designer はこれらのヒットを参照しながら設計値を決定し、採用した根拠の rank 番号を source_hit_ranks として記録する。

プロンプトには以下の重要な指示を含めている：

- source_hit_ranks には RAG 検索の hits の rank 番号のみを記載すること
- 根拠が曖昧または見当たらない場合は source_hit_ranks を空リストとし、notes に「仮定」「実務目安」と明記すること
- 主桁本数の候補 {3, 4, 5, 6} を列挙し、各候補について桁間隔と床版厚を評価した上で最終案を選択すること
- 床版厚は 10mm 刻みで設計し、必要厚に余裕を持たせること
- パネル長は {4000, 5000, 6000}mm から選択し、パネル数が整数になるものを優先すること

表-5: DesignRule モデルのフィールド

フィールド	説明
rule_id	ルール ID (R1, R2, ...)
category	カテゴリ (dimensions, girder_section, deck, crossbeam_section, other)
summary	日本語要約 (1~3 文)
condition_expression	条件式
formula_latex	LaTeX 形式の数式表現
applies_to_fields	影響するフィールドパスのリスト
source_hit_ranks	根拠となる RAG ヒットの rank 番号リスト
notes	補足 (仮定の場合はその旨を記載)

- 腹板幅厚比の制約として $t_{web} \geq h_{web}/130$ を満たすこと (例: $h_{web} = 1500 \text{ mm}$ の場合, $t_{web} \geq 1500/130 \approx 11.5 \text{ mm} \rightarrow 12 \text{ mm}$ 以上)。この制約を満たさない場合、Judge 照査の腹板幅厚比で不合格となる

c) 設計ルール抽出

Designer は設計値だけでなく、適用した設計ルールを明示的に抽出する。各 DesignRule には表 5 に示すフィールドが含まれる。

さらに、部材間の依存関係を表す DependencyRule も抽出する。例えば、横桁高さが主桁高さの 80%程度とする場合、対象フィールド (target_field: crossbeam.total_height)、参照フィールド (source_field: girder.web_height)、係数 (factor: 0.8) を記録する。この依存関係ルールは、Judge 修正ループ (後述) において主桁高さが変更された際に横桁高さを自動更新するために使用される。DependencyRule の抽出条件として、RAG コンテキストに「横桁高さは主桁の〇〇%」等の記述がある場合のみ抽出し、係数が読み取れない場合は抽出しないよう指示している。

図 7 に、Designer が実際に抽出した設計ルールの例を示す。reasoning フィールドには LLM の設計方針が記述され、rules リストには個々のルールが構造化されている。例えばルール R1 は RC 床版の最小全厚に関する標準式 ($d = 30L_{\text{support}} + 110$) を示し、applies_to_fields により適用先が deck (床版) の厚さであることが明示されている。また、source_hit_ranks により RAG 検索の rank 17 番のチャンクが根拠であることが追跡可能である。ルール R2 は主桁高さの経験式 ($h \approx L/20$) を示し、notes には文献の示唆内容と採用理由が記録されている。このように、各設計値の決定根拠が構造化データとして保持されるため、設計者による事後検証が容易となる。

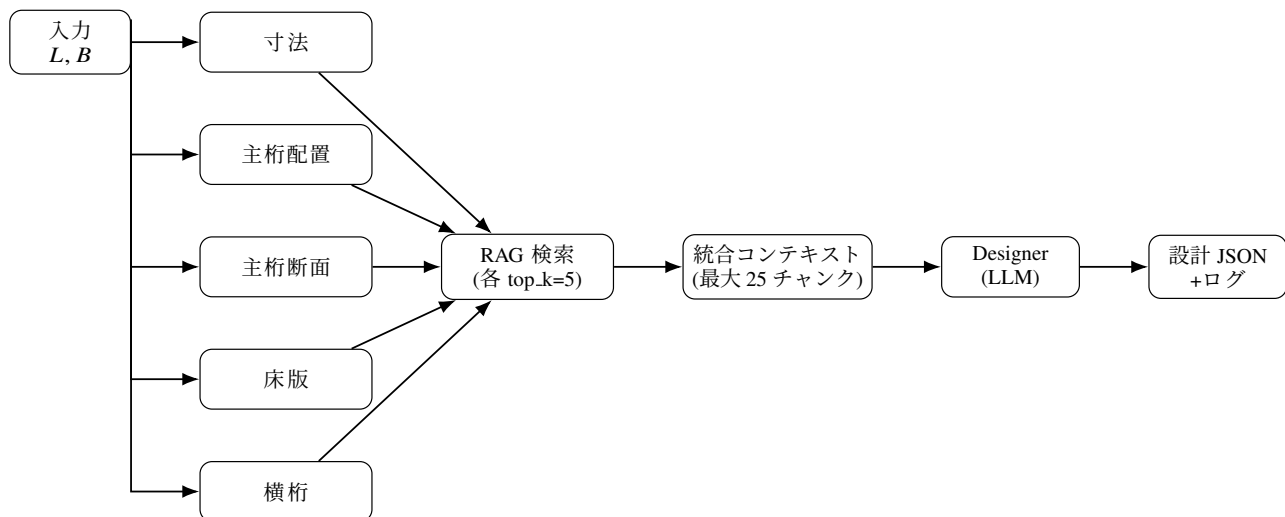


図-5: マルチクエリ RAG (5 観点) と Designer への統合

```

{
  "rank": 3, "score": 0.694,
  "source": "鋼橋設計の基礎第七章_ プレートガーダー橋.pdf",
  "page": 0,
  "text": "7 プレートガーダー橋 7.1 概説プレートガーダー (plate girder) 橋は、橋梁として..."
},
{
  "rank": 4, "score": 0.682,
  "source": "鋼橋設計の基礎第七章_ プレートガーダー橋.pdf",
  "page": 108,
  "text": "7.5 主桁断面の設計 223 7.5 主桁断面の設計プレートガーダー橋の設計手順をまとめて ..."
},
{
  "rank": 5, "score": 0.679,
  "source": "鋼橋設計の基礎第一章_ 概論.pdf",
  "page": 2,
  "text": "1.3 橋梁の種類①腹板 Web plate ②上フランジ Upper flange ③下フランジ Low..."
},
{
  "rank": 6, "score": 0.662,
  "source": "鋼橋設計の基礎第七章_ プレートガーダー橋.pdf",
  "page": 110,
  "text": "7.5 主桁断面の設計 225 なく、数回の試算で逐次収束するようにする。したがって、数値..."
},
{
  "rank": 7, "score": 0.659,
  "source": "鋼橋設計の基礎第七章_ プレートガーダー橋.pdf",
  "page": 60,
  "text": "7.3 断面力の解析 175 本主桁の単純格子桁を考える。まず、曲げによる垂 P=1 直応力度..."
}
}

```

図-6: RAG 検索ヒット結果の例 (rank 3~7)。スコアは小数第 3 位で四捨五入した。text は冒頭のみ表示している。

d) Structured Output による LLM 呼び出し

LLM の呼び出しには OpenAI Responses API²¹⁾ と Structured Output 機能¹³⁾ を使用している。Structured Output は、JSON スキーマまたは Pydantic スキーマを指定することで LLM の出力を型安全な構造化データとして受け取る機能である。

```

"reasoning": "重視点と全体方針:\n- 参考文献では「桁高は支間に対して経験的に  $h/L \approx 1/20$  程度が目安」図(7.134, 主桁高 (以下省略) ",
"rules": [
  {
    "rule_id": "R1",
    "category": "deck",
    "summary": "床版の最小全厚・支間による標準式 (道路橋示方書) RC",
    "condition_expression": "d = 30 * L_support + 110",
    "formula_latex": "d = 30L + 110\\quad\\text{連続版}",
    "applies_to_fields": ["components.deck.thickness"],
    "source_hit_ranks": [17],
    "notes": "道路橋示方書 表-11.5.1 に基づく。床版の最小全厚は160 の下限規定も適用する (参照mm: )。[19]"
  },
  {
    "rule_id": "R2",
    "category": "girder_section",
    "summary": "主桁高さの経験式 (スパン比の目安:  $h/L \approx 1/15 \sim 1/20$ ",
    "condition_expression": "h \approx L / 20",
    "formula_latex": "h \approx L / 20",
    "applies_to_fields": ["sections.girder_standard.web_height", "dimensions.bridge_length"],
    "source_hit_ranks": [13],
    "notes": "文献は  $h/L \approx 1/20$  を示唆しているため、30m 支間では  $h \approx 1500 \text{ mm}$  を一次採用とする。"
  }
],
...
]

```

図-7: Designer が抽出した設計ルール例 (R1: 床版厚, R2: 主桁高さ)。reasoning フィールドは紙面の都合上一部のみ掲載した。

具体的には、DesignerOutput モデルをスキーマとして指定し、LLM がこのスキーマに準拠した JSON を生成するよう制約する。呼び出しは以下の形式で行う：

```

response = client.responses.parse(
    model=model_name,
    input=prompt,

```


表-6: Judge 照査の材料特性デフォルト値

パラメータ	値	説明
E	$2.0 \times 10^5 \text{ N/mm}^2$	ヤング率
鋼種	SM490	降伏点 325 N/mm^2 ($t \leq 16\text{mm}$)
γ_{steel}	$78.5 \times 10^{-6} \text{ N/mm}^3$	鋼の単位体積重量
γ_{concrete}	$25.0 \times 10^{-6} \text{ N/mm}^3$	コンクリートの単位体積重量
α_{bend}	0.6	曲げ許容応力度係数
α_{shear}	0.6	せん断許容応力度係数

text_format=DesignerOutput

)

OpenAI API が自動でバリデーションを行い、response.output_parsed として Pydantic インスタンスを返すため、出力の型安全性が保証される。ここで、Pydantic¹⁵⁾ は Python のデータバリデーションライブラリであり、型ヒントを用いてデータモデルを定義し実行時にバリデーションを行う。Pydantic モデルは JSON スキーマへ自動変換されるため、前述の DesignerOutput モデルをそのまま Structured Output のスキーマとして指定できる。この出力をさらに Python で決定的に IFC に変換することで、山本ら⁸⁾ が報告した IFC テキスト直接生成時のスキーマ違反や構文エラーの問題を回避することが可能になる。

(4) Judge (照査・修正提案) サブシステム

Judge サブシステムは、Designer が生成した BridgeDesign に対して決定論的な照査計算を行い、不合格時には LLM を用いて修正提案 (PatchPlan) を生成する。本サブシステムの特徴は、(1) 照査計算を数式ベースの決定論的処理で実行し LLM を使用しない点、(2) 修正提案のみ LLM が生成するハイブリッドアプローチ、(3) 許可されたアクションの範囲内で修正を提案することで安全性を確保している点、(4) 照査と修正のループにより合格するまで自動で繰り返す点である。

a) 入出力仕様

Judge への入力 は JudgeInput モデルで定義され、BridgeDesign に加えて材料特性と照査パラメータを含む。材料特性は表 6 に示すデフォルト値を持つ。これらの物性値は基本的に道路橋示方書の規定に沿ったものを採用している。

降伏点 f_y は鋼種と板厚に応じて表 7 により決定される。

出力は JudgeReport モデルで定義され、合否判定

表-7: 鋼種と板厚による降伏点

鋼種	$t \leq 16\text{mm}$	$16 < t \leq 40\text{mm}$	$t > 40\text{mm}$
SM400	245 N/mm^2	235 N/mm^2	215 N/mm^2
SM490	325 N/mm^2	315 N/mm^2	295 N/mm^2

(pass_fail), 各照査項目の utilization ratio (Utilization), 中間計算値 (Diagnostics), 修正提案 (PatchPlan) を含む。Utilization には、床版厚 (deck), 曲げ応力度 (bend), せん断応力度 (shear), たわみ (deflection), 腹板幅厚比 (web_slenderness) の各 utilization ratio と、それらの最大値 (max_util), 支配的なチェック項目 (governing_check) が含まれる。

b) 荷重計算

照査計算に先立ち、死荷重と活荷重の断面力を計算する。活荷重には道路橋示方書¹⁶⁾ の B 活荷重に基づく L 荷重を適用する。L 荷重は、部分載荷を全スパン等分布荷重に換算する等価係数 γ を用いて計算される。

まず載荷長 D を $D = \min(10.0, L)[\text{m}]$ で定め、等価係数を式 (3) で算出する。

$$\gamma = \frac{D(2L - D)}{L^2} \quad (3)$$

次に、等価面圧 p_{eq} を式 (4) で計算する。

$$p_{\text{eq}} = p_2 + p_1 \times \gamma \quad (4)$$

ここで、 $p_2 = 3.5 \text{ kN/m}^2$ (支間 80m 以下の場合)、 $p_1 = 10.0 \text{ kN/m}^2$ (曲げ照査用) または $p_1 = 12.0 \text{ kN/m}^2$ (せん断照査用) である。なお、本研究では簡易的な検討を目的としているため、衝撃係数および群集荷重は考慮していない。

各主桁の受け持ち幅 b_i は、端桁と中間桁で異なる。張り出し幅 c を式 (5) で計算する。

$$c = \frac{B - (n_g - 1) \times s_g}{2} \quad (5)$$

ここで B は全幅、 n_g は主桁本数、 s_g は主桁間隔である。端桁の受け持ち幅は $b_i = c + s_g/2$ 、中間桁は $b_i = s_g$ となる。

さらに、主載荷幅 5.5m を考慮した実効幅 b_{eff} を式 (6) で計算する。

$$b_{\text{eff}} = 0.5 \times b_i + 0.5 \times \min(b_i, 5.5) \quad (6)$$

この式は、B 活荷重の主載荷幅 5.5m を最不利に配置した場合の等価的な受け持ち幅を表す。

活荷重による等価線荷重 w_{live} は $w_{\text{live}} = p_{\text{eq}} \times b_{\text{eff}}$ となり、単純桁として曲げモーメント $M_{\text{live}} = w_{\text{live}} L^2/8$ 、せん断力 $V_{\text{live}} = w_{\text{live}} L/2$ を算出する。

死荷重については、床版の RC 重量と主桁の鋼重

表-8: Judge 照査項目

照査項目	判定条件
曲げ応力度	$\sigma/\sigma_{\text{allow}} \leq 1.0$
せん断応力度	$\tau_{\text{avg}}/\tau_{\text{allow}} \leq 1.0$
たわみ	$\delta/\delta_{\text{allow}} \leq 1.0$
床版厚	$t_{\text{req}}/t_d \leq 1.0$
腹板幅厚比	$t_{\text{web,min}}/t_{\text{web}} \leq 1.0$
横桁配置	パネル長×パネル数≒橋長

量を各主桁の受け持ち幅に応じて個別に計算する。床版による線荷重 w_{deck} は式 (7) で計算される。

$$w_{\text{deck}} = \gamma_c \times t_d \times b_i \quad (7)$$

ここで γ_c はコンクリート単位体積重量、 t_d は床版厚である。主桁の鋼重量 w_{steel} は断面積 A_{girder} と鋼の単位体積重量 γ_s から $w_{\text{steel}} = \gamma_s \times A_{\text{girder}}$ として算出する。

各主桁について死荷重と活荷重による断面力を算出し、全主桁の中で最大となる桁 (governing girder) の断面力を照査に使用する。曲げとせん断では支配的な桁が異なる場合があるため、それぞれ独立に評価する。

c) 照査項目

Judge は 6 項目の照査を実行し、各項目について需要と許容値の比 (utilization ratio) を算出する。すべての項目で utilization ratio が 1.0 以下であれば合格とする。なお、あくまで簡易的な照査であり、疲労などの照査や厳密な照査は行っていないことに注意が必要である。表 8 に照査項目の一覧を示す。

なお、本システムでは簡略化のため、合成桁としての床版の剛性寄与を無視している。具体的には、主桁断面諸量 (中立軸位置 \bar{y} , 断面二次モーメント I) を計算する際に、鋼 I 断面のみ (上フランジ・ウェブ・下フランジの 3 矩形) で計算している。RC 床版との合成効果を考慮すると、床版コンクリートが鋼桁と一体となって抵抗するため中立軸が上方に移動し、断面二次モーメントが増加して断面性能が向上する。これを無視することで、保守側 (安全側) の評価となっている。

第一に、曲げ応力度照査を行う。死荷重と活荷重による全曲げモーメント M_{total} に対し、上縁および下縁の曲げ応力度 σ を式 (8) で計算する。

$$\sigma = \frac{M_{\text{total}} \times y}{I} \quad (8)$$

ここで、 y は中立軸から上縁または下縁までの距離、 I は断面二次モーメントである。許容曲げ応力度 σ_{allow} は式 (9) で計算する。

$$\sigma_{\text{allow}} = \alpha_{\text{bend}} \times f_y \quad (9)$$

表-9: 許容たわみの算定式

支間長	許容たわみ
$L \leq 10\text{m}$	$L/2000$
$10\text{m} < L \leq 40\text{m}$	$L^2/20000 \text{ [mm]}$
$L > 40\text{m}$	$L/500$

上フランジと下フランジでは板厚が異なる場合があるため、それぞれの降伏点から許容値を算出し、上縁と下縁それぞれについて utilization ratio を評価する。

第二に、せん断応力度照査を行う。平均せん断応力度 τ_{avg} を式 (10) で計算する。

$$\tau_{\text{avg}} = \frac{V_{\text{total}}}{t_{\text{web}} \times h_{\text{web}}} \quad (10)$$

許容せん断応力度 τ_{allow} は式 (11) で計算する。

$$\tau_{\text{allow}} = \alpha_{\text{shear}} \times \frac{f_y}{\sqrt{3}} \quad (11)$$

第三に、たわみ照査を行う。活荷重によるたわみ δ を式 (12) で計算する。

$$\delta = \frac{5w_{\text{live}}L^4}{384EI} \quad (12)$$

許容たわみ δ_{allow} は道路橋示方書に基づき、支間長に応じて表 9 のように設定する。

第四に、床版厚照査を行う。要求床版厚 t_{req} を式 (13) で算出する。

$$t_{\text{req}} = \max(30L_{\text{support}} + 110, 160) \text{ [mm]} \quad (13)$$

ここで L_{support} は床版支間 (主桁間隔) [m] である。設計床版厚がこれ以上であることを確認する。

第五に、腹板幅厚比照査を行う。SM490 鋼の場合、幅厚比の制限値は 130 であり、要求腹板厚 $t_{\text{web,min}}$ を式 (14) で算出する。

$$t_{\text{web,min}} = \frac{h_{\text{web}}}{130} \quad (14)$$

設計腹板厚がこれ以上であることを確認する。

第六に、横桁配置照査を行う。パネル長とパネル数の積が橋長とほぼ一致すること (許容誤差 1.0mm 以内)、およびパネル長が 20m 以下であることを確認する。

照査結果は JudgeReport として JSON 形式で出力される。図 8 に不合格時の出力例を示す。pass_fail は合否判定、utilization は各照査項目の utilization ratio と支配的な照査項目 (governing.check), diagnostics は照査計算の中間値 (断面力・応力度・たわみ等) を記録する。この例では曲げの utilization ratio が 1.33 と 1.0 を超過しており、曲げが支配的照査項目となっている。この照査結果が PatchPlan 生成

```

"pass_fail": false,
"utilization": {
  "deck": 0.950,
  "bend": 1.332,
  "shear": 0.229,
  "deflection": 1.193,
  "web_slenderness": 0.962,
  "max_util": 1.332,
  "governing_check": "bend"
},
"diagnostics": {
  "M_total": 7789900000.000,
  "V_total": 825662.5,
  "ybar": 869.819,
  "moment_of_inertia": 36675237661.791,
  "y_top": 1185.181,
  "y_bottom": 869.819,
  "sigma_top": 251.735,
  "sigma_bottom": 184.751,
  "tau_avg": 25.802,
  "delta": 95.444,
  "delta_allow": 80.0,
  ...
}

```

図-8: Judge 照査結果の出力例（不合格時）。数値は小数第3位で四捨五入した。

表-10: 許可される修正アクション

アクション	許可される変更量
腹板高さ増加	+100, +200, +300, +500 mm
腹板厚増加	+2, +4, +6 mm
上フランジ厚増加	+2, +4, +6 mm
下フランジ厚増加	+2, +4, +6 mm
上フランジ幅増加	+50, +100 mm
下フランジ幅増加	+50, +100 mm
床版厚を要求値に設定	（自動計算）
横桁配置の修正	（自動計算）

時に LLM へ入力され、修正方針の判断根拠となる。

d) PatchPlan 生成

照査で不合格となった場合、LLM が修正提案（PatchPlan）を生成する。PatchPlan は、許可されたアクション（PatchAction）の組み合わせで構成される。各 PatchAction には操作種別（op）、対象フィールドパス（path）、変更量（delta_mm）、変更理由（reason）が含まれる。

許可されるアクションを表 10 に示す。各アクションには許可される変更量が事前に定義されており、LLM はこの範囲内でのみ修正を提案できる。これにより、過度な修正や不適切な設計変更を防止している。

LLM へのプロンプトでは、照査結果（各項目の utilization ratio、支配的な照査項目）と現在の設計値を提示し、max_util（全項目の utilization ratio の最大値）を 1.0 以下（できれば 0.98 以下）にするための修正案を要求する。プロンプトには判断の方針として以下を示している：

```

{
  "actions": [
    {
      "op": "increase_top_flange_thickness",
      "path": "sections.girder_standard.top_flange_thickness",
      "delta_mm": 6.0,
      "reason": "上フランジの断面係数を大きく増やし、sigma_top を直接低減するため。"
    },
    {
      "op": "increase_top_flange_width",
      "path": "sections.girder_standard.top_flange_width",
      "delta_mm": 100.0,
      "reason": "上フランジ幅を拡大して断面係数をさらに増し、曲げ支配をフランジ強化で解消するため。"
    }
  ]
}

```

図-9: PatchPlan の出力例（曲げ支配時）

- 曲げが支配的な場合：フランジ厚または腹板高さの増加を優先
- せん断が支配的な場合：腹板厚の増加を優先
- たわみが支配的な場合：断面二次モーメントを増やす方向（腹板高さ増加）を優先
- 床版厚が支配的な場合：床版厚を必要厚に設定するアクションを必ず使用

図 9 に、図 8 の不合格結果に対して LLM が生成した PatchPlan の出力例を示す。曲げが支配的照査項目であるため、上フランジ厚を 6mm 増加させるアクションと上フランジ幅を 100mm 拡大するアクションの 2 つが提案されている。各アクションには対象フィールドのパス、変更量、および変更理由が自然言語で記述されており、LLM が照査結果を解釈して適切な修正方針を導出していることがわかる。

また、本システムでは複数候補方式を採用している。LLM が最大 3 つの異なるアプローチ（例：腹板高さ重視、フランジ厚重視、バランス型など）による修正案を同時に生成し、各案を仮適用して照査計算をシミュレーションする。シミュレーションは決定論的な照査計算を再実行するのみであり、LLM は使用しない。そして、改善度（improvement = 現在の max_util - シミュレーション後の max_util）が最大となる案を選択する。この方式により、LLM の提案の不確実性を軽減し、効率的な設計改善を実現している。

評価済み候補は EvaluatedCandidate モデルとして記録され、各候補の PatchPlan、シミュレーション後の max_util、シミュレーション後の Utilization、改善量が含まれる。これにより LLM の提案がどのように評価されたかを追跡できる。図 10 に評価済み候補の出力例を示す。candidate には PatchPlan とアプローチの要約（approach_summary）が含まれ、simulated_max_util と simulated_utilization

```

{
  "candidate": {
    "plan": {
      "actions": [
        {
          "op": "increase_top_flange_thickness",
          "path": "sections.girder_standard.top_flange_thickness",
          "delta_mm": 6.0,
          "reason": "上フランジの断面係数を大きく増やし、sigma_top を直接低減するため。"
        },
        {
          "op": "increase_top_flange_width",
          "path": "sections.girder_standard.top_flange_width",
          "delta_mm": 100.0,
          "reason": "上フランジ幅を拡大して断面係数をさらに増し、曲げ支配をフランジ強化で解消するため。"
        }
      ]
    },
    "approach_summary": "支配: 曲げ (上フランジ)。方針: フランジ強化 (厚さ・幅) で断面係数を大幅増..."
  },
  "simulated_max_util": 1.053,
  "simulated_utilization": {
    "deck": 0.950,
    "bend": 1.053,
    "shear": 0.231,
    "deflection": 1.010,
    "web_slenderness": 0.962,
    "max_util": 1.053,
    "governing_check": "bend"
  },
  "improvement": 0.279
}

```

図-10: 評価済み候補 (EvaluatedCandidate) の出力例。数値は小数第 3 位で四捨五入した。

にはこの修正案を仮適用した場合の照査シミュレーション結果が記録される。この例では、修正前の max_util が 1.33 (図 8) であったのに対し、シミュレーション後は 1.05 に低下しており、improvement として 0.28 の改善が得られている。なお、たわみの utilization ratio が 1.00 とぎりぎりであり、1 回の修正では完全に合格に至っていないことも読み取れる。このような場合、修正ループが継続して追加の修正を行う。

e) 修正ループ

Designer-Judge 修正ループの処理フローを図 11 に示す。

まず Designer が初期設計を生成し、Judge が照査を実行する。合格 ($\text{max_util} \leq 1.0$ かつ横桁配置 OK) であれば終了する。不合格の場合、LLM が PatchPlan を生成し、設計に適用する。さらに、Designer が抽出した依存関係ルール (例: 横桁高さが主桁高さの 80%) を適用し、部材間の整合性を維持する。その後、再度 Judge が照査を実行する。このループを合格するまで、または最大修正回数 (デフォルト 5 回) に達するまで繰り返す。

修正ループの結果は RepairLoopResult モデルとして記録される。これには、収束フラグ (converged:

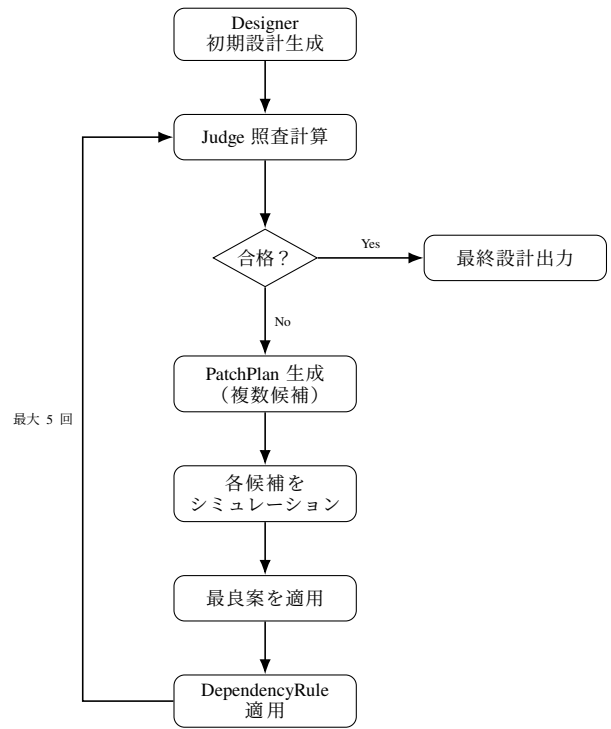


図-11: Designer-Judge 修正ループの処理フロー

合格に到達したか)、全修正ループの履歴 (iterations: RepairIteration のリスト)、最終設計 (final.design: BridgeDesign)、最終照査結果 (final.report: JudgeReport)、および RAG 検索ログ (rag.log: DesignerRagLog) が含まれる。各 RepairIteration にはイテレーション番号、その時点の BridgeDesign、JudgeReport が記録される。これにより、設計改善のプロセス全体を追跡可能としている。

(5) IFC 変換サブシステム

IFC 変換サブシステムは、Designer が生成した (または Judge 修正後の) BridgeDesign JSON を IFC (Industry Foundation Classes) 形式に変換し、BIM/CIM ソフトウェアで利用可能な 3D モデルを出力する。変換は 2 段階のパイプラインで行われる (図 12)。第一段階で BridgeDesign JSON を中間形式である Senkei JSON に変換し、第二段階で Senkei JSON を ifcopen-shell²²⁾ により IFC ファイルに変換する。この 2 段階方式により、橋梁 BIM の既存エコシステムとの接続性を確保しつつ、将来的な拡張にも対応可能な設計としている。

a) BridgeDesign から Senkei JSON への変換

Senkei JSON は、橋梁の 3 次元形状を線形 (Senkei) ベースで定義する中間形式である。BridgeDesign の断面諸元から、主桁・横桁・床版の 3 次元位置を具体的な座標値として算出する。Senkei JSON の主要構成要素を表 11 に示す。



図-12: IFC 変換の 2 段階パイプライン

表-11: Senkei JSON の主要構成要素

要素	説明
Infor	橋梁基本情報（名称等）
Senkei	線形データ（各主桁の上下フランジ端点座標）
MainPanel	主桁パネル（ウェブ、上フランジ、下フランジ）
Yokogeta	横桁（隣接主桁間を接続）
Shouban	床版パネル

主桁の配置では、まず全主桁の総スパン $(n_g - 1) \times s_g$ を計算し、全幅との差から張り出し幅（overhang）を式 (5) により求める。各主桁の幅員方向位置（Y 座標）は $\text{overhang} + i \times s_g$ ($i = 0, 1, \dots, n_g - 1$) となる。

各主桁について 6 本の線形を定義する。これらは上フランジの左端 (TG*L)、中央 (TG*), 右端 (TG*R)、下フランジの左端 (BG*L)、中央 (BG*), 右端 (BG*R) に対応する。各線形は橋軸方向に並ぶ点列 (SenkeiPoint) で構成され、各点には断面名 (S1, C1, C2, ..., E1), X 座標 (橋軸方向位置), Y 座標 (幅員方向位置), Z 座標 (鉛直方向位置) が含まれる。本実装では、下フランジ中心線 (BG*) を $Z = 0$ とし、上フランジ中心線 (TG*) の Z 座標はウェブ高 h_{web} となる。図 13 に、第 4 主桁の上フランジ右端 (TG4R) の線形データの出力例を示す。始端 S1 ($X = 0$) からパネル長 5000mm ごとに C1, C2, C3 と点列が定義されており、各点が同一の Y 座標・Z 座標を持つ直線桁であることがわかる。

橋軸方向 (X 座標) の分割は、パネル長に基づいて行う。始端 (S1, $x = 0$) から終端 (E1, $x = L$) まで、パネル長ごとに断面位置 (C1, C2, ...) を設定する。断面名リストは例えば「S1, C1, C2, ..., C9, E1」のように生成される (パネル数 10 の場合)。

```

{
  "Name": "TG4R",
  "Point": [
    { "Name": "S1", "X": 0.0, "Y": 8251.0, "Z": 2000.0 },
    { "Name": "C1", "X": 5000.0, "Y": 8251.0, "Z": 2000.0 },
    { "Name": "C2", "X": 10000.0, "Y": 8251.0, "Z": 2000.0 },
    { "Name": "C3", "X": 15000.0, "Y": 8251.0, "Z": 2000.0 },
    ...
  ]
}

```

図-13: Senkei JSON における線形点列 (SenkeiPoint) の出力例

横桁は、隣接する主桁間を接続する形で配置される。横桁の橋軸方向位置はパネル境界位置 (C1, C2, ...) に一致し、両端部 (S1, E1) を除く $n_{\text{panels}} - 1$ 箇所に配置される。横桁の命名規則は「CB.G{ 左桁番号 }_G{ 右桁番号 }_C{ 断面番号 }」であり、例えば「CB.G1_G2_C1」は第 1 主桁と第 2 主桁の間の C1 位置に配置される横桁を表す。各横桁の断面諸元 (高さ、腹板厚、フランジ寸法) は BridgeDesign の crossbeam_standard から取得する。

床版は、橋梁の四隅を頂点とする矩形として定義される。line 属性には床版の外周を定義する 4 本の線形名 (TG1L, TG1R, TG*R, TG*L) が指定される。厚さは BridgeDesign の deck.thickness から取得し、overhang の値も記録して床版端部の位置を明確にする。

b) 部材の厚み方向分割

Senkei JSON への変換では、将来的な維持管理情報の格納を考慮し、各部材を厚み方向に分割して個別の要素として出力する。具体的には、主桁パネル (ウェブ、上フランジ、下フランジ) はそれぞれ板厚を 2 等分した 2 枚の板要素に分割される。例えば板

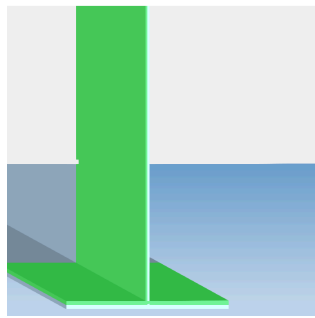


図-14: 厚み方向に2分割された主桁部材の BIM モデル表示例

厚 t のウェブは、厚さ $t/2$ の2枚のパネルとして記録される。床版についても同様に厚み方向に2分割される。

この分割により、IFC モデル上の各要素が橋梁部材の表裏面に対応するため、点検記録や劣化情報を部材の表裏ごとに個別に紐付けることが可能となる。これは BIM/CIM を活用した維持管理において、損傷の位置を3次元モデル上で特定し、補修履歴を蓄積するための基盤となる。図14に、厚み方向に2分割された主桁部材の BIM モデル表示例を示す。画像において、ウェブの左半分と下フランジの上半分が選択されて緑色に表示されており、厚さ方向に分割されていることが確認できる。なお、ビューアには Open IFC Viewer を使用している。

c) Senkei JSON から IFC への変換

第二段階では、ifcopenshell²²⁾ ライブラリを使用して Senkei JSON を IFC ファイルに変換する。ifcopenshell は Python バインディングを提供するオープンソースの IFC 操作ライブラリであり、buildingSMART⁴⁾ が策定する IFC2x3 および IFC4x3 の両方に対応している。本システムでは IFC4X3 スキーマを使用する。

IFC (Industry Foundation Classes) は建設プロジェクトのデータ交換のためのオープン標準であり、3次元形状と属性情報を統合的に記述できる。IFC ファイルは ISO 10303-21 形式 (STEP 物理ファイル形式) のテキストファイルとして出力される。

変換処理では、まず IFC の階層構造をセットアップする。IFC モデルは図15に示す階層構造を持つ。

次に、Senkei JSON の各要素に対応する IFC エンティティに変換する。表12に要素と IFC エンティティの対応を示す。

床版は IfcBeam エンティティとして生成し、形状表現には Brep (Boundary Representation: 境界表現) を使用する。床版の外形を定義する頂点座標から閉じた多面体 (IfcFacetedBrep) を生成することで直方体を表現する。床版は厚さ方向 (T)、橋軸方向 (X)、

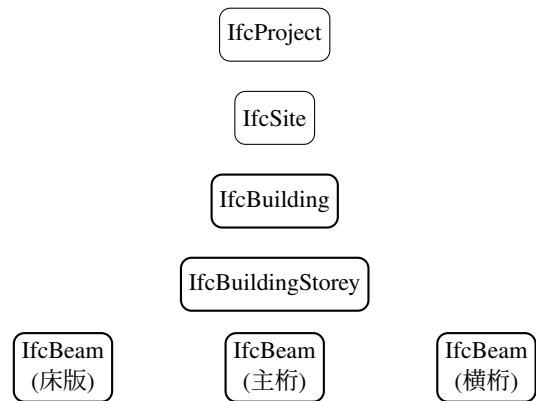


図-15: IFC モデルの階層構造

表-12: 構造要素と IFC エンティティの対応

構造要素	IFC エンティティ	形状表現
床版	IfcBeam	Brep (境界表現)
主桁	IfcBeam	Brep (境界表現)
横桁	IfcBeam	Brep (境界表現)

幅員方向 (Y) の3軸で分割され、各セグメントには Deck.T{ 厚み番号 }_X{ 橋軸番号 }_Y{ 幅員番号 } の命名規則で一意な名前を付与する。

主桁も IfcBeam エンティティとして生成する。形状表現には Brep (境界表現) を使用し、ウェブ・上フランジ・下フランジの各板要素を個別の IfcFacetedBrep として生成する。各主桁パネルには以下の命名規則で一意な名前を付与する：

- ウェブ：G{ 桁番号 }B{ ブロック番号 }W
- 上フランジ：G{ 桁番号 }B{ ブロック番号 }UF
- 下フランジ：G{ 桁番号 }B{ ブロック番号 }LF

横桁も同様に IfcBeam エンティティと Brep 形状表現で生成する。横桁は隣接する主桁間を橋軸直角方向 (幅員方向) に接続する形で配置される。主桁と同様に、横桁もウェブ・上フランジ・下フランジの3要素に分割し、さらに厚さ方向の分割を行う。各要素には CB{ 横桁番号 }_{W/UF/LF}_T{ 厚み番号 }_X{ 橋軸番号 } の命名規則で一意な名前を付与する。

生成された IfcBeam 要素は IfcRelContainedInSpatialStructure を通じて IfcBuildingStorey に格納される。なお、材料情報 (SM490A 等) は Senkei JSON に記録されているが、現時点では IFC エンティティ (IfcMaterial 等) への変換は行っていない。

図16に、出力された IFC ファイルの実データ抜粋を示す。この例は主桁1の第1ブロックウェブ (G1B1W.T0.X0) を構成するエンティティ群であり、色定義 (IFCCOLOURRGB) → 表面スタイル (IFCSURFACESTYLE) → 形状表現 (IFCSHAPEREPRE-

```
#717=IFCCOLOURRGB($,0.675,0.812,0.925);
#718=IFCSURFACESTYLESHADING(#717,$);
#719=IFCSURFACESTYLE($,.BOTH.,(#718));
#720=IFCSTYLEDITEM(#56,(#719),$);
#721=IFCSHAPEREPRESENTATION(#12,'Body','Brep',(#56))
;
#722=IFCPRODUCTDEFINITIONSHAPE($,$,(#721));
#723=IFCCARTESIANPOINT((0.,0.,0.));
#724=IFCAXIS2PLACEMENT3D(#723,$,$);
#725=IFCLOCALPLACEMENT($,#724);
#726=IFCBEAM('1acce528-f9dd-11f0-9600-9626031620ab',
$, 'G1B1W_T0_X0', $,$,#725,#722,$,.USERDEFINED.);
#727=IFCRELCONTAINEDINSPATIALSTRUCTURE(
'1acce6a4-f9dd-11f0-9600-9626031620ab',
$,$,$,(#726),#6);
```

図-16: 出力 IFC ファイルの実データ抜粋（主桁ウェブ要素）。RGB 値は小数第 3 位で四捨五入した。

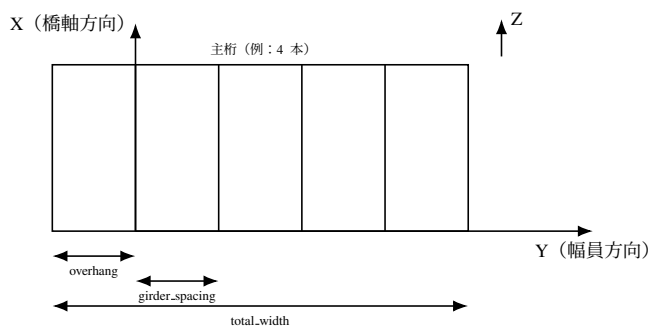


図-17: 座標系と主桁配置の模式図（平面図）

SENTATION, Brep) → 配置 (IFCLOCALPLACEMENT) → 要素本体 (IFCBEAM) → 空間格納 (IFCRELCONTAINEDINSPATIALSTRUCTURE) という積み上げ構造で IFC エンティティが構成されている。IFCBEAM の名称「G1B1W_T0_X0」は、主桁 1 (G1)、ブロック 1 (B1)、ウェブ (W)、厚み分割区分 (T0)、橋軸方向セグメント (X0) を表す命名規則に従っている。最終行の IFCRELCONTAINEDINSPATIALSTRUCTURE により、この要素が IfcBuildingStorey (#6) に紐付けられ、空間階層と幾何情報が Relationship (Rel) エンティティで結ばれるという IFC の基本構造が実現されている。

d) 座標系と出力形式

3D モデルの座標系は、X 軸を橋軸方向（橋長方向）、Y 軸を橋軸直角方向（幅員方向）、Z 軸を鉛直方向（上向き正）とする右手系である。本実装では、 $X = 0$ を始端断面 (S1)、 $Y = 0$ を第 1 主桁中心線 (TG1/BG1 の中央)、 $Z = 0$ を下フランジ中心線 (BG*) に設定している。このため、主桁線形 (SenkeiPoint) の Z 座標は原則として $Z \geq 0$ で表現される。ただし、板要素は中心線から厚さ方向に押し出してソリッド化するため、下フランジ等では $-t/2$ 程度の小さな負の Z が現れる場合がある。座標系と主桁配置の模式図を図 17 に示す。

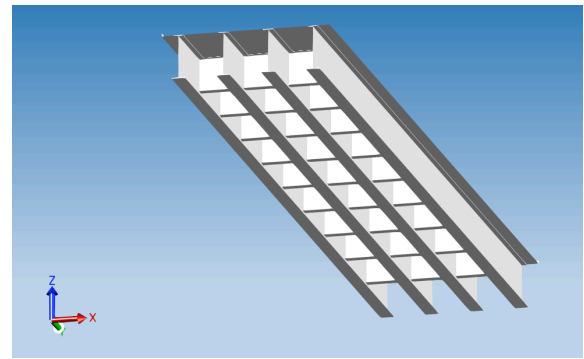


図-18: BIM ビューワーで表示した IFC モデル

生成される IFC ファイルは IFC2x3 形式または IFC4x3 形式で出力可能であり、BIMvision, Autodesk Revit, FreeCAD, Blender (BIM Add-on 使用) などの汎用 BIM ビューワーで即座に可視化・検証できる。これにより、生成された設計が意図通りの形状となっているかを視覚的に確認でき、さらに BIM/CIM ワークフローへの統合が可能となる。図 18 に生成された IFC モデルの BIM ビューワーでの表示例を示す。

4. 評価実験

(1) 実験設定

提案システムの有効性を検証するため、橋長 L と幅員 B の組み合わせを変化させた 192 件の評価実験を実施した。

a) 評価ケースの構成

評価ケースは、32 種類の $L \times B$ 条件に対して RAG 有無の 2 水準、各条件 3 試行の計 $32 \times 2 \times 3 = 192$ 件である。橋長は 20m～70m の 11 段階 (5m 刻み)、幅員は 8m, 10m, 12m, 16m, 20m, 24m の 6 段階を設定した。ただし、すべての $L \times B$ の組み合わせではなく、実務的に想定される 32 条件を選定した (表 13)。短スパン ($L = 20 \sim 45$ m) では狭幅員 ($B = 8 \sim 20$ m)、長スパン ($L = 50 \sim 70$ m) では広幅員 ($B = 10 \sim 24$ m) を中心に選定しており、これは実際の道路橋設計において、短い支間では比較的狭い幅員の橋梁が多く、長支間では広幅員となる傾向を反映したものである。

b) 実験パラメータ

LLM には gpt-5.1 を使用した。Designer-Judge の修正ループにおける最大修正回数は 5 回とした。すなわち、Designer が初期設計を生成した後、Judge による照査で不合格となった場合、最大 5 回まで修正と再照査を繰り返す。5 回の修正で合格に至らなかった場合は不合格として記録する。各条件について 3 試行を実施しているのは、LLM の出力の確率的な変

表-13: 評価対象とした $L \times B$ の組み合わせ (✓ が評価対象)

L [m] \ B [m]	8	10	12	16	20	24
20	✓	✓				
25	✓	✓	✓			
30	✓	✓	✓			
35	✓	✓		✓		
40	✓	✓			✓	
45	✓	✓			✓	
50		✓		✓		✓
55		✓		✓		✓
60		✓		✓		✓
65			✓	✓		✓
70			✓	✓		✓

表-14: 評価に用いる照査項目

照査項目	内容
deck	床版厚チェック (必要厚 \leq 実厚)
bend	曲げ応力チェック ($\sigma \leq \sigma_{\text{allow}}$)
shear	せん断応力チェック ($\tau \leq \tau_{\text{allow}}$)
deflection	たわみチェック ($\delta \leq \delta_{\text{allow}}$)
web_slenderness	腹板幅厚比チェック
crossbeam_layout	横桁配置チェック (パネル長 \times パネル数 \approx 橋長)

動を考慮し、結果の安定性を評価するためである。

RAG 有無の 2 水準は、RAG を有効にした場合（道路橋示方書等の関連条文をプロンプトに付与）と無効にした場合（関連条文なしで LLM の事前学習知識のみに依拠）を比較するものであり、RAG が設計品質に与える効果を定量的に評価することを目的としている。

(2) 評価指標

a) 照査項目

Judge サブシステムが実行する照査項目は表 14 に示す 6 項目である。各項目について合格 (pass) または不合格 (fail) を判定し、すべての項目に合格した場合のみ設計全体を合格とする。なお、あくまで簡易的な照査であり、実際の設計業務における詳細な照査を代替するものではない。

b) 評価指標の定義

評価実験では以下の指標を用いて提案システムの性能を評価する。

- **最終合格率 (収束率)**：全試行に対する、最大修正回数 (5 回) 以内に全照査項目に合格した試行の割合 (本研究では収束率と同義)。RAG 有無・橋長・幅員ごとに集計する。
- **チェック項目別初回合格率**：各照査項目について、初回設計 (iter0) で合格した試行の割合。どの照査項目が初期設計の困難さに影響しているかを分析する。

表-15: 全体・RAG 有無別の主要指標

指標	全体	RAG あり	RAG なし
総試行数	192	96	96
初回合格率	18.8%	37.5%	0.0%
収束率	80.2%	85.4%	75.0%
平均修正回数	1.93	1.40	2.53

て、初回設計 (iter0) で合格した試行の割合。どの照査項目が初期設計の困難さに影響しているかを分析する。

- **収束までの修正回数**：合格に至った試行について、Designer-Judge の修正ループが何回で収束したかを集計する。修正回数が少ないほど、初期設計の品質が高いことを示す。

(3) 結果

a) 全体サマリー

表 15 に全体・RAG 有無別の主要指標を示す。RAG ありの場合、初回合格率は 37.5% であり、RAG なしの 0.0% と比較して大幅に高い。収束率 (最大 5 回の修正で全項目合格に至る割合) は RAG ありで 85.4%、RAG なしで 75.0% であった。また、収束ケースにおける平均修正回数は RAG ありで 1.40 回、RAG なしで 2.53 回であり、RAG により初期設計品質が向上していることが確認される。

RAG なしでは 96 件全てが初回不合格であり、本評価条件においては LLM の事前学習知識のみでは初回から照査を通過する設計を生成できないことが示された。ただし、各条件の試行数は 3 回であり、低頻度で初回合格が生じる可能性は排除できない。

b) チェック項目別の初回合格率

表 16 にチェック項目別の初回合格率を示す。shear (せん断) および web_slenderness (腹板幅厚比) は全ケースで初回合格率 100% であった。本評価の橋長・幅員範囲 ($L = 20\text{--}70\text{ m}$, $B = 8\text{--}24\text{ m}$) の単純桁について、提案手法が生成した設計案では、最大利用率を与える (最もきびしい) 照査項目は曲げ・たわみ等であり、せん断が支配となる例はなかった。なお、単純桁では支点付近でせん断が卓越し得るため、せん断照査は全ケースで実施し、いずれも満足した。deck (床版厚) は RAG ありで 100%、RAG なしで 66.7% であった。deflection (たわみ) は RAG の効果が最も顕著であり、RAG ありで 51.0% に対し RAG なしで 6.2% と、44.8 ポイントの差が生じた。bend (曲げ) は RAG ありで 46.9%、RAG なしで 13.5% であった。

初回不合格ケースの不合格項目パターンを表 17 および表 18 に示す。RAG あり・初回不合格 60 件

表-16: チェック項目別の初回合格率

チェック項目	全体	RAG あり	RAG なし	RAG 効果
shear	100.0%	100.0%	100.0%	±0.0%
web_slenderness	100.0%	100.0%	100.0%	±0.0%
deck	83.3%	100.0%	66.7%	+33.3%
bend	30.2%	46.9%	13.5%	+33.4%
deflection	28.6%	51.0%	6.2%	+44.8%

表-17: RAG あり：初回不合格パターン（60 件）

パターン	件数	割合
bend + deflection	38	63.3%
bend のみ	13	21.7%
deflection のみ	9	15.0%

表-18: RAG なし：初回不合格パターン（96 件）

パターン	件数	割合
bend + deflection	50	52.1%
deck + bend + deflection	28	29.2%
deflection のみ	11	11.5%
bend のみ	3	3.1%
deck + bend	2	2.1%
その他	2	2.1%

では、bend+deflection の同時不合格が 63.3%を占め、bend のみが 21.7%、deflection のみが 15.0%であった。RAG なし・初回不合格 96 件では、bend+deflection が 52.1%、deck+bend+deflection が 29.2%を占めた。RAG ありでは deck 不合格が完全に解消されている点が特徴的である。

c) 橋長別・幅員別の傾向

図 19 に橋長別の結果を示す。RAG ありでは $L=45\text{ m}$ で最高の初回合格率 66.7%を記録した。長スパン ($L = 65\text{--}70\text{ m}$) でも RAG ありは初回合格率 44–56%を維持している一方、RAG なしは全橋長で初回合格率 0%であった。収束率は橋長が長くなるほど低下する傾向があり、特に RAG なしでは $L = 65\text{ m}$ で 11%まで低下した。

図 20 に幅員別の結果を示す。 $B = 8\text{ m}$ では RAG ありの初回合格率が 66.7%と最高であり、 $B = 8\text{--}16\text{ m}$ では RAG ありの収束率は 94–100%であった。一方、 $B = 20\text{--}24\text{ m}$ では難易度が顕著に上昇し、 $B = 24\text{ m}$ では RAG なしの収束率が 7%（15 件中 1 件のみ合格）まで低下した。 $B = 20\text{ m}$ では RAG なしの収束率が RAG ありを上回る逆転現象（67% vs 50%）が観測された。設計ログを確認すると、 $L = 40\text{ m}$ 、 $B = 20\text{ m}$ の条件において RAG ありでは 4 本桁（桁間隔 6,000 mm）を選択する傾向があったのに対し、RAG なしでは 5–6 本桁を選択していた。RAG が提供する h/L ルールはウェブ高の設計に効果的であるが、桁本数につ

表-19: $L = 40\text{ m}$ 、 $B = 20\text{ m}$ における初回設計の桁本数と収束結果

ケース	桁本数	桁間隔 (mm)	収束
RAG あり trial 1	4	6,000	×
RAG あり trial 2	4	6,000	×
RAG あり trial 3	5	4,500	○
RAG なし trial 1	5	4,500	○
RAG なし trial 2	5	4,375	○
RAG なし trial 3	6	3,600	○

表-20: 不収束ケースの不合格項目内訳（ $n = 38$ ）

不合格項目	件数	割合
bend のみ	22	57.9%
bend + web_slenderness	13	34.2%
web_slenderness のみ	3	7.9%
合計	38	100.0%

いての明確なガイダンスを含まないため、RAG ありでは「少ない桁本数で桁高を確保」という設計方針となりやすい。しかし $B = 20\text{ m}$ では桁本数増加による荷重分散の方が収束に有効であり、結果的に RAG なしケースが収束しやすかった。表 19 に $L = 40\text{ m}$ 、 $B = 20\text{ m}$ における各試行の初回設計パラメータと収束結果を示す。

d) 収束しなかったケース

全 192 試行のうち 38 件（19.8%）が最大修正回数（5 回）以内に収束しなかった。内訳は RAG あり 14 件（14.6%）、RAG なし 24 件（25.0%）であった。

不収束ケースの不合格項目内訳を表 20 に示す。bend のみが 22 件（57.9%）と最多であり、次いで bend+web_slenderness が 13 件（34.2%）であった。web_slenderness のみは 3 件（7.9%）にとどまり、deck, shear, deflection による不収束は 0 件であった。

不収束ケースは主に $B \geq 20\text{ m}$ の広幅員条件に集中しており、 $B = 24\text{ m}$ では RAG ありでも 10/15 件、RAG なしでは 14/15 件が不収束であった。また、最終利用率が 1.0–1.1 の範囲にあるケースが 38 件中 24 件（63%）であり、最大修正回数を超過してもなお合格に近い状態で停止していた。

e) RAG による初回設計パラメータの変化

表 21 に、初回設計（修正ループ前）における主要断面パラメータの RAG 有無による比較を示す。32 条件それぞれで RAG あり/なし各 3 試行の平均を 1 データ点とし、対応ありの Wilcoxon 符号付き順位検定（両側）を行った。ウェブ高は RAG ありで平均 2,408 mm、RAG なしで 2,119 mm であり、13.6%の増加が高度に有意であった（ $p = 1.4 \times 10^{-6}$ ）。上フラン

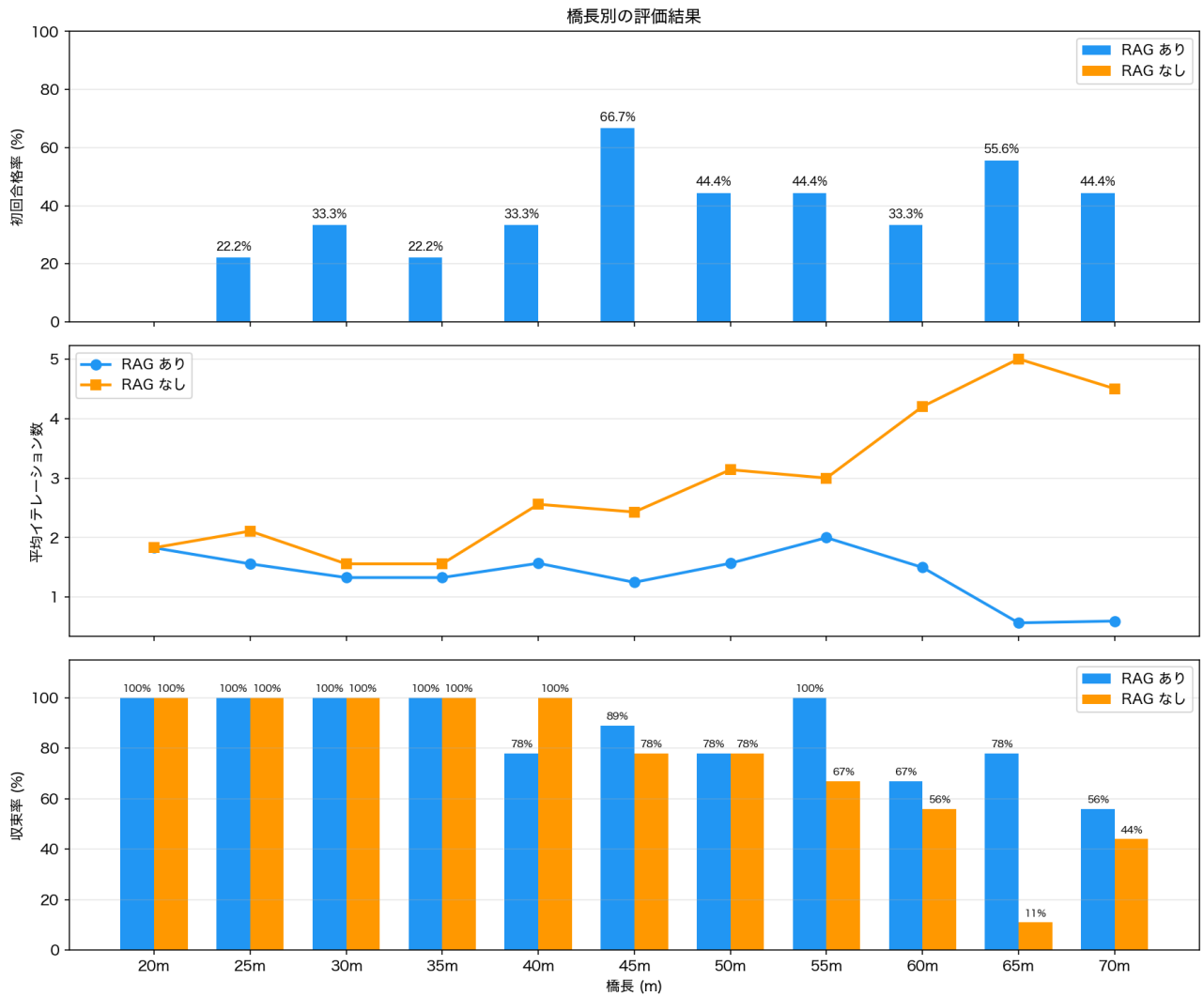


図-19: 橋長別の評価結果

表-21: RAG による初回設計パラメータの変化 (Wilcoxon 検定)

パラメータ	RAG あり	RAG なし	差	p 値
ウェブ高	2,408 mm	2,119 mm	+13.6%	1.4×10^{-6}
上フランジ幅	542 mm	443 mm	+22.5%	7.5×10^{-6}
下フランジ幅	637 mm	531 mm	+20.0%	1.7×10^{-5}
桁本数	4.4 本	4.4 本	—	—

ジ幅 (+22.5%, $p = 7.5 \times 10^{-6}$) および下フランジ幅 (+20.0%, $p = 1.7 \times 10^{-5}$) も有意に増加した。一方、桁本数には有意差がなかった。

5. 考察

(1) RAG の効果のメカニズム

RAG が初回設計品質を向上させるメカニズムについて考察する。RAG の検索ログを確認すると、桁高/支間比に関するガイダンス「 $h/L \approx 1/15 \sim 1/20$ が

適当」(道路橋示方書¹⁶⁾ 式 (7.364)) がランク 2 付近で安定して検索されていることが確認される。このルールが初回設計のウェブ高選択に直接影響を与えていることは、RAG reasoning の記述(「桁高目安 $h \approx L/15 \sim L/20$ を採用」)から裏付けられる。

表 21 で示したとおり、RAG ありではウェブ高が 13.6%有意に増加している。一方、フランジ幅も有意に増加しているが、RAG ヒットにフランジ幅の具体的な数値規定は含まれていない。下フランジ幅/ウェブ高の比率を確認すると、RAG ありで 0.273, RAG なしで 0.261 であり、有意差はなかった ($p = 0.15$)。すなわち、LLM は RAG の有無にかかわらず概ね一定の比率でフランジを設計しており、フランジ幅の増加はウェブ高増加に伴う間接的な結果と解釈される。

以上から、RAG の効果は以下の因果関係で説明できる。

1. RAG が $h/L \approx 1/15 \sim 1/20$ のルールを提供し、ウェブ高が 13.6%増加する (直接効果)。

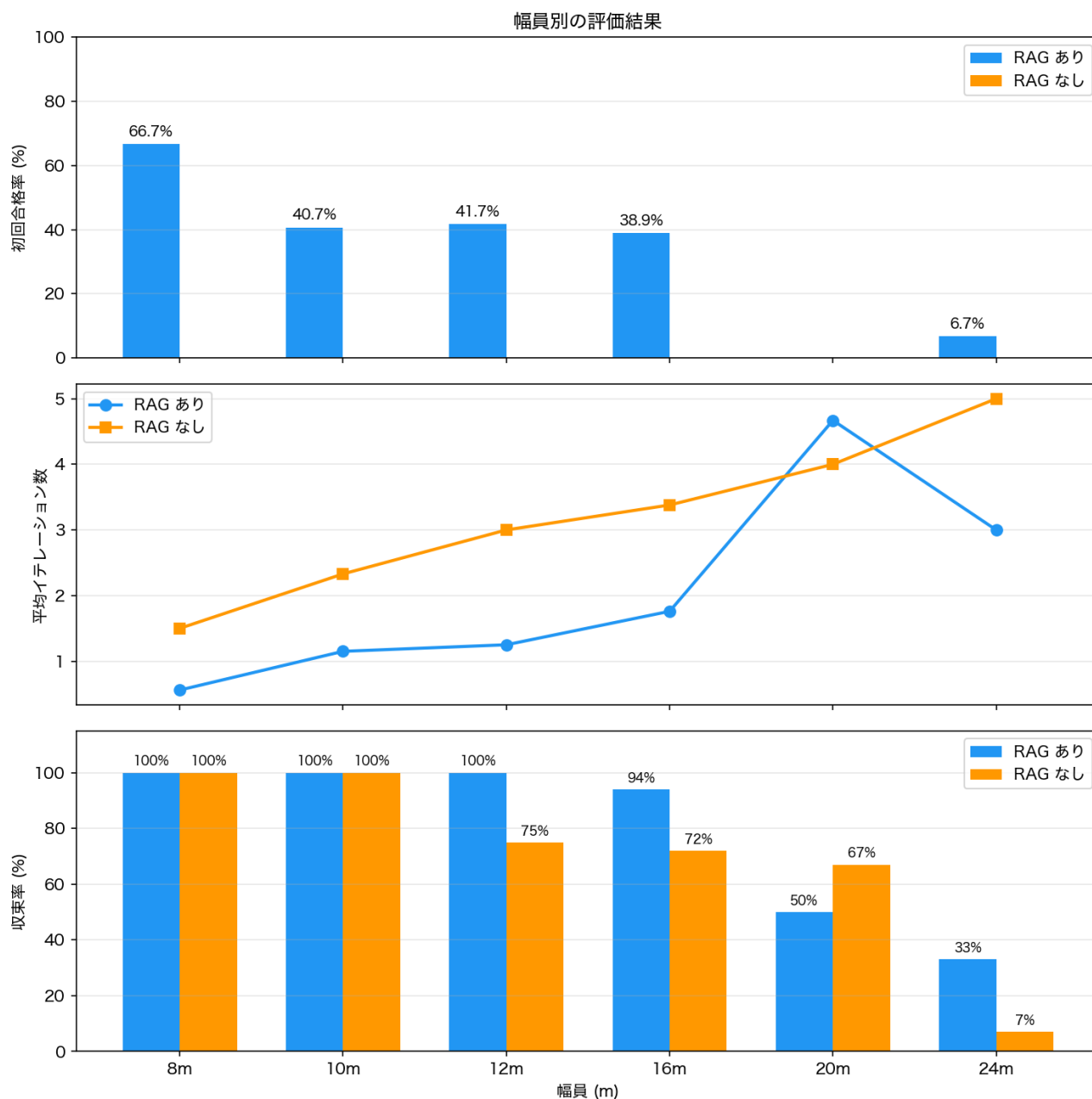


図-20: 幅員別の評価結果

- LLM がフランジ/ウェブ比を概ね一定 (~ 0.27) に保つため、フランジ幅も連動して増加する (間接効果)。
- ウェブ高・フランジ幅の両方が大きくなることで断面二次モーメント I が増加し、たわみが改善される。

この因果関係は、deflection の初回合格率改善 (+44.8 ポイント) が最も顕著であったことと整合する。deflection 初回合格群と不合格群のウェブ高中央値を比較すると、RAG あり合格群は 2,800 mm, RAG あり不合格群は 1,900 mm であり、900 mm の差が存在する。 I はウェブ高の 3 乗に比例するため、この差がたわ

み照査の可否に決定的に作用している。

deck (床版厚) の 100% 合格についても、RAG ログから床版厚関連の条文 (道路橋示方書¹⁶⁾ p.117: 最小全厚テーブル $d_0 = 30L + 110$) がランク 16–17 (スコア 0.70–0.71) で安定して検索されていることが確認される。RAG ありでは床版厚の公式に準拠した値が生成される傾向があり、96 件中 96 件で初回合格を達成している。RAG なしでは条文を参照できず、過小な床版厚を設定するケースが 32 件 (33.3%) 発生した。

(2) 広幅員条件における収束困難の構造的な原因

$B = 24\text{ m}$ における収束率の極端な低さ (RAG あり)

"reasoning": "全体方針: 与条件 $L=65\text{ m}$, $B=24\text{ m}$ の単純鋼プレートガーダー合成桁橋を対象とし, RAG から得られるプレートガーダー橋・RC 床版・床組・横桁の記述を根拠に, (1) 桁配置 (本数・間隔), (2) 主桁概略断面, (3) RC 床版厚, (4) 横桁パネル割を決める. 特に重視した根拠: $h/L \approx 1/15 \sim 1/20$ (式(7.364)) [13], 図 7.134 のスパン腹板高さ関係-[13], 断面設計フロー[10], 横桁間隔 $d \leq 20 \cdot$ 中央に m 1 本以上 [22], 横桁高 \approx 主桁高の 0.8 倍程度 [22], 最小全厚 $d_0 = 30L + 110$ と下限 160 mm . [17][19] ただし桁本数・間隔の推奨範囲や主桁の具体寸法表は RAG に与えられていないため, 根拠が不足する部分は「実務目安」として仮定扱いとする. 主桁本数の決定: 候補 $\in \text{num_girders} \{3, 4, 5, 6\}$ を比較し, 床版支間 (主桁間隔) に対する最小全厚 $d_0 = 30L + 110$ (連続版・主鉄筋直角) [17] と, 鋼重・製作性のバランスで決定する. overhang は 1.0 m 程度を確保 (明確根拠は RAG にないため仮定). $B=24.0\text{ m}$ より有効幅 22.0 m , $s = 22.0/(n-1)$. $n=3$: $s=11.0\text{ m} \rightarrow d_0 = 440\text{ mm}$ (過大) $n=4$: $s=7.33\text{ m} \rightarrow d_0 \approx 330\text{ mm}$ (厚い) $n=5$: $s=5.50\text{ m} \rightarrow d_0=275\text{ mm}$ $n=6$: $s=4.40\text{ m} \rightarrow d_0=242\text{ mm}$ 3 本は床版厚が非経済的. 4 本も死荷重増で不利. 6 本は床版厚は減るが桁・横桁数増で製作性が悪化. $s = 5 \sim 6\text{ m}$ 程度がバランス範囲とみなし, 5 本桁を採用. 採用: $\text{num_girders}=5$, $\text{girder_spacing}=5500\text{ mm}$, $\text{overhang}=1000\text{ mm}$ ($1.0 + 4 \times 5.5 + 1.0 = 24.0\text{ m}$) "

図-21: 桁本数決定に関する reasoning の抜粋 ($L = 65\text{ m}$, $B = 24\text{ m}$, 5 本桁選択例)

33%, RAG なし 7%) について, その構造的原因を考察する.

$B = 24\text{ m}$ の全 30 件で実際に選択された桁本数の分布は, 5 本桁が 19 件 (63%), 6 本桁が 10 件 (33%), 7 本桁が 1 件 (3%) であった. これらの収束率はそれぞれ 16% (3/19), 20% (2/10), 100% (1/1) であり, 5–6 本桁の収束率が極めて低い.

LLM が 5 本桁を選択する背景には, RAG ログの reasoning フィールドに記録された判断根拠がある. 図 21 に, 広幅員条件 ($L = 65\text{ m}$, $B = 24\text{ m}$) における桁本数決定に関する reasoning の抜粋を示す.

図 21 に示す reasoning では, $\text{overhang}=1.0\text{ m}$ を仮定し, $\text{num_girders} \in \{3, 4, 5, 6\}$ の各候補について主桁間隔 $s = 22.0/(n-1)$ を床版支間とみなして最小全厚式 $d_0 = 30L_{\text{support}} + 110$ [17] で床版厚を見積もり, 床版厚と桁本数増による鋼重・製作性のトレードオフから 5 本桁 ($s = 5.5\text{ m}$) を選択している. しかし, この判断は曲げ照査の定量的な見通しを持たないまま桁本数を固定しており, 結果として構造的に過小な配置が選択されてしまう.

さらに, Judge の修正ループでは断面寸法 (ウェブ高・板厚・フランジ寸法) の増減のみが可能であり, 桁本数の変更は Designer の初回選択に依存している. 5 本桁 \times 桁間隔 5500 mm では中間桁への荷重集中が大きく, 断面増加のみでは収束が困難である.

具体例として, $L60.B24.\text{rag_true_trial.1}$ のケースでは, 5 桁・桁間隔 5500 mm で設計が開始され, 中間桁の受け持ち幅 $b_i = 5.5\text{ m}$ の条件下で iter0 から iter5 まで 5 回の修正が実施された. 表 22 に推移を示す. 断面二次モーメント I は $207 \times 10^9\text{ mm}^4$ から $420 \times 10^9\text{ mm}^4$

へ 2 倍以上に増加したにもかかわらず, 曲げ利用率は 1.79 から 1.09 にとどまり合格 (≤ 1.0) に届かなかった. これは, (1) 桁間隔 5500 mm による高い荷重集中, (2) 断面増 \rightarrow 自重増の悪循環 (M_{total} が $38.5 \rightarrow 40.1\text{ GN}\cdot\text{mm}$ へ微増), (3) 腹板幅厚比制約 (桁高増 \rightarrow 必要最小腹板厚増), (4) フランジ厚 40 mm 超過時の降伏点低下 (SM490 : $f_y = 315 \rightarrow 295\text{ N/mm}^2$) の 4 つの制約が重なった結果である.

(3) bend と deflection の同時不合格

RAG ありの初回不合格 60 件のうち 63.3% (38 件) が bend+deflection の同時不合格であった (表 17). この同時不合格が頻発するメカニズムと, 修正ループにおける解消過程の非対称性について考察する.

ウェブ高の増加は断面二次モーメント I を増加させるが, I は主にたわみ ($\delta \propto 1/I$) の改善に直接的に寄与する. 一方, 曲げ応力 ($\sigma = M \cdot y/I$) の改善には I の増加とともに中立軸からの距離 y も増加するため, 効果が間接的である. この非対称性により, 修正ループにおいてたわみは早期に解消されるが, 曲げの収束にはより多くの修正回数を要する傾向がある.

不収束 38 件のうち, deflection による不収束は 0 件であり, 全て bend または web_slenderness が残存していた. すなわち, 修正ループはたわみを優先的に解消できる一方で, 曲げの収束に至らないケースが存在する.

(4) 修正ループの有効性と限界

Designer-Judge 修正ループにより, 全体の収束率は 80.2% に達しており, 初回合格率 18.8% から大幅に改善されている. このことは, LLM の初期設計が不十分であっても, 決定論的照査と LLM による修正提案の繰り返しにより設計品質を一定水準まで引き上げられることを示している.

一方で, 修正ループには構造的な限界も存在する. web_slenderness は初回設計では全ケースで合格 (100%) であったが, 修正ループ中にウェブ高を増加させた結果として二次的に不合格に転じるケースが観測された. ウェブ高を増加させると必要最小腹板厚 ($= h_w/130$, SM490 の場合) も増加し, 既存の腹板厚では幅厚比制約を満たせなくなる. この連鎖は, bend や deflection の改善を目的とした修正が新たな不合格を誘発するというトレードオフの存在を示している.

また, 不収束 38 件のうち 24 件 (63%) は最終利用率が 1.0–1.1 の範囲にあり, 合格に近い状態で修正回数上限に達している. 最大修正回数の引き上げ (5 回

表-22: 修正ループの推移例 (L60.B24_rag_true_trial_1)

iter	ウェブ高×厚	上F幅×厚	下F幅×厚	I ($\times 10^9 \text{ mm}^4$)	曲げ利用率	備考
0	3200×28	450×40	600×55	207	1.79	—
1	3500×28	450×44	600×55	263	1.63	TF 厚 >40mm→ f_y 低下
2	3700×28	450×48	600×55	308	1.46	web 幅厚比 NG
3	3700×30	550×50	600×55	339	1.29	web 厚増で対応
4	3700×30	650×50	650×55	367	1.19	—
5	3900×30	650×52	650×55	420	1.09	web 幅厚比が上限付近

→7-10 回) によってこれらのケースの収束が期待される。ただし、 $B = 24 \text{ m}$ で 5 本桁が選択されたケースのように、断面増加のみでは原理的に収束困難な場合も存在しており、桁本数を含む設計変数の修正が可能なループ構造への拡張が今後の課題である。

6. まとめと今後の課題

(1) まとめ

インフラ老朽化対策と BIM/CIM 推進が進む中、橋梁モデリングの工数削減が課題となっている。LLM の活用により自動化が期待される一方、LLM 単独では構造計算の正確性や設計根拠の明示が困難である。

本研究では、RAG (検索拡張生成) による設計知識の参照、決定論的照査による工学的妥当性の担保、LLM による柔軟な修正提案を組み合わせたハイブリッドシステムを提案した。本システムは、(1) RAG により道路橋示方書から設計根拠を明示、(2) 数式ベースの照査で LLM の計算誤りを排除、(3) IFC 出力までの一貫パイプラインを実現、という 3 つの特徴を持つ。

192 件の設計生成実験により、本実験条件において RAG ありの初回合格率が 37.5%、修正ループ適用後の最終合格率 (収束率) が 85.4% であることを確認した。橋長 20-70 m、幅員 8-24 m の範囲で、IFC ファイルまで出力し、構造的にある程度妥当な断面モデルの自動生成を実証した。

以上より、本システムは橋梁工学的にある程度妥当な BIM モデルを出力できるシステムとして一定の成果を示した。ただし、現状は構造形式の限定、照査の簡易化、収束率の課題が残っており、今後の改良が必須である。

(2) 今後の課題

a) BIM モデルとしての完成度向上

現在のシステムでは主桁、横桁、床版のみをモデル化している。実際の鋼プレートガーダー橋には対傾構・横構などの水平ブレースング部材が含まれる

ため、これらを追加し BIM モデルとしての完成度を高める必要がある。

b) 照査の厳密化

現状の照査は簡易的であり、以下の項目を追加することでより正確な構造照査が可能となる。

- **合成効果の考慮**：RC 床版との合成断面として計算 (現在は保守側で鋼 I 断面のみ)
- **座屈照査**：フランジ局部座屈、ウェブ全体座屈
- **疲労照査**：溶接部等の疲労強度評価
- **荷重の精緻化**：本研究では衝撃係数および群集荷重を考慮していないが、これらは Designer への入力情報として追加することで対応可能である。同様に温度応力等も拡張可能である

さらに、FEM による構造解析などと組み合わせることで、より現実的な挙動評価が可能となると考えられる。

c) 収束率向上への対応

広幅員条件 ($B \geq 20 \text{ m}$) での収束率改善のため、桁本数を含む設計変数の修正機構を検討する。

(3) 長期的展望

本研究では、道路橋示方書と教科書をテキスト化してそれを設計知として参照する形をとった。しかし、単純なチャンク化+検索の形では、前後の文脈を完全に拾えないことや、より複雑な関係を表現できないことが課題として残る。これを、道路橋示方書の条文間の参照関係 (「○○の場合は△△条を参照」等) をナレッジグラフとして表現し、本研究と同様に適切な形のパラメータセットたる JSON の形を作れば、様々な橋梁形式に対応可能なシステムへ拡張できると考える。これにより、鋼プレートガーダー橋以外の形式 (箱桁、トラス、アーチなど) に対しても、形式に応じた設計ルールを自動的に適用することが期待される。

さらに、MCP4IFC⁶⁾ に見られるような MCP (Model Context Protocol) を介したツール連携仕様が普及すれば、LLM は自然言語処理だけでなく外部ツール群を組み合わせる設計業務を遂行する「オーケストレー

タ」として機能し得る。本研究で実装した RAG 検索、決定論的照査、IFC 生成をそれぞれ MCP サーバとして分離すれば、他の橋梁形式や他システムからも再利用可能な設計支援基盤へ発展できる。さらに FEM 解析や疲労・座屈照査、数量計算、施工計画、維持管理データベース等を同一の呼び出し形式で接続できれば、概略設計から詳細設計、デジタルツイン運用までを連続したワークフローとして統合することが期待される。一方で、ツール呼び出し権限の管理、入出力スキーマの標準化、参照条文・計算ログの監査可能性確保など、運用面の課題も大きい。これらを踏まえ、MCP を介した検証可能な計算ツール群と知識表現（例：条文参照関係の構造化）を組み合わせることが、説明可能で再現可能な自律設計支援への長期的な方向性である。

謝辞

本研究を進めるにあたり、終始ご指導・ご助言を賜りました東京大学大学院工学系研究科の全邦釘先生に深く感謝申し上げます。何かとご心配・ご迷惑をおかけしましたが、本当にご多忙な中でも温かく見守ってくださり、無事本研究を終えることができました。深く感謝申し上げます。

また、研究室の先輩の皆様にも感謝申し上げます。特に阿部さん、田村さん、松尾さん、稲富さんは研究室で一緒に過ごす機会が多く、日常的にかけてくださる励ましの言葉や助言に支えられました。深く感謝いたします。同じく先輩である岡内さんには、本研究の範囲に限らず、本当に多くの技術的な助言をいただきました。一つしか年齢差がないとは思えないほど圧倒的な実力をお持ちであり、どんな質問をしても 120% の内容の答えが返ってくるのは衝撃的でしたが、同時にそれが自分にもできるという良い刺激になりました。心より感謝申し上げます。

また、日々のゼミや研究室で温かいお声掛けやアドバイスをくださいました堀田先生、鈴木先生、Yu 先生、小澤先生、沼田先生、秘書の皆さま、研究員の皆さまにも深く感謝申し上げます。皆さまのお心遣いと励ましのおかげで、研究に専念できる環境が整えられ、本研究の推進に大いに助力となりました。

プライベートで関わる友人にも感謝します。特に、同期の親友とも呼べる 8 人の存在には大変助けられました。将来の話、研究の話、その他どんな話でも共有できる彼らの存在は、なかなか研究が進まないような時期に特に、自分の心の大きな支えでした。改めて感謝の意を表します。

最後に、ここまでの人生を支えてくれた家族に心から感謝します。教育、精神面、金銭面、本当に何から何まで、家族の無償の愛と支えがあったからこそ、今日の自分があります。深く感謝しています。

本研究の成果は、上記の皆さまのご尽力なくしては成し遂げられなかったものであり、改めて心より御礼申し上げます。今後も皆さまからいただいた知見を生かし、土木業界の DX・デジタルツイン推進に少しでも貢献できるよう、精進してまいります。

REFERENCES

- 1) 国土交通省：社会資本の老朽化の現状と将来, https://www.mlit.go.jp/sogoseisaku/maintenance/02research/02_01.html (参照 2026-01-26) .
- 2) 国土交通省：i-Construction, <https://www.mlit.go.jp/tec/i-construction/index.html> (参照 2026-01-28) .
- 3) 国土交通省：BIM/CIM 関連, https://www.mlit.go.jp/tec/tec_tk_000037.html (参照 2026-01-26) .
- 4) buildingSMART International: Industry Foundation Classes (IFC), <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/> (accessed 2026-01-26).
- 5) 国土交通省：BIM/CIM の進め方について, <https://www.mlit.go.jp/tec/content/001757200.pdf> (参照 2026-01-26) .
- 6) B. K. Nithyanantham, T. Sesterhenn, A. Nedungadi, S. Peral Garijo, J. Zenkner, C. Bartelt, and S. Lüdtke: MCP4IFC: IFC-Based Building Design Using Large Language Models, arXiv:2511.05533v1 [cs.CL], 2025, doi:10.48550/arXiv.2511.05533.
- 7) Yang, Y., Jing, X. and Liu, Y.-M.: Automated Checking of Highway Bridge BIM Models Based on Large Language Models, Buildings, Vol. 15, No. 19, 3465, 2025. doi:10.3390/buildings15193465.
- 8) 山本 敦大, 緒方 陸, 藤井 純一郎, 山本 一浩: 生成 AI による単純形状 3 次元データモデルの対話的な生成手法の一考察, AI・データサイエンス論文集, Vol.6, No.1, pp.96–106, 2025. doi:10.11532/jsceiii.6.1.96.
- 9) 箱石 健太, 一言 正之, 菅田 大輔: 土木分野における事前学習モデル BERT による精度検証, 土木学会論文集特集号 (土木情報学), Vol.79, No.22, 22-22042, 2023. doi:10.2208/jscej.22-22042.
- 10) Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, arXiv:2005.11401, 2020. doi:10.48550/arXiv.2005.11401.
- 11) Soudani, H., Kanoulas, E., and Hasibi, F.: Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge, arXiv:2403.01432 [cs.CL], 2024. doi:10.48550/arXiv.2403.01432.
- 12) Li, X., Wang, W., Li, M., Guo, J., Zhang, Y., and Feng, F.: Evaluating Mathematical Reasoning of Large Language Models: A Focus on Error Identification and Correction, Findings of the Association for Computational Linguistics: ACL 2024, pp. 11316–11360, 2024. doi:10.18653/v1/2024.findings-acl.673.

- 13) OpenAI: Structured model outputs — OpenAI API, <https://platform.openai.com/docs/guides/structured-outputs> (accessed 2026-01-28).
- 14) OpenAI: GPT-5.1 Model — OpenAI API, <https://platform.openai.com/docs/models/gpt-5.1> (accessed 2026-01-29).
- 15) Pydantic: Pydantic Validation, <https://docs.pydantic.dev/> (accessed 2026-01-28).
- 16) 国土交通省：道路橋示方書, <https://www.mlit.go.jp/road/sign/kijyun/pdf/20170721hashikouka.pdf> (参照 2026-01-28) .
- 17) 中井博, 北田俊行: 鋼橋設計の基礎. 共立出版株式会社, 1992.
- 18) Singer-Vine, J. and The pdfplumber contributors: pdfplumber (Version 0.11.9), GitHub repository, 2026. <https://github.com/jsvine/pdfplumber> (accessed 2026-01-28).
- 19) OpenAI: Vector embeddings — OpenAI API, <https://platform.openai.com/docs/guides/embeddings> (accessed 2026-01-28).
- 20) NumPy: NumPy Documentation, <https://numpy.org/doc/stable/> (accessed 2026-01-28).
- 21) OpenAI: Responses — OpenAI API Reference, <https://platform.openai.com/docs/api-reference/responses> (accessed 2026-01-28).
- 22) IfcOpenShell: IfcOpenShell - The open source IFC toolkit and geometry engine, <https://ifcopenshell.org/> (accessed 2026-01-28).

A Study on Automatic Generation Method of BIM Models for Steel Plate Girder Bridges Using LLM

Hiroki ABE

As infrastructure aging countermeasures and BIM/CIM adoption progress, the labor required for creating bridge 3D models has become a barrier to widespread adoption. This study proposes a system that automatically generates BIM models of steel plate girder bridges (with RC deck slabs) using large language models (LLMs). The key features of this system are: (1) autonomously determining initial design parameters by retrieving design rules from sources such as the Japanese Specifications for Highway Bridges via RAG (Retrieval-Augmented Generation) and extracting them as structured DesignRules, thereby documenting the design rationale without hardcoding heuristic rules; (2) eliminating LLM calculation errors by executing structural checks (bending, shear, deflection, etc.) through deterministic processing; (3) incorporating an automatic repair loop where the LLM generates correction proposals when checks fail, iterating until passing; and (4) producing IFC-format 3D models from only two input parameters: bridge length and width. Design generation experiments were conducted for 192 cases across 32 conditions (bridge length 20–70m, width 8–24m) with two levels of RAG presence. With RAG enabled, the first-pass acceptance rate was 37.5% and the convergence rate after the repair loop was 85.4%, compared to 0% and 75.0% respectively without RAG, indicating the overall effectiveness of RAG under the experimental conditions. RAG provided reference to girder height-to-span ratio rules, increasing web height by an average of 13.6%, which improved the first-pass acceptance rate for deflection checks by 44.8 percentage points. However, convergence rates decreased for wide-width conditions (24m) due to the inability to modify girder count during the repair loop. This system demonstrated the capability to automatically generate BIM models with reasonable structural validity. Future work includes adding lateral bracing members, implementing rigorous buckling and fatigue checks, and introducing mechanisms to modify design variables including girder count.