

Un Guide Complet pour la Conception de Produits Augmentée par l'IA Générative : de l'Idéation au Prototype Validé

Partie I : Les Piliers Fondamentaux de la Conception Pilotée par l'IA

Cette première partie établit les principes scientifiques et d'ingénierie qui sous-tendent l'ensemble du flux de travail. Elle est conçue pour fournir au lecteur un modèle mental robuste de chaque technologie clé avant qu'elles ne soient combinées dans la Partie II.

Chapitre 1 : Le Nouveau Paradigme de Conception : Conception Générative et Optimisation Topologique

1.1. Introduction : Au-delà de l'Intuition Humaine

La conception de produits a traditionnellement reposé sur un processus itératif guidé par l'expérience, l'intuition et les connaissances accumulées de l'ingénieur ou du designer. Ce processus, bien qu'efficace, est intrinsèquement limité par les biais cognitifs humains et la difficulté à imaginer des solutions radicalement nouvelles. Les méthodes de conception générative représentent un changement de paradigme fondamental. Plutôt que de demander à un humain de créer une solution, cette approche lui demande de définir méticuleusement le problème. L'ingénieur spécifie les objectifs (par exemple, minimiser la masse, maximiser la

rigidité), les contraintes (matériaux disponibles, procédés de fabrication, espaces à préserver) et les conditions de charge. Ensuite, des algorithmes d'intelligence artificielle (IA) explorent de manière autonome un vaste espace de solutions possibles, générant des centaines, voire des milliers d'options de conception.¹

La proposition de valeur fondamentale de cette approche est sa capacité à découvrir des conceptions novatrices et hautement performantes qu'un concepteur humain n'aurait probablement jamais envisagées.² Ces conceptions présentent souvent des formes organiques, inspirées par la nature, qui suivent les chemins de force de manière optimale, aboutissant à des structures à la fois légères et robustes.⁴ Ce processus ne remplace pas l'ingénieur, mais augmente ses capacités, le transformant d'un créateur de géométrie en un architecte de problèmes et un curateur de solutions.

1.2. Définir la Conception Générative : Un Processus Piloté par les Objectifs

La conception générative (en anglais, *Generative Design*) est une méthodologie d'exploration de la conception où l'ingénieur s'associe à une intelligence artificielle pour découvrir des solutions optimales.¹ Contrairement au processus de conception classique qui commence par la création d'un modèle 3D, la conception générative commence par la définition des paramètres du problème.⁶

Le processus peut être schématisé comme une boucle itérative ⁷ :

1. **Définition du Problème** : L'ingénieur définit les exigences fonctionnelles, les contraintes géométriques (zones à préserver, zones à éviter), les cas de charge (forces, pressions, moments), les matériaux candidats et les procédés de fabrication possibles.⁷
2. **Génération de Solutions** : Le logiciel utilise des algorithmes, souvent basés sur l'IA et l'optimisation, pour générer une multitude de solutions de conception qui satisfont toutes les contraintes spécifiées.⁸
3. **Exploration et Sélection** : Le système présente les résultats à l'ingénieur, souvent sous forme de graphiques interactifs qui permettent de visualiser les compromis entre différents objectifs (par exemple, masse vs. coût vs. facteur de sécurité). L'ingénieur peut alors filtrer, comparer et sélectionner les conceptions les plus prometteuses.⁴
4. **Raffinement** : La conception choisie est souvent exportée sous forme de modèle CAO éditable, que l'ingénieur peut ensuite affiner, modifier et intégrer dans un assemblage plus large.⁹

Il est essentiel de comprendre que la conception générative n'est pas un algorithme unique, mais une méthodologie qui englobe plusieurs technologies informatiques pour explorer

l'espace des possibles.¹⁰

1.3. L'Optimisation Topologique : Le Moteur de l'Efficacité Matérielle

Au cœur de nombreux systèmes de conception générative se trouve un algorithme puissant appelé **Optimisation Topologique** (OT). L'OT est une méthode mathématique qui a pour but de trouver la répartition optimale de matière dans un volume de conception donné, pour un ensemble de charges et de conditions aux limites, afin de maximiser la performance de la pièce.¹¹ L'objectif le plus courant est de minimiser la compliance (l'inverse de la rigidité) pour une contrainte de volume ou de masse donnée, ce qui revient à créer la pièce la plus rigide possible pour une quantité de matière définie.¹²

Le principe fondamental de l'OT est d'éliminer la matière des zones où elle n'est pas nécessaire pour supporter les charges.¹¹ Le processus peut être conceptualisé de la manière suivante :

1. **Définition de l'Espace de Conception** : On commence par un volume maximal que la pièce peut occuper.
2. **Discretisation** : Cet espace est divisé en un très grand nombre de petits éléments (voxels ou éléments finis), formant un maillage.¹⁴
3. **Analyse Itérative** : Un algorithme, tel que la méthode SIMP (*Solid Isotropic Material with Penalization*), attribue à chaque élément une variable de densité, allant de 0 (vide) à 1 (plein).¹⁵
4. **Évaluation** : À chaque itération, une Analyse par Éléments Finis (AEF, ou FEA en anglais) est effectuée pour calculer les contraintes et les déformations dans la structure actuelle.¹⁴
5. **Suppression de Matière** : L'algorithme réduit la densité des éléments qui subissent de faibles contraintes, les rendant "plus vides". La méthode de pénalisation (le "P" de SIMP) décourage les densités intermédiaires, poussant la solution vers une structure claire de "plein" ou "vide".¹⁵
6. **Convergence** : Le processus est répété jusqu'à ce qu'une solution stable soit atteinte, où la matière restante forme un chemin de charge optimal, ressemblant souvent à des structures osseuses ou arborescentes.⁴

Le résultat brut de l'OT est un modèle maillé (semblable à un fichier STL) qui doit ensuite être interprété et reconstruit en un modèle CAO lisse et utilisable (représentation B-rep).¹¹

1.4. Distinction entre Conception Générative et Optimisation Topologique

Une confusion fréquente existe entre les termes "Conception Générative" et "Optimisation Topologique". Bien qu'ils soient étroitement liés, ils ne sont pas interchangeables. La clarification de cette distinction est fondamentale pour comprendre la portée du projet.¹⁸

- **L'Optimisation Topologique (OT)** est un *algorithme* qui, à partir d'une géométrie de départ et d'un ensemble unique de conditions (charges, contraintes, matériau), produit *une seule* proposition de conception optimisée.²⁰ C'est un outil d'optimisation puissant mais focalisé.
- **La Conception Générative (CG)** est un *processus* ou une *méthodologie* qui utilise des algorithmes (souvent, mais pas exclusivement, l'OT) pour explorer un large éventail de possibilités. Une étude de CG exécute de multiples optimisations en faisant varier les paramètres d'entrée, tels que les matériaux, les méthodes de fabrication ou même les objectifs.⁴ Le résultat d'une étude de CG n'est pas une seule conception, mais une *multitude* de conceptions alternatives, chacune étant une solution optimale pour une combinaison spécifique de paramètres.⁴

En somme, l'optimisation topologique peut être considérée comme l'un des principaux moteurs algorithmiques au sein du cadre plus large de la conception générative.¹⁶ La CG est la stratégie d'exploration, tandis que l'OT est une tactique d'optimisation. Par exemple, une étude de conception générative pour un support pourrait générer :

- Une solution optimisée en aluminium pour l'usinage CNC 3 axes.
- Une solution optimisée en titane pour la fabrication additive métallique (SLM).
- Une solution optimisée en plastique ABS pour le moulage par injection.

Chacune de ces solutions pourrait être le fruit d'une optimisation topologique distincte, mais c'est le processus global qui constitue la conception générative.²

Chapitre 2 : Le Langage de la Physique : l'Analyse par Éléments Finis (AEF) comme Moteur de Validation

2.1. Qu'est-ce que l'AEF? Du Problème Physique au Modèle Mathématique

L'Analyse par Éléments Finis (AEF, ou *Finite Element Analysis*, FEA, en anglais) est une méthode de simulation numérique utilisée pour prédire comment un objet physique se

comportera sous l'effet de diverses conditions réelles : forces, chaleur, vibrations, écoulement de fluide, etc..²³ La technique mathématique sous-jacente est la Méthode des Éléments Finis (MEF, ou *Finite Element Method*, FEM).²³

Le concept central de la MEF est la **discrétisation**. Il est mathématiquement très complexe de résoudre les équations différentielles partielles (EDP) qui décrivent le comportement physique (par exemple, les équations de l'élasticité) pour un objet de forme complexe dans son ensemble.²⁵ La MEF contourne cette difficulté en décomposant l'objet complexe en un grand nombre de petites pièces simples, appelées **éléments finis**.²⁷ Ces éléments ont des formes géométriques simples (comme des tétraèdres ou des hexaèdres en 3D) et sont connectés entre eux par des points appelés **nœuds**. L'ensemble de ces éléments et nœuds forme ce que l'on appelle un **maillage** (*mesh*).²³

Pour chaque petit élément, les EDP complexes sont approximées par un système d'équations algébriques beaucoup plus simples à résoudre.²⁵ L'ordinateur assemble ensuite les équations de tous les éléments pour former un immense système d'équations global qui représente le comportement de l'objet entier. En résolvant ce système, on obtient une solution approchée du comportement de la structure, comme les déplacements de chaque nœud, à partir desquels on peut ensuite calculer les déformations et les contraintes.²⁸

2.2. Le Flux de Travail de l'AEF en Pratique

Une analyse par éléments finis typique se déroule en trois étapes principales²⁷ :

1. **Pré-traitement (*Preprocessing*)** : C'est l'étape de définition du modèle de simulation. L'ingénieur :
 - Définit ou importe la **géométrie** de la pièce à analyser.
 - Attribue des **propriétés de matériau** à la géométrie (par exemple, module de Young, coefficient de Poisson, densité pour une analyse structurelle).
 - Crée le **maillage**, c'est-à-dire la division de la géométrie en éléments finis. La finesse du maillage (la taille des éléments) est un compromis entre la précision des résultats et le temps de calcul.
 - Applique les **conditions aux limites** (*boundary conditions*) :
 - Les **charges** (*loads*) : forces, pressions, moments, accélérations ou charges thermiques qui s'appliquent à la pièce.²⁷
 - Les **contraintes** ou **appuis** (*constraints*) : zones de la pièce qui sont fixes ou dont le déplacement est restreint.²⁸
2. **Résolution (*Solving*)** : L'ordinateur, ou le "solveur", prend en charge le modèle défini et procède à la résolution numérique du grand système d'équations matricielles. Cette étape est intensive en calcul et peut prendre de quelques secondes à plusieurs heures

ou jours, selon la complexité du modèle.²³

3. **Post-traitement (*Post-processing*)** : Une fois la résolution terminée, les résultats bruts (par exemple, les déplacements nodaux) sont traités pour être visualisés et interprétés par l'ingénieur. Les résultats sont généralement affichés sous forme de cartes de couleurs (contours) superposées sur la géométrie de la pièce, montrant la distribution des :
 - **Contraintes** (par exemple, la contrainte de von Mises, qui indique les zones où le matériau est le plus susceptible de céder).
 - **Déformations** (*strain*).
 - **Déplacements** (*displacement*).²⁹

L'ingénieur analyse ces résultats pour valider la conception : la pièce est-elle assez solide? Ne se déforme-t-elle pas trop? Y a-t-il des concentrations de contraintes inattendues?.³⁰

2.3. Le Rôle de l'AEF dans la Conception Générative et l'Optimisation Topologique

L'AEF n'est pas seulement un outil de validation finale dans le contexte de la conception générative ; elle est le moteur computationnel qui rend l'optimisation topologique possible. Le processus d'OT est une boucle d'optimisation où, à chaque itération, l'algorithme doit évaluer la performance de la géométrie actuelle pour décider où retirer de la matière.¹³

Cette évaluation de la performance est réalisée par une analyse par éléments finis. Le flux de travail interne de l'optimisation topologique est donc le suivant :

1. **Itération \$n\$** : L'algorithme propose une distribution de densité de matière.
2. **Analyse AEF** : Un solveur AEF est exécuté sur cette géométrie pour calculer son champ de contraintes et de déformations sous les charges définies.
3. **Calcul de la Sensibilité** : À partir des résultats de l'AEF, l'algorithme calcule la "sensibilité" de chaque élément, c'est-à-dire l'impact de la suppression de cet élément sur la performance globale (par exemple, la rigidité).
4. **Mise à Jour de la Conception** : L'algorithme supprime (réduit la densité) les éléments les moins "utiles" (ceux dont la suppression a le moins d'impact négatif sur la rigidité) tout en respectant la contrainte de volume.
5. **Convergence?** : Si la conception a convergé vers une solution stable, le processus s'arrête. Sinon, il retourne à l'étape 1 pour l'**itération \$n+1\$**.

Ce lien intime révèle une dépendance critique : la qualité et l'efficacité du processus de conception générative sont directement liées à la précision et à la rapidité des milliers d'analyses AEF exécutées en arrière-plan. C'est pourquoi les études de conception générative sont souvent gourmandes en ressources de calcul et bénéficient grandement du calcul en

nuage (*cloud computing*) qui permet de paralléliser ces analyses.⁸ L'AEF n'est donc pas seulement un juge final, mais un guide constant tout au long du chemin de l'optimisation.

Chapitre 3 : Faire le Pont entre le Numérique et le Physique : Conception pour la Fabrication et l'Assemblage (DFM/DFA)

3.1. Principes Fondamentaux du DFM et du DFA

La conception pour la fabrication et l'assemblage (DFMA, *Design for Manufacturing and Assembly*) est une méthodologie d'ingénierie qui vise à optimiser la conception d'un produit pour en faciliter la production et l'assemblage, réduisant ainsi les coûts et le temps de mise sur le marché.³² Elle se compose de deux piliers complémentaires :

- **Conception pour la Fabrication (DFM, *Design for Manufacturing*)** : Se concentre sur l'optimisation de la conception de chaque pièce individuelle pour la rendre plus facile, plus rapide et moins coûteuse à produire avec un procédé de fabrication spécifique (usinage, moulage, impression 3D, etc.).³² Les principes clés incluent la simplification des géométries, le choix de matériaux appropriés, la définition de tolérances réalisables et la collaboration avec les fournisseurs.³⁶
- **Conception pour l'Assemblage (DFA, *Design for Assembly*)** : Se concentre sur la simplification du processus d'assemblage du produit final. L'objectif est de réduire le nombre de pièces, de standardiser les composants, de minimiser les manipulations et de concevoir des jonctions simples (par exemple, des clips au lieu de vis).³²

L'approche DFMA combine ces deux aspects de manière synergique. Par exemple, réduire le nombre de pièces (un objectif DFA) peut conduire à une seule pièce plus complexe, ce qui pourrait augmenter son coût de fabrication (un problème DFM). Le DFMA cherche le meilleur compromis pour minimiser le coût *total* du produit.³³

3.2. L'Importance Critique du DFM/DFA pour la Conception Générative

Les formes organiques et complexes, souvent en treillis, qui résultent de l'optimisation topologique sont l'une des caractéristiques les plus reconnaissables de la conception

génération.² Cependant, ces formes sont souvent très difficiles, voire impossibles, à fabriquer avec des méthodes traditionnelles comme l'usinage ou le moulage.⁴ Une conception "optimale" sur le plan structurel mais infabriquable n'a aucune valeur pratique.

C'est là que l'intégration des principes DFM/DFA devient essentielle. Les logiciels de conception générative modernes ne se contentent pas d'optimiser la performance structurelle ; ils permettent d'intégrer les contraintes de fabrication directement dans le processus d'optimisation.⁹ En spécifiant le procédé de fabrication envisagé (par exemple, usinage 3 axes, moulage par injection, fabrication additive) comme une contrainte de l'étude, l'algorithme est forcé d'explorer uniquement les solutions qui sont non seulement performantes mais aussi réalisables et rentables.³⁹ Sans ces contraintes, la conception générative reste un exercice académique ; avec elles, elle devient un outil de production puissant.

3.3. Traduire les Principes DFM en Contraintes Mathématiques

Pour qu'un algorithme d'optimisation puisse tenir compte des contraintes de fabrication, les règles qualitatives du DFM doivent être traduites en un langage qu'il peut comprendre : des contraintes mathématiques et géométriques.⁴⁰ Ce processus de formalisation est au cœur de la conception générative orientée fabrication. Voici quelques exemples qui seront détaillés dans la Partie II :

- **Pour le moulage par injection** : La règle "la pièce doit pouvoir être démoulée" est traduite par une contrainte de **direction de démoulage** (*pull direction*). Mathématiquement, cela signifie que pour un vecteur de démoulage donné, aucune surface de la pièce ne peut créer de contre-dépouille (*undercut*). L'algorithme ne générera que des formes qui respectent cette condition.⁴²
- **Pour la fabrication additive (impression 3D)** : La règle "éviter les surplombs excessifs pour minimiser les supports" est traduite par une contrainte d'**angle de surplomb** (*overhang angle*). L'algorithme pénalisera ou interdira la création de surfaces dont l'angle par rapport à la direction de construction est inférieur à un seuil critique (par exemple, 45 degrés).⁴⁴
- **Pour l'usinage CNC** : La règle "la pièce doit être accessible par un outil de coupe depuis une ou plusieurs directions" est traduite par une contrainte d'**extrusion linéaire** ou de **direction de fraisage**. L'algorithme ne créera que des géométries qui ne présentent pas de cavités inaccessibles pour l'outil.⁵

En intégrant ces contraintes mathématiques dès le départ, le processus de conception générative produit des solutions qui sont "correctes par construction", réduisant considérablement le besoin de retouches manuelles post-optimisation et garantissant que les

gains de performance théoriques peuvent être réalisés dans le monde physique.⁴⁶

Partie II : Le Flux de Travail Augmenté : Un Guide d'Implémentation Étape par Étape

Cette partie constitue le cœur pratique du rapport, guidant l'utilisateur à travers chaque phase du projet proposé. Elle est riche en conseils pratiques et en recommandations de frameworks.

Chapitre 4 : Phase 1 - Idéation et Ingénierie des Exigences Assistées par l'IA

4.1. Le Rôle des Grands Modèles de Langage (LLM) dans la Phase d'Avant-Projet

La phase initiale de tout projet de conception, souvent qualifiée de "floue" (*fuzzy front end*), est caractérisée par l'incertitude, l'exploration d'idées et la définition progressive des exigences. C'est une phase traditionnellement intensive en travail humain, impliquant des brainstormings, des analyses de marché et la rédaction de documents de spécifications. Les grands modèles de langage (LLM), tels que les modèles de la série GPT d'OpenAI, Llama de Meta, ou Gemini de Google, offrent des capacités nouvelles pour structurer et accélérer cette phase.⁴⁷

Un LLM est un type d'intelligence artificielle générative entraîné sur d'immenses corpus de texte et de code, lui permettant de comprendre et de générer du langage humain avec une grande fluidité.⁴⁷ Dans le contexte de l'ingénierie de conception, les LLM peuvent être utilisés comme des assistants pour⁵⁰ :

- **Générer des idées de conception** en réponse à une description de problème.
- **Analyser les retours d'utilisateurs** ou des documents techniques pour en extraire les besoins et les contraintes clés.
- **Rédiger des ébauches de documents d'exigences produit** (PRD - *Product Requirements Document*) ou de cahiers des charges fonctionnels.

- **Identifier les informations manquantes** ou les ambiguïtés dans un brief de conception initial.⁵⁰
- **Traduire des exigences en langage naturel** en spécifications plus structurées, comme des récits utilisateurs (*user stories*).⁵⁰

L'objectif n'est pas de remplacer le jugement de l'ingénieur, mais de lui fournir une base de travail solide et de systématiser la capture des exigences, une tâche souvent sujette aux erreurs et aux oublis.⁴⁸

4.2. Le Prompt Structuré pour les Spécifications Techniques

Pour interagir efficacement avec un LLM, en particulier pour des tâches techniques, la simple formulation d'une question est insuffisante. La qualité de la sortie est directement proportionnelle à la qualité de l'entrée. Cette discipline de conception des entrées est appelée **l'ingénierie de prompt** (*prompt engineering*).⁵³ Plusieurs techniques sont essentielles pour obtenir des résultats précis et fiables.

- **Apprentissage "Zero-shot" vs. "Few-shot" :**
 - Un prompt *zero-shot* demande au modèle d'effectuer une tâche sans aucun exemple.⁵⁵ Par exemple : "Rédige les exigences pour un support de moteur". Le résultat sera basé uniquement sur les connaissances générales du modèle.
 - Un prompt *few-shot* inclut un ou plusieurs exemples du format de sortie désiré.⁵⁷ Cette technique conditionne le modèle à suivre un format et un style spécifiques, améliorant considérablement la pertinence et la structure de la réponse. Pour une tâche technique, fournir des exemples est presque toujours préférable.⁵⁶
- **Prompting en Chaîne de Pensée (*Chain-of-Thought, CoT*) :**
 - Cette technique consiste à instruire explicitement le LLM de "réfléchir étape par étape" avant de donner sa réponse finale.⁵⁵ En forçant le modèle à décomposer un problème complexe en une séquence de raisonnements logiques, on améliore sa capacité à résoudre des problèmes complexes et on réduit le risque d'erreurs factuelles ou logiques.⁵⁴ Par exemple, au lieu de demander "Quelles sont les charges sur le support?", on demandera "Décompose les étapes pour déterminer les charges sur le support. D'abord, identifie les sources de vibration. Ensuite, calcule les forces d'accélération. Enfin, résume les cas de charge critiques."
- **Modèles Structurés (*Structured Templates*) :**
 - Pour garantir que le LLM fournisse toutes les informations nécessaires dans un format utilisable, il est très efficace d'utiliser des modèles structurés. Le prompt peut contenir un squelette de document (en Markdown, YAML, ou même avec des balises XML personnalisées) que le modèle doit remplir.⁵² Cela transforme la tâche de génération de texte libre en une tâche de remplissage de formulaire, ce qui est

beaucoup plus contraint et fiable.⁶³

Exemple de Prompt Structuré (Few-shot avec CoT et Template) :

Vous êtes un ingénieur mécanique expert en conception de composants aérospatiaux. Votre tâche est de générer un brief de conception détaillé pour un nouveau support (bracket).

Réfléchissez étape par étape pour remplir chaque section du modèle ci-dessous.

EXEMPLE 1

Demande utilisateur : "Support pour une pompe hydraulique dans le train d'atterrissage."

Brief de conception généré :

- **Titre du Projet** : Support de pompe hydraulique - Train d'atterrissage principal
- **Objectif Principal** : Réduction de masse de 30% par rapport à la pièce actuelle (PN: 78B-45-2) tout en maintenant un facteur de sécurité de 1.5.
- **Matériaux Candidats** : Aluminium 7075-T6, Titane Ti-6Al-4V.
- **Cas de Charge** :
 1. **Statique** : Poids de la pompe (15 kg) avec un facteur de charge de 3g.
 2. **Vibratoire** : Analyse modale requise pour éviter les fréquences de résonance entre 50-200 Hz.
 3. **Atterrissage** : Charge de choc de 15g appliquée sur l'axe Z pendant 100 ms.
- **Contraintes de Fabrication** : Usinage CNC 5 axes OU Fabrication Additive (SLM).

FIN DE L'EXEMPLE 1

Maintenant, générez un brief de conception pour la demande utilisateur suivante en utilisant exactement le même format.

Demande utilisateur : "Un support léger pour un boîtier électronique dans la soute d'un satellite."

Remplissez le modèle ci-dessous :

- **Titre du Projet** :
- **Objectif Principal** :
- **Matériaux Candidats** :
- **Cas de Charge** :
- **Contraintes de Fabrication** :

4.3. Assurer la Reproductibilité : Traçabilité et Contrôle de Version des Prompts

Un des aspects les plus critiques et souvent négligés de l'utilisation des LLM dans un processus d'ingénierie est la gestion des prompts eux-mêmes. Un prompt n'est pas une simple question jetable ; c'est un artefact de conception, une forme de code source qui définit le comportement d'un système d'IA. Pour garantir la reproductibilité, l'auditabilité et la collaboration, les prompts doivent être gérés avec la même rigueur que le code logiciel.⁶⁴

Le flux de travail le plus robuste pour la gestion des prompts est basé sur des systèmes de contrôle de version comme **Git**.⁶⁴

1. **Centralisation** : Tous les prompts du projet sont stockés dans un dépôt Git dédié. Chaque prompt est un fichier texte (par exemple, `design_brief_generator.md`).
2. **Historique des Modifications** : Chaque modification apportée à un prompt est enregistrée via un commit. Le message de commit doit expliquer la nature du changement et son impact attendu ("Ex: Ajout d'un exemple pour améliorer la gestion des cas de charge vibratoires").⁶⁶
3. **Expérimentation avec les Branches** : Pour tester une nouvelle version d'un prompt sans affecter la version de production, un développeur crée une nouvelle branche (feature/improve-material-suggestions). Les expériences sont menées sur cette branche. Si les résultats sont concluants, la branche est fusionnée (*merged*) dans la branche principale.⁶⁷
4. **Vernionnement Sémantique** : Les versions stables des prompts sont identifiées par des tags (par exemple, v1.0, v1.1). Cela permet de lier une version spécifique d'un brief de conception à la version exacte du prompt qui l'a généré.⁶⁸

Ce processus garantit une **traçabilité** complète, demandée explicitement dans le cahier des charges du projet. Si une exigence de conception s'avère incorrecte, il est possible de remonter à la version exacte du prompt utilisé et de comprendre pourquoi le LLM a généré cette sortie.

Frameworks et Outils :

- **Gestion de Version** : **Git** est le standard de l'industrie.⁶⁴
- **Plateformes de Prompt Management** : Des outils spécialisés émergent pour faciliter ce processus, offrant des interfaces visuelles pour le versionnement, les tests A/B et la collaboration entre ingénieurs et experts métier non-techniques.
 - **PromptLayer**⁷⁰ : Permet de suivre, versionner et évaluer les prompts, agissant comme un CMS (Content Management System) pour les prompts.
 - **Maxim AI**⁶⁴ : Offre une plateforme de gestion de prompts de bout en bout, incluant

des pistes d'audit et des évaluations intégrées.

- **Latitude**⁶⁸ : Une plateforme open-source pour l'ingénierie de prompt qui s'intègre aux flux de travail Git.

L'adoption d'un tel flux de travail transforme le prompting d'un art empirique en une discipline d'ingénierie rigoureuse, une condition sine qua non pour l'intégration fiable des LLM dans des processus de conception critiques.

Chapitre 5 : Phase 2 - L'Étude Générative : Définition de l'Espace du Problème

Une fois le brief de conception initialisé, l'étape suivante consiste à le traduire en un problème mathématiquement solvable que le logiciel de conception générative peut interpréter. Cette phase est cruciale car elle définit les "règles du jeu" pour l'algorithme d'optimisation.

5.1. Traduction du Brief en un Modèle Solvable

Cette étape se déroule dans l'environnement de Conception Assistée par Ordinateur (CAO) et d'Ingénierie Assistée par Ordinateur (IAO). Elle consiste à définir les frontières géométriques et physiques du problème.

- **Géométrie à Préserver (*Preserve Geometry*)** : Ce sont les zones de la conception qui sont non négociables et doivent exister dans la forme finale. Il s'agit généralement des interfaces fonctionnelles : points de fixation, surfaces de contact, passages pour des fluides ou des câbles. Ces zones sont "gelées" et l'algorithme ne peut pas y retirer de matière.⁹
- **Géométrie d'Obstacle (*Obstacle Geometry*)** : Ce sont les volumes que la conception finale ne doit pas pénétrer. Ils représentent l'espace occupé par d'autres composants de l'assemblage, les chemins d'accès pour les outils de montage ou de maintenance, ou encore l'enveloppe de mouvement de pièces adjacentes.⁹ L'algorithme traitera ces zones comme des "keep-out zones".
- **Charges et Contraintes (*Loads and Constraints*)** : C'est la traduction des conditions de fonctionnement réelles en un modèle de simulation. L'ingénieur applique des forces, des pressions, des moments, des accélérations ou des conditions thermiques aux géométries à préserver. Il définit également les appuis, c'est-à-dire les zones où la pièce est fixée à son environnement. Il est possible de définir plusieurs cas de charge (*load cases*) pour simuler différentes conditions de fonctionnement (par exemple, un freinage,

une accélération, un virage pour une pièce de suspension automobile).⁹

5.2. Encodage des Contraintes de Fabrication : La Clé de la Faisabilité

C'est l'étape qui ancre la conception générative dans la réalité industrielle. Sans elle, l'algorithme pourrait produire des formes magnifiques mais infabriquables. Pour chaque procédé de fabrication envisagé, des contraintes spécifiques doivent être activées.

- **Usinage CNC (CNC Machining) :**
 - **Contrainte : Extrusion Linéaire / Direction de Fraisage.** Cette contrainte garantit que la pièce peut être usinée à partir d'une ou plusieurs directions sans que l'outil n'entre en collision avec la pièce elle-même. Pour un usinage 3 axes, on définit une seule direction d'usinage. Pour un usinage 5 axes, les contraintes sont plus complexes mais suivent le même principe d'accessibilité de l'outil.⁵
 - **Implémentation :** L'utilisateur définit un vecteur représentant l'axe de l'outil. L'algorithme ne génère alors que des formes qui ne présentent pas de contre-dépouilles par rapport à cette direction.
- **Fabrication Additive (FDM/SLM) :**
 - **Contrainte : Angle de Surplomb (Overhang Angle).** La plupart des procédés de fabrication additive construisent la pièce couche par couche. Les surfaces en surplomb prononcé (proches de l'horizontale) ont tendance à s'effondrer si elles ne sont pas soutenues par des structures de support temporaires. Cette contrainte limite l'angle des surplombs à une valeur critique (typiquement 45 degrés par rapport à la verticale) pour créer des pièces "auto-portantes" qui nécessitent peu ou pas de supports.⁴⁴
 - **Implémentation :** L'utilisateur définit un vecteur de direction de construction (généralement l'axe Z) et un angle maximal. L'algorithme pénalise ou interdit la formation de surfaces qui violent cette contrainte.⁴⁵
- **Moulage par Injection / Fonderie (Injection Molding/Casting) :**
 - **Contrainte : Direction de Démoulage (Pull Direction) et Angle de Dépouille (Draft Angle).** Pour qu'une pièce puisse être éjectée d'un moule rigide, elle ne doit pas présenter de contre-dépouilles par rapport à la direction d'ouverture du moule. De plus, les parois parallèles à la direction de démoulage doivent avoir une légère inclinaison (angle de dépouille) pour faciliter l'éjection.⁹
 - **Implémentation :** L'utilisateur définit un plan de joint et une ou deux directions de démoulage. L'algorithme est contraint de générer une géométrie qui peut être divisée en deux moitiés sans contre-dépouilles. Certains outils avancés permettent également d'imposer un angle de dépouille minimum.⁴³

Ces contraintes sont fondamentales car elles guident l'optimisation vers des solutions non

seulement performantes, mais aussi économiquement viables à produire.⁴⁶

5.3. Frameworks et Logiciels

Le marché offre une gamme de solutions logicielles pour la mise en œuvre d'études de conception générative.

- **Solutions Commerciales Intégrées** : Ces plateformes offrent un flux de travail complet, de la modélisation CAO à l'optimisation et la simulation, dans un environnement unifié. Elles sont généralement plus matures, robustes et faciles à utiliser.
 - **Autodesk Fusion** : Très populaire pour son intégration cloud et son interface accessible. Il propose des contraintes pour la fabrication additive, l'usinage (2.5, 3 et 5 axes) et le moulage (fonderie, moulage par injection).²
 - **PTC Creo** : Offre des extensions puissantes (Generative Topology Optimization - GTO, et Generative Design Extension - GDX) bien intégrées dans l'environnement CAO paramétrique. Il propose des contraintes similaires, notamment pour l'extrusion linéaire et les lignes de joint.⁷
 - **Siemens NX** : Intègre des outils de conception générative et d'optimisation topologique dans sa suite de développement produit, tirant parti de son solveur Simcenter.⁷
 - **nTop (anciennement nTopology)** : Une plateforme spécialisée et très puissante qui se concentre sur les géométries complexes (treillis, surfaces implicites) et offre un contrôle granulaire sur l'ensemble du processus de conception générative via des flux de travail réutilisables.⁷⁷
 - **Altair Inspire/OptiStruct** : Altair est un pionnier de l'optimisation structurelle avec son solveur OptiStruct. Inspire fournit une interface utilisateur intuitive pour mettre en place des optimisations topologiques avec des contraintes de fabrication.⁸⁰
- **Approche Open Source** : Il n'existe pas encore de suite open source intégrée qui rivalise directement avec les solutions commerciales en termes de facilité d'utilisation et de complétude. Cependant, il est possible d'assembler un flux de travail en combinant plusieurs outils spécialisés. Cette approche offre une flexibilité maximale et aucun coût de licence, mais requiert une expertise technique plus élevée.
 - **Modélisation CAO : FreeCAD** est un modèleur paramétrique open source puissant et extensible via Python.⁸²
 - **Ateliers (Workbenches) pour FreeCAD** : Des add-ons développés par la communauté tentent d'intégrer des fonctionnalités de conception générative. **FEMbyGEN**⁸³ et **AMGeneration**⁸⁵ sont des projets notables qui s'appuient sur des solveurs d'optimisation topologique. Leur maturité peut varier.
 - **Bibliothèques Python d'Optimisation Topologique** : Pour un contrôle total, un ingénieur peut utiliser directement des bibliothèques Python. Celles-ci peuvent être

intégrées dans des scripts personnalisés.

- **DL4TO** ⁸⁶ : Basée sur PyTorch, elle est conçue pour la recherche à l'intersection du deep learning et de l'optimisation topologique.
- **OpenPisco** ⁸⁷ : Un framework modulaire développé pour la recherche et le développement en optimisation topologique, interfacé avec des solveurs éléments finis comme Code_Aster.
- **TopOpt (zfergus)** ⁸⁸ et les **codes de DTU** ⁸⁹ : Des implémentations plus fondamentales et éducatives des algorithmes d'OT, excellentes pour comprendre les mécanismes de base.

La faisabilité du projet dépend du choix de l'outil. Pour un déploiement industriel rapide, une solution commerciale est souvent privilégiée. Pour la recherche, l'éducation ou le développement d'outils personnalisés, une approche open source est viable et formatrice.

Chapitre 6 : Phase 3 - Naviguer dans l'Océan des Possibilités : Optimisation Multi-Objectif et Sélection

Après l'exécution de l'étude générative, l'ingénieur n'est pas confronté à une seule solution, mais à des dizaines, voire des centaines de conceptions candidates. Chacune est une réponse valide aux contraintes, mais elles représentent différents compromis entre les objectifs de performance. Cette phase consiste à analyser cet ensemble de solutions et à en sélectionner la plus appropriée.

6.1. Le Défi des Objectifs Conflictuels

En ingénierie, les problèmes sont rarement mono-objectif. Une conception idéale doit souvent satisfaire simultanément plusieurs critères qui sont en conflit les uns avec les autres. Par exemple, pour une pièce mécanique, on souhaite généralement ⁹⁰ :

- **Minimiser la masse** (pour réduire les coûts de matière et améliorer l'efficacité énergétique).
- **Maximiser la rigidité** (c'est-à-dire minimiser le déplacement sous charge).
- **Minimiser la contrainte maximale** (pour augmenter la durabilité et le facteur de sécurité).
- **Minimiser le coût de fabrication.**

Ces objectifs sont souvent contradictoires : une pièce plus rigide et plus résistante nécessite

généralement plus de matière, ce qui augmente sa masse et son coût.⁹¹ L'**Optimisation Multi-Objectif** (OMO, ou *Multi-Objective Optimization*, MOO) est le domaine qui traite de la résolution de ces problèmes complexes, où il n'existe pas une seule solution "parfaite", mais un ensemble de solutions de compromis.⁹³

6.2. Comprendre le Front de Pareto

Le concept central de l'OMO est le **Front de Pareto** (ou frontière de Pareto). Le front de Pareto est l'ensemble de toutes les solutions "non-dominées".⁹³

Une solution A "domine" une solution B si A est meilleure ou égale à B sur tous les objectifs, et strictement meilleure sur au moins un objectif. Une solution est dite "non-dominée" ou "Pareto-optimale" si aucune autre solution ne la domine. En d'autres termes, pour une solution sur le front de Pareto, il est impossible d'améliorer un objectif sans en dégrader au moins un autre.⁹⁶

Pour illustrer ce concept, imaginons un graphique à deux dimensions où l'on trace la masse (axe X, à minimiser) et le déplacement (axe Y, à minimiser) de chaque conception générée. Le front de Pareto formera une courbe des points les plus performants. Tout point sur cette courbe représente un compromis optimal : pour réduire davantage la masse, il faudra accepter un déplacement plus important, et vice-versa. Tout point qui n'est pas sur le front est sous-optimal, car il existera toujours un point sur le front qui est à la fois plus léger et plus rigide.⁹¹

Le logiciel de conception générative affiche souvent ces résultats sous forme de nuages de points interactifs, permettant à l'ingénieur de visualiser le front de Pareto et de comprendre les compromis inhérents à son problème de conception.⁹⁸

6.3. Méthodes de Notation et de Classement pour la Prise de Décision

Le front de Pareto fournit un ensemble de solutions mathématiquement équivalentes en termes d'optimalité. Cependant, pour le projet, une seule conception doit être choisie pour la fabrication. Il faut donc une méthode pour classer ces solutions en fonction des priorités spécifiques du projet.⁹⁹

- **Méthode de la Somme Pondérée (*Weighted Sum Method*)** : C'est l'approche la plus simple. On assigne un poids (une importance relative) à chaque objectif. Par exemple,

60% pour la masse et 40% pour la rigidité. Pour chaque conception, on calcule un score unique en additionnant les valeurs normalisées de chaque objectif, multipliées par leur poids. La conception avec le meilleur score est sélectionnée. Cette méthode est facile à mettre en œuvre mais peut être sensible à la forme du front de Pareto et à la mise à l'échelle des objectifs.⁹³

- **TOPSIS (*Technique for Order of Preference by Similarity to Ideal Solution*)** : C'est une méthode de prise de décision multicritères plus robuste et largement utilisée.¹⁰⁰ Le processus est le suivant ¹⁰¹ :
 1. **Normalisation** : Les valeurs de tous les objectifs pour toutes les solutions sont normalisées (généralement mises à l'échelle entre 0 et 1) pour les rendre comparables.
 2. **Pondération** : Les valeurs normalisées sont multipliées par les poids des objectifs.
 3. **Définition des solutions Idéale et Anti-Idéale** : La solution "idéale" est une solution hypothétique qui a la meilleure valeur pour chaque objectif. La solution "anti-idéale" est celle qui a la pire valeur pour chaque objectif.
 4. **Calcul des Distances** : Pour chaque conception réelle, on calcule sa distance euclidienne par rapport à la solution idéale et à la solution anti-idéale.
 5. **Calcul du Score de Proximité** : Un score final est calculé pour chaque conception, qui représente sa proximité relative à la solution idéale. La conception avec le score le plus élevé est celle qui est simultanément la plus proche de l'idéal et la plus éloignée de l'anti-idéal.
- **AHP (*Analytic Hierarchy Process*)** : Cette méthode peut être utilisée en amont pour déterminer les poids des objectifs de manière structurée. Elle consiste à demander au décideur de comparer les objectifs deux à deux (par exemple, "La masse est-elle légèrement plus importante, beaucoup plus importante, ou aussi importante que la rigidité?"). Une analyse mathématique de cette matrice de comparaison par paires permet de dériver un vecteur de poids cohérent.⁹⁹

Le tableau suivant illustre comment une matrice de décision peut être utilisée pour appliquer une méthode de somme pondérée et classer les conceptions candidates.

ID Conc eption	Mass e (kg)	Dépla ceme nt Max (mm)	Contr ainte Max (MPa)	Score Mass e (Norm .)	Score Dépla ceme nt (Norm .)	Score Contr ainte (Norm .)	Score Final Pondé ré	Rang
Poids				50%	30%	20%		
GEN-	0.45	0.85	250	1.00	0.15	0.67	0.68	2

001								
GEN-002	0.52	0.60	220	0.82	0.60	0.87	0.74	1
GEN-003	0.60	0.50	280	0.64	0.80	0.47	0.65	3
GEN-004	0.75	0.40	300	0.33	1.00	0.33	0.53	4

*Note : Les scores normalisés sont calculés de telle sorte que 1.0 représente la meilleure performance et 0.0 la pire pour chaque critère. Le score final pondéré est calculé comme suit : Score Final = (Score Masse * 0.5) + (Score Déplacement * 0.3) + (Score Contrainte * 0.2).*

Ce processus structuré transforme la sélection subjective d'une conception en une décision transparente, justifiable et alignée sur les objectifs stratégiques du projet.

Partie III : Raffinement et Validation du Prototype Numérique

Cette partie se concentre sur la transformation du concept sélectionné en un actif numérique final, prêt pour la production, en tirant à nouveau parti de l'automatisation et de l'IA.

Chapitre 7 : Phase 4 - Itération et Automatisation avec le Scripting Paramétrique

7.1. La Puissance de la Modélisation Paramétrique

La modélisation paramétrique est une approche de la CAO où la géométrie des modèles n'est

pas statique, mais est pilotée par des **paramètres** et des **relations**.¹⁰² Ces paramètres peuvent être des dimensions (longueur, rayon), des contraintes (parallélisme, tangence) ou des variables définies par l'utilisateur (épaisseur, nombre de trous). Lorsqu'un paramètre est modifié, le logiciel recalcule automatiquement l'ensemble du modèle pour refléter ce changement, en maintenant les relations définies.¹⁰³

Cette approche est fondamentale pour l'automatisation. Un modèle paramétrique bien construit est essentiellement un programme visuel. Au lieu de redessiner manuellement les variations, l'ingénieur peut explorer différentes configurations simplement en modifiant quelques valeurs clés.¹⁰⁴ C'est la base qui permet de connecter la logique de conception à des scripts externes pour une automatisation poussée.

7.2. Automatisation de la CAO avec des Scripts Python

La plupart des logiciels de CAO modernes exposent une **Interface de Programmation d'Application** (API), qui permet de contrôler le logiciel par le biais de code. Python est devenu le langage de script de facto dans de nombreux environnements d'ingénierie en raison de sa simplicité et de sa puissance.⁸²

Via l'API, un script Python peut effectuer presque toutes les actions qu'un utilisateur peut réaliser manuellement via l'interface graphique :

- Créer des esquisses 2D et des géométries 3D (boîtes, cylindres, extrusions, etc.).¹⁰⁶
- Modifier des objets existants en accédant à leurs paramètres.¹⁰⁷
- Effectuer des opérations booléennes (union, soustraction, intersection).
- Parcourir la structure d'un modèle pour en extraire des informations (par exemple, l'aire de chaque face).¹⁰⁶

Exemples d'API pour la CAO :

- **FreeCAD** : Entièrement scriptable en Python. L'utilisateur peut interagir directement avec le noyau géométrique Open CASCADE pour créer et manipuler des formes de manière programmatique.⁸²
- **Autodesk Fusion 360** : Offre une API complète accessible en Python et C++, permettant d'automatiser des tâches complexes, de créer des compléments (*add-ins*) personnalisés et de manipuler des modèles paramétriques.¹⁰⁵

Cette capacité de scriptage est essentielle pour la phase de raffinement. Une fois qu'une conception est sélectionnée à partir de l'étude générative, un script paramétrique peut être créé pour reconstruire sa topologie de manière propre et modifiable. Ce script peut ensuite

être utilisé pour des études d'optimisation fine ou pour générer rapidement des variantes.¹¹¹

7.3. Les LLM comme "Copilotes CAO" : Génération et Modification de Scripts

L'une des applications les plus avancées des LLM en ingénierie est leur utilisation pour écrire du code qui pilote des logiciels de CAO. Un ingénieur peut décrire en langage naturel la géométrie ou la modification souhaitée, et le LLM tente de générer le script Python correspondant.¹¹³

Faisabilité et Défis Actuels :

Les recherches et les expériences pratiques montrent que cette approche est prometteuse mais encore immature.¹¹⁶

- **Succès** : Les LLM, en particulier ceux spécialisés dans le code (comme les modèles Code Llama ou les versions "Code Interpreter" de GPT), peuvent générer avec succès des scripts pour des géométries simples (créer un cube, percer un trou, ajouter un congé).¹¹⁸
- **Échecs** : Pour des tâches complexes, les LLM ont tendance à "halluciner" des fonctions d'API qui n'existent pas, à produire des erreurs de syntaxe, ou à générer un code qui s'exécute sans erreur mais ne produit pas la géométrie attendue.¹¹⁶ La précision mathématique requise par la CAO est un défi pour des modèles probabilistes.¹¹⁷

Le flux de travail le plus efficace à l'heure actuelle est une **boucle de débogage interactive**

¹¹⁴ :

1. L'ingénieur fournit un prompt au LLM.
2. Le LLM génère un script CAO.
3. Le script est exécuté dans l'environnement CAO (par exemple, FreeCAD en mode "headless").
4. Si le script échoue, le message d'erreur est capturé.
5. L'ingénieur soumet un nouveau prompt au LLM, incluant la demande initiale, le code généré et le message d'erreur, en lui demandant de corriger le script.
6. Le processus est répété jusqu'à l'obtention d'un résultat valide.

L'utilisation la plus pragmatique et la plus robuste des LLM pour le scriptage CAO n'est pas la génération de pièces complexes à partir de zéro, mais plutôt l'**assistance à un développeur humain** ou l'**automatisation de modifications paramétriques**. Un LLM excelle dans la manipulation de code existant. Par exemple, un prompt tel que *"Dans le script `bracket_model.py`, modifie le paramètre `thickness` à 8 mm et change le matériau en `'Titanium-Ti-6Al-4V'`"* est une tâche de manipulation de texte bien définie, beaucoup plus fiable pour un LLM que *"Crée un support aérospatial complexe"*.¹²¹ Cela positionne le LLM comme un "copilote" qui accélère les tâches de modification et d'itération sur un modèle

paramétrique bien structuré, plutôt que comme un concepteur autonome.

Frameworks et Outils :

- **API de scriptage** : FreeCAD Python API ¹⁰⁶, Fusion 360 API. ¹⁰⁹
- **Bibliothèques Python pour la CAO** : **CadQuery** est une bibliothèque populaire qui permet de créer des modèles CAO paramétriques avec un code Python plus intuitif et moins verbeux que les API natives. Des recherches sont en cours pour affiner des LLM spécifiquement pour la génération de code CadQuery. ¹²²
- **Projets de recherche** : **BlenderLLM** est un projet visant à affiner un LLM pour générer des scripts Python pour le logiciel de modélisation 3D Blender. ¹¹⁵ Des approches similaires sont explorées pour FreeCAD. ¹²³

Chapitre 8 : Phase 5 - Validation Finale et Dossier Technique

Cette dernière phase du projet consiste à vérifier rigoureusement la performance de la conception finale et à compiler un dossier technique complet qui documente le processus et quantifie les gains obtenus.

8.1. L'Importance de la Vérification Indépendante

Le processus de conception générative utilise des solveurs AEF rapides, souvent simplifiés, pour pouvoir exécuter des milliers d'itérations dans un temps raisonnable. ³¹ De plus, le résultat initial de l'optimisation est un maillage, qui est ensuite converti en un modèle CAO lisse (B-rep). Cette conversion peut introduire de légères variations géométriques.

Pour ces raisons, il est impératif de réaliser une **analyse AEF de validation indépendante et de haute fidélité** sur le modèle CAO final et affiné. ¹⁴ Cette simulation, menée avec un maillage fin et des paramètres de solveur précis, sert de "signature" finale qui confirme que la conception répond à toutes les exigences de performance (contraintes, déformations, facteur de sécurité) dans des conditions réalistes. ¹²⁴ Elle permet de s'assurer que les hypothèses simplificatrices de la phase d'optimisation n'ont pas conduit à une conception non conforme et de valider la performance du modèle B-rep, et non seulement celle du maillage initial. ¹²⁶

8.2. Estimation Automatisée de la Masse et des Coûts

Pour compléter le dossier technique, des indicateurs métier clés doivent être calculés de manière fiable.

- **Masse** : C'est le calcul le plus simple. Tout logiciel de CAO peut calculer instantanément le volume de la pièce. En lui attribuant une densité de matériau, la masse est obtenue par une simple multiplication.
- **Coût de Production** : L'estimation des coûts est plus complexe mais peut être largement automatisée. Des logiciels spécialisés, souvent appelés outils de *Cost Engineering*, analysent la géométrie 3D de la pièce et estiment les coûts de fabrication en se basant sur ¹²⁷ :
 - Le **procédé de fabrication** : Les modèles de coût sont radicalement différents pour l'usinage, le moulage ou l'impression 3D.
 - Les **caractéristiques géométriques** : Le logiciel identifie les caractéristiques (trous, poches, nervures) et estime le temps nécessaire pour les produire (temps d'usinage, temps de remplissage du moule).¹²⁹
 - Le **coût de la matière première**, le **coût de l'outillage** (pour le moulage), et les **coûts de main-d'œuvre et de machine**.

Ces outils permettent d'obtenir rapidement une estimation de coût objective, qui est cruciale pour évaluer la viabilité économique de la conception générée.¹³¹

Frameworks et Outils :

- **aPriori** ¹²⁷ : Une plateforme leader qui utilise des jumeaux numériques d'usines pour simuler les processus de fabrication et générer des estimations de coûts détaillées.
- **Paperless Parts** ¹²⁸ : Une plateforme de devis automatisé qui analyse la géométrie CAO pour estimer les coûts de fabrication.
- **Costimator** ¹²⁹ : Un logiciel d'estimation des coûts qui contient des centaines de modèles de coûts pour divers processus de fabrication.
- **Simus Systems** ¹³⁰ : S'intègre directement dans les systèmes CAO pour fournir des calculs de coûts en temps réel basés sur l'analyse de la géométrie 3D.

8.3. Assemblage du Dossier Technique

Le livrable final du projet est un dossier technique qui non seulement présente la conception optimisée, mais justifie également sa supériorité par une comparaison quantitative rigoureuse avec une conception de référence ("baseline"). Cette baseline peut être la pièce existante que l'on cherche à améliorer, ou une conception réalisée avec des méthodes traditionnelles.

Le tableau comparatif ci-dessous est l'élément central de ce dossier. Il synthétise les résultats du projet et communique sa valeur aux parties prenantes de manière claire et concise. Des études de cas réels, comme celle du support de siège de General Motors qui a permis une réduction de masse de 40% et une augmentation de la résistance de 20%, servent de référence pour les gains potentiels.¹³² D'autres exemples dans l'industrie aéronautique montrent également des réductions de masse significatives sur des supports et autres composants structurels.⁷

Tableau : Analyse Comparative des Performances : Conception de Référence vs. Conception Générative

Métrique	Conception de Référence (Baseline)	Conception Générative	Amélioration (%)
Propriétés Géométriques et Massiques			
Nombre de pièces	8	1	-87.5%
Masse totale (kg)	1.25	0.75	-40.0%
Performance Structurelle (Validation AEF)			
Contrainte de Von Mises maximale (MPa)	350	280	-20.0%
Déplacement maximal (mm)	0.50	0.45	-10.0%
Facteur de sécurité minimal	1.2	1.5	+25.0%
Indicateurs Métier et Fabrication			

Coût de production estimé par unité (€)	45	55 (pour SLM)	+22.2%
Temps de production estimé par unité (heures)	2.5 (assemblage inclus)	4.0 (impression 3D)	+60.0%
Score de fabricabilité (1-10)	9	7	-22.2%

Note : Les valeurs de ce tableau sont illustratives et basées sur des cas d'étude typiques comme celui de General Motors.¹³² Elles montrent un compromis courant : la conception générative offre des gains de performance spectaculaires (masse, résistance) mais peut entraîner un coût unitaire ou un temps de production plus élevé, en particulier si elle repose sur la fabrication additive. Le dossier technique doit analyser ce compromis et justifier le choix final en fonction des priorités du projet (la performance prime-t-elle sur le coût?).

Ce dossier constitue la preuve documentée de la valeur ajoutée du processus de conception augmentée, transformant les promesses de l'IA en résultats d'ingénierie mesurables et validés.¹³⁵

Conclusion : La Synthèse de l'Intention Humaine et de l'Intelligence Artificielle

Ce rapport a détaillé un flux de travail complet pour la conception de produits augmentée par l'intelligence artificielle, depuis l'idéation assistée par les LLM jusqu'à la validation d'un prototype numérique par AEF. L'analyse de chaque phase, des technologies sous-jacentes et des frameworks disponibles démontre que nous sommes à l'aube d'une transformation majeure dans les métiers de l'ingénierie.

Le processus décrit ne vise pas à remplacer l'ingénieur. Au contraire, il élève son rôle. L'ingénieur n'est plus seulement un dessinateur de géométrie, limité par son expérience et son intuition. Il devient un **architecte de problèmes**, dont la compétence principale est de traduire des besoins fonctionnels complexes en un ensemble précis d'objectifs, de règles et de contraintes. Il devient un **curateur de solutions**, capable de naviguer dans un vaste espace de possibilités généré par l'IA pour identifier le compromis optimal qui répond le mieux aux impératifs techniques et commerciaux. Enfin, il devient un **orchestrateur de systèmes intelligents**, combinant des outils d'IA générative, des solveurs d'optimisation, des

plateformes de simulation et des scripts d'automatisation pour créer un flux de travail intégré et hautement efficace.

La faisabilité de ce projet est élevée, à condition de s'appuyer sur les bons outils et de suivre une méthodologie rigoureuse. Les solutions logicielles commerciales offrent des environnements intégrés robustes, tandis que l'écosystème open source fournit des briques technologiques puissantes pour ceux qui souhaitent un contrôle plus granulaire. La gestion rigoureuse des prompts, traitée comme une discipline de génie logiciel, est une condition non négociable pour garantir la fiabilité et la reproductibilité des phases amont.

Le dossier technique final, avec sa comparaison quantitative entre la conception de référence et la solution générative, constitue la démonstration ultime de la valeur de cette approche. Les gains potentiels en termes de performance, de réduction de masse et d'innovation sont substantiels, comme l'ont démontré de nombreuses études de cas industriels. La fusion de l'expertise humaine, qui définit l'intention et le contexte, avec la puissance de calcul de l'intelligence artificielle, qui explore et optimise, ne représente rien de moins que l'avenir du développement de produits à haute performance.

Sources des citations

1. Conception Générative : quand l'IA aide les concepteurs - KapTech, consulté le octobre 19, 2025, <https://kaptech.be/?p=3128>
2. Conception générative pour la fabrication | Autodesk Fusion, consulté le octobre 19, 2025, <https://www.autodesk.com/fr/solutions/generative-design/manufacturing>
3. What is Generative Design | Tools Software - Autodesk, consulté le octobre 19, 2025, <https://www.autodesk.com/solutions/generative-design>
4. Topology Optimization vs. Generative Design: Which Modeling Tool Should You Choose?, consulté le octobre 19, 2025, <https://www.3dnatives.com/en/topology-optimization-vs-generative-design-190920244/>
5. Generative Design, Prototyping, and CNC Milling - DATRON, consulté le octobre 19, 2025, <https://www.datron.com/resources/blog/generative-design-prototyping-and-cnc-milling/>
6. formlabs.com, consulté le octobre 19, 2025, <https://formlabs.com/fr/blog/conception-generative/#:~:text=D%C3%A9finition%20de%20la%20conception%20g%C3%A9n%C3%A9rative&text=Contrairement%20%C3%A0%20la%20conception%20classique,artificielle%20pour%20g%C3%A9n%C3%A9rer%20le%20mod%C3%A8le.>
7. Generative Design 101 - Formlabs, consulté le octobre 19, 2025, <https://formlabs.com/blog/generative-design/>
8. Generative Design for Manufacturing | Autodesk Fusion, consulté le octobre 19, 2025, <https://www.autodesk.com/solutions/generative-design/manufacturing>
9. Conception générative : une nouvelle façon de concevoir | PTC (FR), consulté le

- octobre 19, 2025,
<https://www.ptc.com/fr/blogs/cad/generative-design-new-way-of-designing>
10. Topology Optimization Vs. Generative Design | Diabatix, consulté le octobre 19, 2025,
<https://www.diabatix.com/blog/generative-design-topology-optimization-and-parametric-optimization-what-are-the-differences>
 11. Optimisation Topologique - 3DnCut, consulté le octobre 19, 2025,
<https://3dncut.com/optimisation-topologique/>
 12. Optimisation topologique, - Eduscol STI, consulté le octobre 19, 2025,
<https://sti.eduscol.education.fr/sites/eduscol.education.fr.sti/files/ressources/techniques/10977/10977-203-p18.pdf>
 13. Topology Optimization 101: How to Use Algorithmic Models to Create Lightweight Design, consulté le octobre 19, 2025,
<https://formlabs.com/blog/topology-optimization/>
 14. Qu'est-ce que l'optimisation de la topologie ? | Ansys, consulté le octobre 19, 2025,
<https://www.ansys.com/fr-fr/simulation-topics/what-is-topology-optimization>
 15. Optimisation topologique - Wikipédia, consulté le octobre 19, 2025,
https://fr.wikipedia.org/wiki/Optimisation_topologique
 16. Le b.a-ba de l'optimisation topologique : comment utiliser des modèles algorithmiques pour concevoir des pièces légères | Formlabs, consulté le octobre 19, 2025, <https://formlabs.com/fr/blog/optimisation-topologique/>
 17. Generative Design et Optimisation Topologique - Digicad, consulté le octobre 19, 2025, <https://www.digicad.fr/generative-design/>
 18. Optimisation topologique vs. Conception générative : r/IsaacArthur - Reddit, consulté le octobre 19, 2025,
https://www.reddit.com/r/IsaacArthur/comments/114z6gl/topology_optimization_vs_generative_design/?tl=fr
 19. Topology Optimization VS Generative Design - Neural Concept, consulté le octobre 19, 2025,
<https://www.neuralconcept.com/post/topology-optimization-vs-generative-design>
 20. Generative Design Versus Topology Optimization | by 3DHEALS - Medium, consulté le octobre 19, 2025,
<https://medium.com/healthcare-3d-printing-stories/generative-design-versus-topology-optimization-ed549dbf19b4>
 21. Topological optimization generative design and how they differ - Autodesk Forums, consulté le octobre 19, 2025,
<https://forums.autodesk.com/t5/fusion-design-validate-document/topological-optimization-generative-design-and-how-they-differ/td-p/13128945>
 22. Progress and recent trends in generative design - ResearchGate, consulté le octobre 19, 2025,
https://www.researchgate.net/publication/343659110_Progress_and_recent_trends_in_generative_design
 23. What is Finite Element Analysis (FEA)? - Ansys, consulté le octobre 19, 2025,

- <https://www.ansys.com/simulation-topics/what-is-finite-element-analysis>
24. What is Finite Element Analysis? FEA Explained | Dassault Systèmes, consulté le octobre 19, 2025, <https://www.3ds.com/store/cad/finite-element-analysis>
 25. Finite element method - Wikipedia, consulté le octobre 19, 2025, https://en.wikipedia.org/wiki/Finite_element_method
 26. What Is FEM & FEA Explained | Finite Element Method - SimScale, consulté le octobre 19, 2025, <https://www.simscale.com/blog/what-is-finite-element-method/>
 27. The Finite Element Method (FEM) – A Beginner's Guide - Jousef Murad, consulté le octobre 19, 2025, <https://www.engineered-mind.com/fem/the-finite-element-method-beginners-guide/>
 28. Introduction to Finite Element Analysis (FEA) or Finite Element Method (FEM) - Engineering and Computer Science, consulté le octobre 19, 2025, https://www.engr.uvic.ca/~mech410/lectures/FEA_Theory.pdf
 29. Generative Part Structural Analysis Fundamentals, consulté le octobre 19, 2025, https://birimler.dpu.edu.tr/app/views/panel/ckfinder/userfiles/74/files/fkarakoc/bd/ma/1-catia_V5_Generative_Part_Structural_Analysis_Fundamentals.pdf
 30. Finite Element Analysis: How to Validate Your Designs With Simulations, consulté le octobre 19, 2025, <https://blog.hagerman.com/finite-element-analysis-how-to-validate-your-designs-with-simulations>
 31. Fusion 360 Introduction to Generative Design | Autodesk University, consulté le octobre 19, 2025, <https://www.autodesk.com/autodesk-university/article/Fusion-360-Introduction-Generative-Design>
 32. What is DFMA? (Design for Manufacturing & Assembly) | Boothroyd Dewhurst, consulté le octobre 19, 2025, <https://www.dfma.com/resources/what-is-dfma.asp>
 33. Design for Manufacturing and Assembly (DFMA) | Principles Explained - Fractory, consulté le octobre 19, 2025, <https://fractory.com/design-for-manufacturing-and-assembly-dfma/>
 34. Design for Manufacture & Assembly (DFMA) - Siemens Digital Industries Software, consulté le octobre 19, 2025, <https://www.sw.siemens.com/en-US/technology/design-for-manufacturing-assembly-dfma/>
 35. Design for Manufacturing and Assembly I: General Principles - Cloudfront.net, consulté le octobre 19, 2025, https://d2t1xqejoj9utc.cloudfront.net/files/15802/DFMA_I.pdf?1352522847
 36. Design for Manufacturing / Assembly (DFM/DFA) - Quality-One, consulté le octobre 19, 2025, <https://quality-one.com/dfm-dfa/>
 37. Design for Manufacturing Software | Autodesk, consulté le octobre 19, 2025, <https://www.autodesk.com/solutions/design-for-manufacturing-software>
 38. DFM & DFA Principles Can Lower Manufacturing Costs and Improve Efficiency, consulté le octobre 19, 2025, <https://www.kellertechnology.com/blog/utilizing-dfm-and-dfa-principles/>

39. Generative AI For PCB Design with Allegro X AI, consulté le octobre 19, 2025, <https://resources.pcb.cadence.com/blog/2024-generative-ai-for-pcb-design-with-allegro-x-ai>
40. Manufacturability constraint derivation and use in engineering design... - ResearchGate, consulté le octobre 19, 2025, https://www.researchgate.net/figure/Manufacturability-constraint-derivation-and-use-in-engineering-design-the-design_fig3_327321651
41. Incorporating Manufacturing Constraints in Topology Optimization Methods: A Survey | Request PDF - ResearchGate, consulté le octobre 19, 2025, https://www.researchgate.net/publication/320849159_Incorporating_Manufacturing_Constraints_in_Topology_Optimization_Methods_A_Survey
42. Generative Design: The Latest Tool in the Engineer's Toolbox - pdsvision, consulté le octobre 19, 2025, <https://pdsvision.com/wp-content/uploads/2022/12/E-book-Generative-Design-The-Latest-Tool-in-the-Engineers-Toolbox.pdf>
43. Design Criteria - PTC Support Portal, consulté le octobre 19, 2025, https://support.ptc.com/help/creo/creo_pma/r11.0/usascii/generative_design/design_criteria.html
44. Topology optimization considering overhang constraint in additive manufacturing | Request PDF - ResearchGate, consulté le octobre 19, 2025, https://www.researchgate.net/publication/330799900_Topology_optimization_considering_overhang_constraint_in_additive_manufacturing
45. How to use the Overhang Constraint - nTop Support, consulté le octobre 19, 2025, <https://support.ntop.com/hc/en-us/articles/360060738793-How-to-use-the-overhang-constraint-in-Topology-Optimization>
46. Topology optimization with manufacturing constraints - Paulino's group, consulté le octobre 19, 2025, https://paulino.princeton.edu/journal_papers/2016/AES_16_TopologyOptimizationWithManufacturing.pdf
47. Intelligence artificielle générative - Wikipédia, consulté le octobre 19, 2025, https://fr.wikipedia.org/wiki/Intelligence_artificielle_g%C3%A9n%C3%A9rative
48. How AI and LLMs Are Revolutionizing Physical Product Development - mHUB Chicago, consulté le octobre 19, 2025, <https://www.mhubchicago.com/blog/how-ai-and-llms-are-revolutionizing-physical-product-development>
49. Que sont les grands modèles linguistiques (LLM) ? | Microsoft Azure, consulté le octobre 19, 2025, <https://azure.microsoft.com/fr-ca/resources/cloud-computing-dictionary/what-are-large-language-models-llms>
50. Research directions for using LLM in software requirement engineering: a systematic review, consulté le octobre 19, 2025, <https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1519437/full>
51. Generative large language models in engineering design: opportunities and

- challenges, consulté le octobre 19, 2025,
https://www.researchgate.net/publication/380653779_Generative_large_language_models_in_engineering_design_opportunities_and_challenges
52. Product requirement document generation using LLM task oriented dialogue - GitHub Gist, consulté le octobre 19, 2025,
<https://gist.github.com/Dowwie/151d8efea738ea486ddec9208ddb3a19>
53. Prompt Engineering Guide, consulté le octobre 19, 2025,
<https://www.promptingguide.ai/>
54. What is Prompt Engineering? - AI Prompt Engineering Explained - AWS, consulté le octobre 19, 2025, <https://aws.amazon.com/what-is/prompt-engineering/>
55. Prompt Engineering Techniques | IBM, consulté le octobre 19, 2025,
<https://www.ibm.com/think/topics/prompt-engineering-techniques>
56. Zero-Shot, One-Shot, and Few-Shot Prompting, consulté le octobre 19, 2025,
https://learnprompting.org/docs/basics/few_shot
57. Prompt engineering techniques - Azure OpenAI | Microsoft Learn, consulté le octobre 19, 2025,
<https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/prompt-engineering>
58. What is few shot prompting? - IBM, consulté le octobre 19, 2025,
<https://www.ibm.com/think/topics/few-shot-prompting>
59. Prompt design strategies | Gemini API | Google AI for Developers, consulté le octobre 19, 2025, <https://ai.google.dev/gemini-api/docs/prompting-strategies>
60. Chain-of-Thought Prompting | Prompt Engineering Guide, consulté le octobre 19, 2025, <https://www.promptingguide.ai/techniques/cot>
61. AI Prompting (2/10): Chain-of-Thought Prompting—4 Methods for Better Reasoning - Reddit, consulté le octobre 19, 2025,
https://www.reddit.com/r/ChatGPTPromptGenius/comments/1if2dai/ai_prompting_210_chainofthought_prompting4/
62. General Tips for Designing Prompts - Prompt Engineering Guide, consulté le octobre 19, 2025, <https://www.promptingguide.ai/introduction/tips>
63. Overview of prompting strategies | Generative AI on Vertex AI - Google Cloud, consulté le octobre 19, 2025,
<https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/prompt-design-strategies>
64. 5 Tools for Versioning Prompts: Ensuring Consistency at Scale - DEV Community, consulté le octobre 19, 2025,
https://dev.to/kuldeep_paul/5-tools-for-versioning-prompts-ensuring-consistency-at-scale-3fme
65. The Importance of Prompt Versioning in 2025 - FeatBit, consulté le octobre 19, 2025,
<https://www.featbit.co/articles2025/importance-of-prompt-versioning-2025>
66. GitHub flow - GitHub Docs, consulté le octobre 19, 2025,
<https://docs.github.com/en/get-started/using-github/github-flow>
67. Gitflow Workflow | Atlassian Git Tutorial, consulté le octobre 19, 2025,
<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

68. Prompt Versioning: Best Practices - Ghost, consulté le octobre 19, 2025, <https://latitude-blog.ghost.io/blog/prompt-versioning-best-practices/>
69. I Built PromptOps: Git-Native Prompt Management for Production LLM Workflows - Medium, consulté le octobre 19, 2025, <https://medium.com/@jision/i-built-promptops-git-native-prompt-management-for-production-llm-workflows-ae49d1faa628>
70. PromptLayer - Your workbench for AI engineering. Platform for prompt management, prompt evaluations, and LLM observability, consulté le octobre 19, 2025, <https://www.promptlayer.com/>
71. How to Integrate Prompt Versioning with LLM Workflows - Ghost, consulté le octobre 19, 2025, <https://latitude-blog.ghost.io/blog/how-to-integrate-prompt-versioning-with-llm-workflows/>
72. (PDF) Generative design for additive manufacturing (G-DFAM): An explorative study of aerospace brackets - ResearchGate, consulté le octobre 19, 2025, https://www.researchgate.net/publication/370975072_Generative_design_for_additive_manufacturing_G-DFAM_An_explorative_study_of_aerospace_brackets
73. Fusion Generative Design: How to Add Loads and Constraints - YouTube, consulté le octobre 19, 2025, https://www.youtube.com/watch?v=SBvx_kqKx_A
74. Undercut and Overhang Angle Control in Topology Optimization: a Density Gradient based Integral Approach, consulté le octobre 19, 2025, https://cdm.me.wisc.edu/pub/PUP_out.pdf
75. Creo Parametric - Topology Optimization - Manufacturing Constraints - YouTube, consulté le octobre 19, 2025, <https://www.youtube.com/watch?v=N-0XWVFr42Y>
76. The Impact of Additive Manufacturing Constraints and Design Objectives on Structural Topology Optimization - MDPI, consulté le octobre 19, 2025, <https://www.mdpi.com/2076-3417/13/18/10161>
77. Generative Engineering: The AI-Powered Future of How Things Are Made | All3DP Pro, consulté le octobre 19, 2025, <https://all3dp.com/1/the-best-generative-design-software/>
78. Autodesk Fusion Generative Design - MF Software - Moldflow, consulté le octobre 19, 2025, <https://www.moldflow.eu/autodesk-fusion-generative-design/?lang=en>
79. Powerful generative design software - nTop, consulté le octobre 19, 2025, <https://www.ntop.com/software/capabilities/generative-design/>
80. Topology Optimization in 3D Printing: A Guide - Sculpteo, consulté le octobre 19, 2025, <https://www.sculpteo.com/en/3d-learning-hub/create-3d-file/topology-optimization/>
81. Design with topology optimization for Additive Manufacturing of a racing car component - DSpace@NTUA, consulté le octobre 19, 2025, <https://dspace.lib.ntua.gr/xmlui/bitstream/handle/123456789/61346/Design%20with%20topology%20optimization%20for%20Additive%20Manufacturing%20of%20a%20racing%20car%20component.pdf?sequence=1&isAllowed=y>
82. Your own 3D parametric modeler - FreeCAD, consulté le octobre 19, 2025,

- <https://www.freecad.org/features.php?lang=en>
83. Serince/FEMbyGEN: A FreeCAD module that uses Generative Design to design optimally for Additive Manufacturing - GitHub, consulté le octobre 19, 2025, <https://github.com/Serince/FEMbyGEN>
 84. FreeCAD Live #2: Generative Design & Optimization in FreeCAD - YouTube, consulté le octobre 19, 2025, <https://www.youtube.com/watch?v=viP9JBiU1R8>
 85. MightyBucket/AMGeneration: A FreeCAD module that uses Generative Design to design optimally for Additive Manufacturing - GitHub, consulté le octobre 19, 2025, <https://github.com/MightyBucket/AMGeneration>
 86. DL4TO is a Python library for 3D topology optimization that is based on PyTorch and allows easy integration with neural networks. - GitHub, consulté le octobre 19, 2025, <https://github.com/dl4to/dl4to>
 87. OpenPISCO, consulté le octobre 19, 2025, <https://openpisco.irt-systemx.fr/>
 88. zfergus/topopt: A Python Library for Topology Optimization - GitHub, consulté le octobre 19, 2025, <https://github.com/zfergus/topopt>
 89. Topology optimization codes written in Python - TopOpt, consulté le octobre 19, 2025, <https://www.topopt.mek.dtu.dk/apps-and-software/topology-optimization-codes-written-in-python>
 90. Optimisation multiobjectif - Patrick Siarry , Yann Collette - Librairie Eyrolles, consulté le octobre 19, 2025, <https://www.eyrolles.com/Informatique/Livre/optimisation-multiobjectif-9782212111682/>
 91. Multi-Objective Optimization Applications in Chemical Process Engineering: Tutorial and Review - MDPI, consulté le octobre 19, 2025, <https://www.mdpi.com/2227-9717/8/5/508>
 92. Multi-objective optimization in machine learning assisted materials design and discovery, consulté le octobre 19, 2025, <https://www.oaepublish.com/articles/jmi.2024.108>
 93. Full article: A review of multi-objective optimization: Methods and its applications, consulté le octobre 19, 2025, <https://www.tandfonline.com/doi/full/10.1080/23311916.2018.1502242>
 94. MOOSE-FS - Optimisation multi-objectifs pour la sélection de caractéristiques d'ensemble, consulté le octobre 19, 2025, <https://www.hes-so.ch/recherche-innovation/projets-de-recherche/detail-projet/moose-fs-optimisation-multi-objectifs-pour-la-selection-de-caracteristiques-de-nsemble>
 95. Pareto front - Wikipedia, consulté le octobre 19, 2025, https://en.wikipedia.org/wiki/Pareto_front
 96. Multi-objective optimization - Wikipedia, consulté le octobre 19, 2025, https://en.wikipedia.org/wiki/Multi-objective_optimization
 97. Pareto Front Visualization for Engineering Design - Digital Library, consulté le octobre 19, 2025, <https://diglib.eg.org/bitstream/handle/10.1111/cgf13990/v39i3pp405-416.pdf>
 98. (PDF) PAVED: Pareto Front Visualization for Engineering Design - ResearchGate,

- consulté le octobre 19, 2025,
https://www.researchgate.net/publication/342200252_PAVED_Pareto_Front_Visualization_for_Engineering_Design
99. Ranking and sorting the Pareto front in optimal shell structure design - ResearchGate, consulté le octobre 19, 2025,
https://www.researchgate.net/publication/396366390_Ranking_and_sorting_the_Pareto_front_in_optimal_shell_structure_design
100. Application and Analysis of Methods for Selecting an Optimal Solution from the Pareto-Optimal Front obtained by Multiobjective Optimization | Industrial & Engineering Chemistry Research - ACS Publications, consulté le octobre 19, 2025,
<https://pubs.acs.org/doi/10.1021/acs.iecr.6b03453>
101. Optimization of Printing Parameters for Biopolymer FDM Components Using Decision-Making Algorithms - ResearchGate, consulté le octobre 19, 2025,
https://www.researchgate.net/publication/395892235_Optimization_of_Printing_Parameters_for_Biopolymer_FDM_Components_Using_Decision-Making_Algorithms
102. What is Parametric Modeling? | Dassault Systèmes, consulté le octobre 19, 2025,
<https://www.3ds.com/store/cad/parametric-modeling>
103. Parametric Modeling Software - Autodesk, consulté le octobre 19, 2025,
<https://www.autodesk.com/solutions/parametric-modeling>
104. Parametric Design Workflow : r/rhino - Reddit, consulté le octobre 19, 2025,
https://www.reddit.com/r/rhino/comments/15py3uq/parametric_design_workflow/
105. Automating Fusion 360 with the API | Autodesk University, consulté le octobre 19, 2025,
<https://www.autodesk.com/autodesk-university/class/Automating-Fusion-360-API-2019>
106. Creating and manipulating geometry · A FreeCAD manual - Yorik van Havre, consulté le octobre 19, 2025,
https://yorikvanhavre.gitbooks.io/a-freecad-manual/content/python_scripting/creating_and_manipulating_geometry.html
107. 3D design with Python and FreeCAD - Lean2, consulté le octobre 19, 2025,
<https://iosoft.blog/2019/05/22/3d-design-python-freecad/>
108. Geometry Scripting in Blender,Pythonocc and FreeCAD using Python - Medium, consulté le octobre 19, 2025,
<https://medium.com/@topo3dam/geometry-scripting-in-blender-pythonocc-and-freecad-using-python-7d8701f2bcf4>
109. Welcome to the Fusion API (Application Programming Interface), consulté le octobre 19, 2025,
<https://help.autodesk.com/view/fusion360/ENU/?guid=GUID-A92A4B10-3781-4925-94C6-47DA85A4F65A>
110. Parametric Modeling With Fusion 360 API : 7 Steps - Instructables, consulté le octobre 19, 2025,
<https://www.instructables.com/Parametric-Modeling-With-Fusion-360-API/>
111. I wrote a Python / Freecad library to do HUNDREDS of simulations in less than 20 minutes, consulté le octobre 19, 2025,

- <https://www.youtube.com/watch?v=cwtgB4KpdJo>
112. Easy parametric FEA with FreeCAD and Python - YouTube, consulté le octobre 19, 2025, https://www.youtube.com/watch?v=RL_g89hDG9M
 113. AI-Driven Design Tool for Blueprints and 3D Models | by Jung-Hua Liu | Sep, 2025 | Medium, consulté le octobre 19, 2025, <https://medium.com/@gwrx2005/ai-driven-design-tool-for-blueprints-and-3d-models-a9145f5ee537>
 114. Generative AI for CAD Automation: Leveraging Large Language Models for 3D Modelling - arXiv, consulté le octobre 19, 2025, <https://arxiv.org/html/2508.00843v1>
 115. BlenderLLM: A LLM specifically designed to generate CAD scripts based on user instructions. These scripts are then executed in Blender to render 3D models. - GitHub, consulté le octobre 19, 2025, <https://github.com/FreedomIntelligence/BlenderLLM>
 116. Using AI tools to automate Python scripting in FreeCAD for part generation and modification, consulté le octobre 19, 2025, <https://forum.v1e.com/t/using-ai-tools-to-automate-python-scripting-in-freecad-for-part-generation-and-modification/47442>
 117. Do you use any AI tools to help you use FreeCAD? - Reddit, consulté le octobre 19, 2025, https://www.reddit.com/r/FreeCAD/comments/1fpav5i/do_you_use_any_ai_tools_to_help_you_use_freecad/
 118. dataset creation for code LLM : r/LLMDevs - Reddit, consulté le octobre 19, 2025, https://www.reddit.com/r/LLMDevs/comments/175eii2/dataset_creation_for_code_llm/
 119. ChatGPT Makes A 3D Model: The Secret Ingredient? Much Patience - Hackaday, consulté le octobre 19, 2025, <https://hackaday.com/2023/05/04/chatgpt-makes-a-3d-model-the-secret-ingredient-much-patience/>
 120. Generating CAD Code with Vision-Language Models for 3D Designs - arXiv, consulté le octobre 19, 2025, <https://arxiv.org/html/2410.05340v2>
 121. Augmented design automation: leveraging parametric designs using large language models | Proceedings of the Design Society, consulté le octobre 19, 2025, <https://www.cambridge.org/core/journals/proceedings-of-the-design-society/article/augmented-design-automation-leveraging-parametric-designs-using-large-language-models/CFB521C3560A3785CBFB856248BF420A>
 122. cad-coder: an open-source vision-language model for computer-aided design code generation - MIT DeCoDE Lab, consulté le octobre 19, 2025, https://decode.mit.edu/assets/papers/IDETC_CadCode_decodeweb.pdf
 123. System architecture of the LLM-driven FreeCAD script generation... - ResearchGate, consulté le octobre 19, 2025, https://www.researchgate.net/figure/System-architecture-of-the-LLM-driven-FreeCAD-script-generation-framework-The-workflow_fig1_394293021

124. Validation and Verification in Finite Element Analysis (FEA) - Technical Articles, consulté le octobre 19, 2025, <https://control.com/technical-articles/validation-and-verification-in-finite-element-analysis-fea/>
125. How to Verify & Validate Your FEA - FEA Academy, consulté le octobre 19, 2025, <https://www.fea-academy.com/index.php/component/content/article/27-blog/fea-generalities/71-vv>
126. Solved: Generative Design outcomes accuracy - Autodesk Community, consulté le octobre 19, 2025, <https://forums.autodesk.com/t5/fusion-design-validate-document/generative-design-outcomes-accuracy/td-p/8579651>
127. Cost and Value Engineering | aPriori, consulté le octobre 19, 2025, <https://www.apriori.com/solutions/roles/cost-and-value-engineering/>
128. Cost Estimating Software for Manufacturing | Paperless Parts, consulté le octobre 19, 2025, <https://www.paperlessparts.com/pricing-costing-automation/>
129. Manufacturing Estimating and Quoting Software | Costimator® MTI Systems, consulté le octobre 19, 2025, <https://www.mtisystems.com/>
130. Automated Cost Estimation for Manufacturing Companies - Simus Systems, consulté le octobre 19, 2025, <https://www.simus-systems.com/en/applications/calculating-manufacturing-costs/>
131. Machine Learning Approach for Early Assembly Design Cost Estimation: A Case from Make-to-Order Manufacturing Industry, consulté le octobre 19, 2025, <https://ijtech.eng.ui.ac.id/download/article/5675>
132. General Motors | Generative Design in Car Manufacturing - Autodesk, consulté le octobre 19, 2025, <https://www.autodesk.com/customer-stories/general-motors-generative-design>
133. Driving a lighter, more efficient future of automotive part design at GM - Autodesk, consulté le octobre 19, 2025, <https://www.autodesk.com/design-make/articles/automotive-design>
134. General Motors Drives Vehicles of the Future with Generative Design - YouTube, consulté le octobre 19, 2025, <https://www.youtube.com/watch?v=rExftrAo3p4>
135. Generative Design Methodology and Framework Exploiting Designer-Algorithm Synergies, consulté le octobre 19, 2025, <https://www.mdpi.com/2075-5309/12/12/2194>
136. Toward Controllable Generative Design: A Conceptual Design Generation Approach Leveraging the Function-Behavior-Structure Ontology and Large Language Models | J. Mech. Des. | ASME Digital Collection, consulté le octobre 19, 2025, <https://asmedigitalcollection.asme.org/mechanicaldesign/article/146/12/121401/1200469/Toward-Controllable-Generative-Design-A-Conceptual>