Optimisation des outils en forgeage à chaud par simulation éléments finis et méthode inverse : applications à des problèmes industriels

Daniel Vieilledent

▶ **To cite this version:**

Daniel Vieilledent. Optimisation des outils en forgeage à chaud par simulation éléments finis et méthode inverse : applications à des problèmes industriels. Sciences de l'ingénieur [physics]. École Nationale Supérieure des Mines de Paris, 1999. Français. NNT : 1999ENMP0936 . pastel-00001861

HAL Id: pastel-00001861
https://pastel.hal.science/pastel-00001861

Submitted on 24 Aug 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée à

L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS

par

Daniel VIEILLEDENT

en vue de l'obtention du titre de

DOCTEUR

en

SCIENCE ET GÉNIE DES MATÉRIAUX

OPTIMISATION DES OUTILS EN FORGEAGE À CHAUD PAR SIMULATION ÉLÉMENTS FINIS ET MÉTHODE INVERSE. APPLICATIONS À DES PROBLÈMES INDUSTRIELS

Soutenue le 29 Septembre 1999, devant le jury composé de :

Mr.	Jean-Claude GELIN	Président & Rapporteur
Mr.	Mohamed MASMOUDI	Rapporteur
Mr.	Jean-Loup CHENOT	Examinateur
Mr.	Philippe POUBANNE	Examinateur
Mr.	Lionel FOURMENT	Examinateur

Remerciements

Je remercie la direction de l'Ecole des Mines de Paris pour m'avoir permis d'effectuer cette thèse au sein du Centre de Mise en Forme de Matériaux, à Sophia-Antipolis. Je tiens à remercier tout particulièrement M. Jean-Loup CHENOT, Directeur du CEMEF, pour la confiance qu'il m'a accordée, en m'accueillant dans ce laboratoire et en me confiant un sujet de recherche ambitieux dont il fut l'instigateur.

Je remercie vivement M. Jean-Claude GELIN et M. Mohamed MASMOUDI, pour avoir accepté de bien vouloir critiquer ce travail en qualité de rapporteur. Merci à M. Philippe POUBANNE pour avoir accepté de participer à mon jury, et pour l'intérêt qu'il a porté à ce travail, lui donnant ainsi une toute autre dimension.

Ce projet s'inscrit dans le cadre d'un contrat BRITE. Merci à M. Benoît MARTY et M. Patrice LASNE, thanks to Mr. Ian DEMPSTER, gracias a la seniora Maria Teresa SANTOS y a los seniores Jose-Luis ROMERO, Miguel ONGAY, Jesus-Mari OJANGUREN, pour l'intérêt accordé à cette étude et pour leur accueil si chaleureux.

La solidité d'un édifice dépend des fondations. Si ce travail a pu aboutir, c'est en grande partie grâce à la qualité du travail effectué par mon prédécesseur Tudor BALĂN. Merci Tudor.

Ce fut un réel plaisir de travailler trois ans sous la direction de Lionel FOURMENT. Merci Lionel pour tes encouragements et ton soutien dans les moments de doute, pour la pertinence de tes conseils et pour les relectures méticuleuses de ce document.

Je remercie tous ceux qui ont contribué à rendre ces années si agréables. Certains m'ont accompagné au début du chemin, Olof, Marie-Laure, Yannick, David, Alexis, Manu, Roland, et ont suivi d'autres routes par la suite. Ceux que j'ai rencontré au milieu de mon périple, Andy et Christine (coucou Chloé), Etienne (et sa douce), Gaby, Karim, David, Bourrin (et sa famille : les pauvres !), la Doude, et tous ceux qui ont contribué, par des mots et des attentions, à rendre ce voyage plaisant de bout en bout.

Et puis, il y a mes deux co-bureau. Oliv qui a supporté mes groumpheries et mon envahissement territorial, et Lolo (coucou Baptiste), qui en plus des charges précédentes, a subi le flot incessant de mes questions diverses et variées. Qu'ils soient remerciés de leur patience.

Karine, les cadeaux ne valent que par les sacrifices qu'ils nécessitent. Je sais ce qu'il en est, alors, mille fois merci.

Table des matières

Introduction générale 1

1 Le problème d'optimisation en mise en forme: contexte et motivations 3

1.1 Méthodes directes 4

1.1.1 Systèmes experts 4

1.1.2 Méthode du forgeage rétrograde 5

1.1.3 Mise en forme idéale 7

1.2 Méthodes d'optimisation 8

1.2.1 Minimisation sans contrainte 10

1.2.2 Minimisation sous contraintes : méthodes primales 14

1.2.3 Minimisation sous contraintes : méthodes de transformation 17

1.2.4 Algorithmes d'ordre 0 et méthodes probabilistes 18

1.2.5 Méthodes statistiques 21

1.3 Calcul des dérivées 23

1.3.1 Méthodes numériques 23

1.3.2 Calcul analytique des dérivées : méthodes varationnelles 25

1.3.3 Calcul analytique des dérivées : méthodes discrètes 27

1.3.4 Méthodes semi-analytiques 30

1.3.5 Les outils informatiques pour l'analyse de sensibilité 31

2 Simulation du procédé de forgeage 33

2.1 Formulation du problème 33

2.1.1 Les équations du problème continu 33

2.1.2 Formulation faible 35

2.2 Discrétisation 36

2.2.1 Discrétisation spatiale : méthode éléments finis 36

2.2.2 Discrétisation temporelle 38

2.2.3 Traitement des conditions aux limites 38

2.2.4 Résolution du système 39

2.2.5 Aspect instationnaire 40

2.2.6 Les pertes de volume 41

2.3 Lissage des variables nodales et remaillage 43

2.3.1 Lissage des variables nodales 43

2.3.2 Remaillage et transport des variables sur le nouveau maillage 44

2.4 Algorithmes de résolution 47

3 Optimisation de forme en forgeage 49

3.1 Les critères d'optimisation 50

3.1.1 Remplissage de la matrice finale 51
3.1.2 Energie de mise en forme 52
3.1.3 Critères métallurgiques 53
3.1.4 Elimination des replis de matière 57
3.1.5 Cas particulier des défauts de repli : défaut d'aspiration 59
3.2 Définition des paramètres d'optimisation 61
3.3 Algorithmes d'optimisation 65
3.3.1 Cas général 65
3.3.2 Cas particulier du forgeage en matrices fermées 66
4 Analyse de sensibilité 69
4.1 Dérivation des fonctions coût 69
4.1.1 Fonction coût remplissage 70
4.1.2 Fonctions coût liées aux déformations 70
4.1.3 Fonction coût énergie 72
4.1.4 Fonction coût repli 73
4.2 Calcul incrémental des sensibilités 73
4.2.1 Sensibilités des variables principales - intégration temporelle 73
4.2.2 Calage des outils 75
4.2.3 Dérivation du pas de temps 77
4.3 Dérivation de l'équation d'équilibre 78
4.4 Traitement du contact 80
5 Applications 83
5.1 Analyse du coût des calculs et de la précision des dérivées 83
5.1.1 Sensibilités des variables principales - intégration temporelle 83
5.1.2 Validation des dérivées 85
5.2 Optimisation des outils de finition – forgeage d'une roue 87
5.3 Optimisation sous contraintes 90
5.3.1 Optimisation sous contrainte de remplissage 90
5.3.2 Optimisation sous contrainte de volume 92
5.4 Le problème de conception 95
5.5 Forgeage d'un rotor aéronautique : défaut de repli 99
5.6 Elimination d'un défaut d'aspiration 101
6 Conclusion 105
Annexes 107
Bibliographie 115

Introduction générale

Le forgeron frappant le fer rouge à coups de marteau : c'est l'image d'Epinal. Bien que simpliste, elle n'en traduit pas moins un métier où le savoir-faire tient une place prépondérante. Mais les pièces à forger deviennent plus complexes, les exigences de qualité plus strictes, et les temps impartis pour les réaliser plus courts. De plus, le procédé de fabrication doit être mis au point en veillant à diminuer les coûts. Lorsque les essais en grandeur réelle sont inenvisageables, par exemple d'un point de vue financier, le forgeron se doit alors de « réussir du premier coup ».

Pour déterminer le meilleur processus de forgeage, la simulation numérique des procédés de mise en forme constitue un moyen d'investigation privilégié. En donnant accès à des grandeurs difficilement accessibles par l'expérience, elle améliore la compréhension des phénomènes simulés. De plus, elle permet d'analyser un procédé, avant qu'il ne soit réellement mis en œuvre. Les idées nouvelles peuvent ainsi être testées rapidement, sans mobiliser l'outil de production. Mais l'obtention d'une gamme de forgeage satisfaisante par cette méthode est souvent le résultat d'un fastidieux travail d'essais-erreurs, durant lequel, la solution est maintes fois testée et modifiée. Le succès de cette approche repose principalement sur l'expérience de l'utilisateur, et certaines solutions peuvent échapper à son intuition.

Dans ce contexte, être en mesure de déterminer la solution optimale par le calcul, sans intervention de l'utilisateur, revêt une importance capitale. Les outils informatiques d'aide à la conception pour les procédés de mise en forme, constituent ainsi le lien entre l'industrie et la recherche numérique : d'un côté les attentes, de l'autre, les méthodes à mettre au point.

De tels logiciels ne sont devenus envisageables que très récemment. Etant donnée la complexité du problème général de mise en forme : non linéarité des équations, instationnarité des écoulements, etc., les outils numériques permettant la simulation des procédés complexes ne sont en effet apparus qu'au début des années quatre-vingts. Le code de simulation Forge2®, développé au CEMEF, est maintenant utilisé dans de nombreuses forges. Un premier outil de conception optimale pour les procédés de forgeage, ayant pour moteur ce logiciel, a été mis en œuvre. Des résultats encourageants ont été obtenus sur des cas académiques. Mais pour exister vraiment, cet outil se doit d'être capable de répondre aux différents problèmes rencontrés dans l'industrie. C'est ce que l'on se propose de faire dans ce travail.

Les différentes méthodes rencontrées dans le cadre de la conception optimale en mise en forme sont présentées dans le premier chapitre. Les équations régissant le problème de forgeage et les méthodes de résolution, sont rappelées dans le deuxième chapitre. Dans le troisième chapitre, le problème de conception est formulé comme un problème de minimisation, en utilisant une approche de type « méthode inverse ». Cette minimisation est réalisée grâce à une analyse de sensibilité aux paramètres définissant la forme des matrices de forgeage. La méthode retenue pour cette analyse, et les équations résultantes, sont présentées dans le quatrième chapitre. Dans le dernier chapitre, la méthode développée est validée, et confrontée, à plusieurs problèmes industriels, de natures et de complexités différentes.

Chapitre 1

Le problème d'optimisation en mise en forme : Contexte et motivations

La réalisation d'un nouveau produit implique la résolution de problèmes complexes. En premier lieu, les formes et les matériaux de chaque pièce (ou composante) doivent être déterminés en fonction des contraintes géométriques ou mécaniques, auxquelles elles sont soumises. Ce problème de conception résolu, il faut déterminer le procédé de mise en forme capable de produire ces pièces. Résoudre les deux problèmes ainsi posés (conception des pièces et conception de la technologie de fabrication) n'est pas chose aisée, et leur forte interactivité ne simplifie pas la tâche. Dans les deux cas, on peut faire le distinguo entre les problèmes de conception et d'optimisation. Schématiquement, le premier consiste à déterminer une solution à partir de la feuille blanche (et du cahier des charges), alors que le deuxième a pour base de départ une pièce ou un procédé déjà connu et maîtrisé. Il s'agit alors d'en améliorer les performances ou de l'adapter au nouveau problème sans changement fondamental.

Si on regarde les méthodes et les outils qui ont été élaborés pour aider les industriels, on est impressionné par le nombre de travaux dédiés au problème de conception des pièces. Les techniques développées en optimisation structurale ou topologique, ont ainsi donné naissance à des outils efficaces d'aide à la conception ou à l'optimisation. Il est par contre étonnant de remarquer que les méthodes d'aide à la conception des procédés de fabrication sont comparativement récentes et rares. Ceci est en particulier dû à la complexité des problèmes de mise en forme, qui sont le plus souvent instationnaires et qui doivent être traités par une approche grandes déformations. Les travaux de recherche, spécifiques à ce domaine, se sont en effet développés de concert avec les méthodes de simulation numérique des procédés (le plus souvent basées sur une méthode éléments finis).

L'objectif de ce chapitre, est de présenter une vue d'ensemble des principales méthodes numériques développées pour l'optimisation des procédés de mise en forme. Le matériau et la forme de la pièce finale étant fixés, le but est alors de déterminer les éléments définissant la technologie du procédé. Parmi ces éléments (*ou paramètres*), les principaux sont :

les paramètres de procédé : température, vitesse, lubrifiant, etc.

les paramètres géométriques : dimensions du lopin et formes des outils.

Nous nous intéresserons plus particulièrement à l'optimisation de forme (des outils) pour le forgeage. Il est toutefois bon de noter que l'éventail des procédés ainsi abordés est large. Le terme forgeage comprend aussi bien la réalisation de pièces de quelques grammes que de plusieurs tonnes, la fabrication en très grande série que la réalisation unitaire de pièces de sécurité.

Le problème auquel on est confronté, peut finalement se résumer de la manière suivante : connaissant la forme de la pièce à obtenir, il faut déterminer les valeurs optimales des paramètres (géométriques ou du procédé) permettant de forger la pièce finale aux

dimensions requises et sans défaut majeur tel que les replis. Selon les procédés, d'autres *critères d'optimisation* peuvent être considérés :

- réduction du nombre de passes nécessaires.
- amélioration de la qualité métallurgique de la pièce, en lui faisant subir une déformation thermomécanique plus appropriée.
- augmentation de la durée de vie des outils, par exemple en déterminant des chemins de déformation entraînant une usure plus lente des matrices.
- réduction de la matière perdue (bavures ou surépaisseurs éliminées par usinage).
- …

Ce problème d'optimisation apparaît vaste et complexe, et les méthodes rencontrées ne fournissent que des solutions partielles.

La classification de ces méthodes est délicate. Nous avons choisi de présenter au §1.1 les méthodes dont le but est de déterminer directement les valeurs optimales de certains paramètres de forme sans avoir recourt à plusieurs simulations du procédé. A l'inverse, les méthodes d'optimisation itératives, où plusieurs simulations sont nécessaires pour calculer ces mêmes paramètres optimaux, seront abordées §1.2.

1.1 Méthodes directes

D'un point de vue industriel, la pratique courante pour aborder le problème d'optimisation repose sur le savoir-faire du praticien de la mise en forme et l'utilisation de techniques empiriques. Ces dernières sont généralement basées sur l'expérimentation en grandeur réelle ou à échelle réduite grâce à l'utilisation de matériaux modèles (plasticine, plomb, cire, etc.), plus faciles à déformer. Quelques modèles approchés ont tout de même été élaborés [Altan 73, Oh 94], par exemple pour la prédiction des efforts ou du remplissage des matrices de finition [Chamouard 66]. Mais leurs champs d'applications sont restreints et leurs utilisations délicates.

1.1.1 Systèmes experts

L'étude de la faisabilité d'un procédé constitue une étape préliminaire de la conception d'un nouveau produit. Il faut être en mesure de déterminer, par une analyse rapide, des solutions satisfaisantes. Ce travail est confié à des personnes expérimentées. Les systèmes experts s'inscrivent dans cette optique : comment utiliser au mieux l'expérience des forgerons, qui constitue la richesse d'une entreprise. Les informations et l'expérience acquise lors de la conception de gammes déjà réalisées, sont incluses sous forme de bases de données. Grâce à une série de règles, ils permettent alors de définir quelques principes standards de conception pour un nouveau produit [Glynn 95][Choi 96]. Ces méthodes permettent en plus de conserver le savoir-faire. Une des principales limitations réside dans la difficulté de rassembler et de formaliser les connaissances. Pour un même procédé, suivant les préoccupations des concepteurs, les solutions retenues peuvent en effet être différentes. De plus, l'expérience passée est parfois difficile, voire impossible, à transposer aux nouvelles pièces.

Le développement des méthodes d'intelligence artificielle a permis de répondre en partie à certaines incohérences rencontrées dans le passé. Dans l'ensemble, elles reposent sur un calcul de la « distance » entre les problèmes déjà résolus et le problème étudié [Bakshi 94] [Kim 95]. Néanmoins, lorsque la nouvelle pièce est trop éloignée des familles répertoriées dans la base de données, le problème ne peut être résolu.

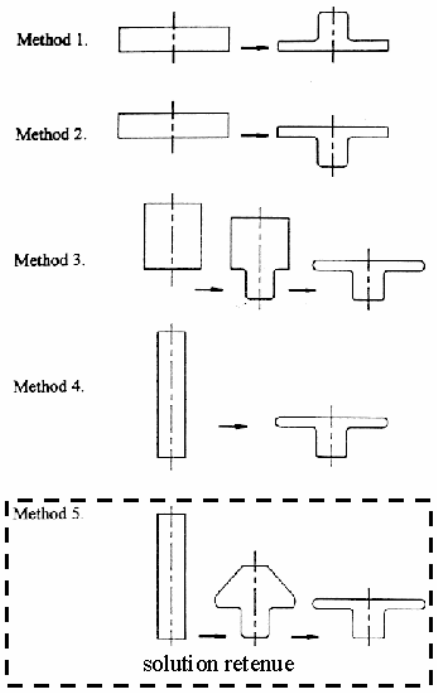


Figure 1.1 : *Sélection d'un procédé par l'utilisation d'un système expert [Hsu 97].*

Les systèmes experts peuvent ainsi se montrer fort utiles, en particulier pour limiter le nombre de solutions à envisager. Toutefois, les solutions retenues doivent ensuite être validées par le calcul ou l'expérience. Il devient ainsi intéressant de coupler l'utilisation de systèmes experts à des techniques d'optimisation [Roy 94] du type des méthodes que nous présenterons au §2.1. C'est par exemple l'approche suivie par Hsu *et al.* [Hsu 97] dans le cadre du forgeage à froid d'un disque. Les auteurs utilisent un système expert pour déterminer le type de gamme de forgeage possible (Figure 1.1). Ils cherchent ensuite la préforme optimale en faisant varier trois paramètres : deux paramètres géométriques de la préforme (Figure 1.2) et la hauteur d'écrasement. Cette phase d'optimisation est réalisée par une méthode statistique utilisant un réseau de neurones. La préforme est testée par une simulation éléments finis, permettant de valider le choix, et aussi d'enrichir la base de données.

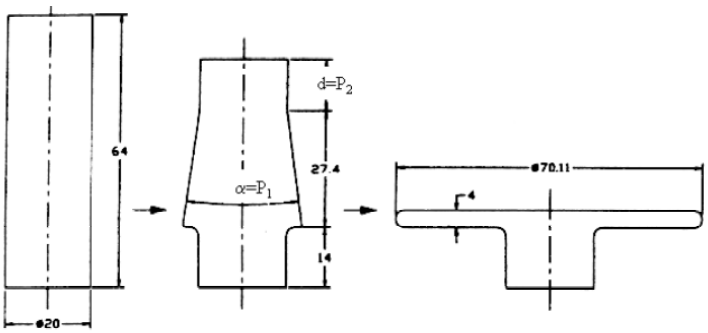


Figure 1.2 : *Optimisation du procédé retenu [Hsu 97].*

1.1.2 Méthode du forgeage rétrograde

Les premiers travaux d'optimisation de forme dédiés aux procédés de mise en forme et utilisant la simulation éléments finis, sont dus à Park *et al.* [Park 83]. La méthode proposée, dite de *forgeage inverse*[1] consiste à « remonter » le temps de manière incrémentale à partir du maillage de la pièce désirée en fin de procédé. Cela revient à déterminer un chemin inverse de

[1] En anglais : « Backward tracing method »

déformation permettant de réaliser la pièce finale. Le problème ainsi posé est complexe. A chaque incrément de temps t, connaissant la configuration Ω_t, il faut déterminer la configuration $\Omega_{t-\Delta t}$ à l'instant $t-\Delta t$ précédent. La méthode de résolution est itérative. La première estimation du maillage $\tilde{\Omega}_{t-\Delta t}$ est réalisée à partir du champ de vitesse de l'instant t :

$$X_k^{t-\Delta t(1)} = X_k^t - V_k^t \Delta t \qquad \forall k \in \tilde{\Omega}_t \tag{1.1}$$

Le champ des vitesses $V^{t-\Delta t(1)}$ est ensuite calculé par résolution du problème éléments finis sur la configuration $\tilde{\Omega}_{t-\Delta t}^{(1)}$ à $t-\Delta t$. La configuration « fictive » $\tilde{\Omega}_t^{(1)}$ est calculée par :

$$X_k^{t(1)} = X_k^{t-\Delta t(1)} + V_k^{t-\Delta t(1)} \Delta t \qquad \forall k \in \tilde{\Omega}_{t-\Delta t}^{(1)} \tag{1.2}$$

Elle est ensuite comparée à Ω_t. Ce schéma est répété jusqu'à ce que l'écart entre Ω_t et $\tilde{\Omega}_t^{(i)}$ soit inférieur à un critère de convergence donné. Cette méthode permet donc, en remontant le temps, d'atteindre le début du procédé et d'obtenir la géométrie de la préforme.

Le forgeage inverse présente certains inconvénients. Tout d'abord, il fournit la préforme elle même et non pas l'outil de préforme permettant de l'obtenir. C'est ensuite à l'utilisateur de déterminer une gamme de forgeage capable de produire une telle préforme (sans garantie d'existence). Cette tâche peut s'avérer d'autant plus difficile que la géométrie de la préforme est complexe. De plus, cette méthode semble difficilement applicable à des matériaux dont le comportement dépend de l'histoire de la déformation. Mais la principale limitation provient de la nécessité de connaître les conditions aux limites en contact au début de chaque incrément. Lors de la simulation directe, l'arrivée des nœuds en contact est un processus irréversible. Déterminer les nœuds en contact à l'instant $t-\Delta t$ est impossible à partir de la seule connaissance des nœuds en contact à l'instant t. Il faut alors définir des critères de « décollement » des nœuds en contact.

Plusieurs méthodes ont été proposées. Le schéma de relâchement des nœuds peut être défini en se basant sur la forme des outils. Dans le cas du forgeage d'un disque axisymétrique, Hwang *et al.* [Hwang 86] ont proposé de relâcher en premier les nœuds (un par incrément) se trouvant à la périphérie, puis progressivement, ceux vers le centre de la pièce. Ce critère étant limité à des géométries simples, il a été généralisé par la suite, en introduisant des facteurs de complexité des outils [Zhao 95] et appliqué au forgeage d'un rotor (Figure 1.3). Le problème est ici de déterminer une préforme permettant le remplissage de la matrice finale.

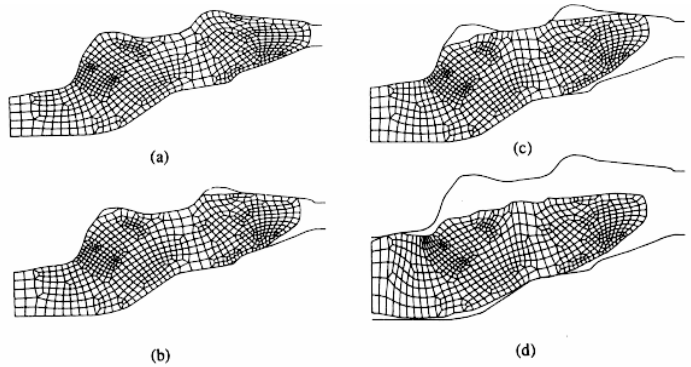


Figure 1.3 : *Forgeage inverse d'un rotor. (a) pièce finale, (b-c) instants intermédiaires de forgeage, (d) préforme [Zhao 95].*

Plusieurs auteurs ont proposé des critères différents : le nœud caractérisé par la plus faible déformation [Hwang 86], la plus faible vitesse de déformation [Han 93] ou par la plus forte contrainte normale [Srinavasan 94]. Biglari *et al.* [Biglari 98] ont utilisé les opérateurs de logique floue afin de définir, pour chaque nœud, une hiérarchie entre ces différents critères. La méthode reste dépendante du maillage, car on ne relâche qu'un nœud par incrément, ce qui rend le temps du procédé, et donc la course des outils, fonction du nombre de nœuds en contact.

Dans tous les cas, le choix d'un critère est dicté par le problème à résoudre. La généralisation est difficile. De plus, la préforme obtenue par cette méthode dépend des hypothèses faites sur l'évolution du contact. Elle n'est donc ni unique ni optimale. Elle permet toutefois de suggérer des types de préformes, qui seront étudiées et optimisées à l'aide de simulations directes.

1.1.3 Mise en forme idéale

Cette méthode a pour objet de concevoir un procédé permettant d'obtenir la pièce finale sans endommagement critique du matériau. Le principe de la mise en forme idéale est qu'un procédé est optimal si chaque élément de matière suit un chemin de déformation à travail minimal. Elle est basée sur les hypothèses suivantes :

- plus la matière est déformée, plus le risque d'endommagement est élevé.
- la déformation minimale est caractérisée par le chemin à travail minimal.

Notons que la première hypothèse n'est pas vérifiée pour les matériaux présentant un adoucissement à la déformation.

Dans le cas de matériaux viscoplastiques ayant un critère de plasticité convexe et régulier, un chemin de déformation est alors optimal si les axes principaux de déformation restent fixes par rapport au matériau et si le rapport des vitesses de déformation principale est constant [Hill 86]. Lorsque le critère de plasticité présente des points anguleux (par exemple le critère de Tresca), le chemin à travail minimal n'est plus unique. Chung et Richmond [Chung 92a] proposent ainsi de définir le procédé optimal, parmi tous les procédés à travail minimal, comme celui qui produit la répartition de déformation la plus homogène dans le matériau. Elle permet alors, connaissant la forme de la pièce finale et l'état initial des déformations, de déterminer le chemin de déformation idéal et les conditions aux limites correspondantes.

Cette méthode présente certaines limitations. Weinberger [Weinberger 97] a ainsi montré, sous certaines restrictions, qu'il existe, dans le cas d'un procédé d'extrusion, plusieurs rapports entre la section d'entrée et la section de sortie pour lesquels il n'existe pas de chemin idéal. De plus, Chung et Richmond [Chung 92b] remarquent que dans le cas général, le chemin de déformation idéal ne peut être suivi par tous les points du matériau sous peine de violer la condition de continuité du matériau. Néanmoins, ce problème n'apparaît pas lorsque le matériau se présente sous forme de membrane. Ceci explique que cette méthode soit principalement utilisée dans le cadre des procédés de mise en forme de plaques [Barlat 94], où l'objectif principal est de déterminer un chemin de déformation ne dépassant pas les limites de formabilité du matériau. Les préoccupations qui ont guidé cette méthode semblent ainsi très éloignées de celles rencontrées en forgeage à chaud. L'application de celle-ci aux

procédés qui nous préoccupent semble discutable.

Il existe d'autres travaux utilisant une approche directe pour déterminer les valeurs optimales de certains paramètres. On peut citer par exemple les méthodes d'optimisation de forme de filières développées par Legat *et al.* [Legat 93] pour les procédés d'extrusion de polymères, ou Srinavasan *et al.* [Srinavasan 90] pour les métaux.

Dans le cadre du forgeage, Berg *et al.* [Berg 95] ainsi que Grandhi *et al.* [Grandhi 93] ont proposé une méthode d'optimisation incrémentale afin de déterminer à chaque pas de temps, la vitesse optimale des outils permettant d'obtenir une vitesse de déformation maximale sur la pièce. Ces méthodes d'optimisation « pas à pas » restent toutefois limitées à des critères d'optimisation qui doivent être vérifiés à chaque instant, et ne concernent que des paramètres dont on cherche l'évolution optimale dans le temps.

1.2 Méthodes d'optimisation

Historiquement, les travaux fondateurs de l'optimisation de forme ont été effectués dans le domaine de la mécanique des structures [Francavilla 75, Bhavikatti 77]. Ces méthodes d'optimisation reposent sur l'idée suivante : transformer le problème de conception optimale en un problème de minimisation. Le procédé est défini en indiquant :

- le matériau
- la géométrie des outillages, du lopin initial
- l'ensemble des paramètres du procédé : température initiale du lopin et des matrices, lubrifiants, pilotage des outils, etc.

La simulation numérique permet alors de procéder à une analyse qualitative : déformations, écoulement, forces nécessaires, apparition de défauts, usure des outils, etc.
A partir de ces résultats, on définit des critères d'optimisation. Si le procédé n'est pas satisfaisant (si l'un des critères d'optimisation n'est pas respecté), on modifie alors le procédé en changeant les valeurs de certains paramètres d'entrée. Pour chaque critère c retenu, la mesure de la « non-qualité » du procédé est obtenue par son expression mathématique Φ_c (fonction coût). Le vecteur optimal des paramètres p est alors solution du problème :

$$\begin{cases} Min\,\Phi(p) \\ c_i(p) \le 0 \quad \forall i = 1,..,m_i \\ h_i(p) = 0 \quad \forall i = 1,..,m_e \\ p \in \mathbb{R}^n \end{cases} \tag{1.3}$$

Dans (1.3), la fonction Φ peut désigner une combinaison de plusieurs critères. Les expressions c_i et h_i désignent les contraintes auxquelles est soumis le vecteur p des paramètres. Elles peuvent être soit de type géométrique, imposant des limites sur les variations des paramètres, soit représenter un critère d'optimisation. C'est en particulier le cas lorsque la vérification d'un critère équivaut à la nullité de sa fonction coût (contrainte d'égalité). Les fonctions Φ, c_i et h_i sont supposées continûment différentiables.

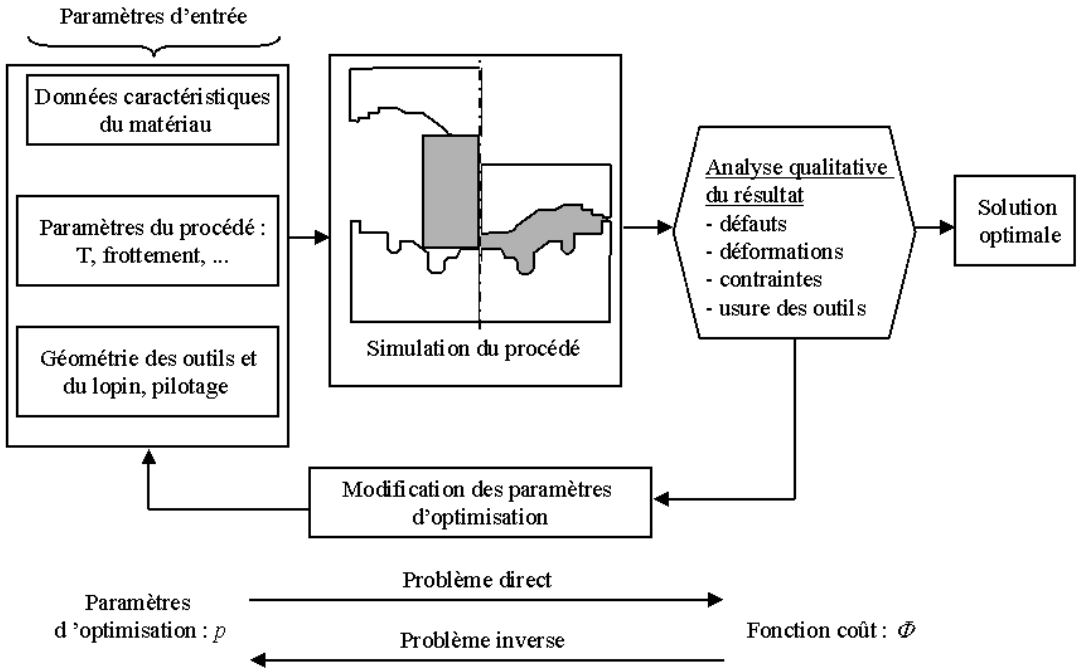


Figure 1.4 : *Problème inverse associé au problème d'optimisation.*

La simulation numérique du procédé de mise en forme apparaît comme le point névralgique des méthodes inverses. Le succès de ces méthodes repose en effet sur une estimation fiable et précise du problème direct. Ceci explique leurs apparitions tardives et leurs développements avec l'avènement des méthodes éléments finis. En dehors de l'optimisation de forme, ces méthodes ont connu un large succès en identification de paramètres [Chenot 93b].

Nous rappelons les conditions nécessaires d'optimalité locale de Khun-Tucker [Khun 51] :

« Une condition nécessaire pour que p^* soit un optimum local de (1.3) est qu'il existe $\lambda \in \mathbb{R}^{m_i}$ et $\mu \in \mathbb{R}^{m_e}$ (multiplicateurs de Khun-Tucker), tels que : »

$$\begin{cases} \nabla \Phi(p^*) + \sum_{i=1}^{m_i} \lambda_i \nabla c_i(p^*) + \sum_{i=1}^{m_e} \mu_i \nabla h_i(p^*) = 0 \\ \lambda_i c_i(p^*) = 0 \quad i = 1,..,m_i \\ \lambda_i \geq 0 \quad \text{et} \quad c_i(p*) \leq 0 \quad i = 1,..,m_i \\ h_i(p*) = 0 \quad i = 1,..,m_e \end{cases} \tag{1.4}$$

Il existe en pratique certaines conditions restrictives à l'application de (1.4) liées à la notion de régularité. Sans entrer dans les détails, notons juste que ces conditions sont vérifiées lorsque les gradients des contraintes forment m_i+m_e vecteurs linéairement indépendants [Schittkowski 85]. L'espace des paramètres p apparaît comme l'espace des variables primales, tandis que les multiplicateurs opèrent sur l'espace dual. Il est possible de classer les méthodes de résolution du problème (1.3) en deux catégories :

- *Les méthodes directes* (ou primales) qui consistent à résoudre directement (1.3) en

se ramenant à une suite de sous-problèmes approchés. On détermine ainsi une suite de p^k vérifiant les contraintes et assurant une décroissance de Φ.

- *Les méthodes de transformation* (ou duales) qui consistent à remplacer le problème (1.3) par une suite de sous-problèmes sans contrainte. Contrairement aux précédentes, les contraintes ne sont vérifiées qu'à la convergence de la suite p^k.

Nous allons examiner dans un premier temps les méthodes qui nécessitent la connaissance des dérivées (d'ordre 1 ou 2) des fonctions et des contraintes, en commençant par les algorithmes classiques de minimisation sans contrainte, certaines de ces techniques servant de base aux algorithmes de minimisation du problème général (1.3). Nous examinerons ensuite quelles sont les méthodes pour lesquelles aucun calcul de dérivée n'est nécessaire.

1.2.1 Minimisation sans contrainte

On se propose ici de résoudre le problème :

(1.5) $$\begin{cases} Min\,\Phi(p) \\ p \in \mathbb{R}^n \end{cases}$$

Le déroulement d'un algorithme itératif d'optimisation sans contrainte peut se résumer, pour chaque itération k, de la manière suivante :

a) recherche d'une direction $d_k \in 3^n$.

b) calcul du déplacement λ_k à effectuer dans la direction d_k, pour minimiser $\Phi(p_k + \lambda_k d_k)$.

c) actualisation pour l'itération suivante : $p_{k+1} = p_k + \lambda_k d_k$.

L'étape a) consiste à déterminer une direction de descente, c'est à dire vérifiant :

(1.6) $$d_k . \nabla \Phi(p_k) < 0$$

L'étape b) constitue un problème d'optimisation unidimensionnel sans contrainte dont les méthodes usuelles de résolution seront résumées au § 1.2.1.2.

(1.7) $$\begin{cases} Min\, f(\lambda) \\ \lambda \in \mathbb{R}^+ \end{cases} \quad \text{avec} \quad f(\lambda) = \Phi(p_k + \lambda d_k)$$

1.2.1.1 *Calcul des directions de descente.*

Méthode du gradient : la direction normalisée d de plus forte décroissance de Φ au voisinage d'un point p est donnée par :

(1.8) $$d = -\frac{\nabla \Phi(p)}{\|\nabla \Phi(p)\|}$$

La convergence peut être très lente. L'utilisation d'une méthode de recherche linéaire exacte

(1.7) génère des directions successives orthogonales entre elles. On observe alors une progression en zig-zag qui peut être d'autant plus lente que l'on est proche de l'optimum. Toutefois, Jensen *et al.* [Jensen 98] s'en servent pour la minimisation d'usure en emboutissage. Il existe des techniques d'accélération [Minoux 83] afin de palier à cet inconvénient. Elles restent néanmoins fort coûteuses et sont peu utilisées.

Des méthodes plus performantes ont donc été développées. On peut citer par exemple l'algorithme du gradient conjugué [Morris 82], proposé à l'origine pour la minimisation de fonctions quadratiques, puis étendu à des fonctions quelconques par Fletcher et Reeves [Fletcher 64]. Néanmoins, cette méthode cumule les erreurs d'arrondi, la convergence n'est alors assurée que si l'on procède à une réinitialisation périodique. Certaines techniques de redémarrage ont été proposées (méthode de Beales [Powell 75] par exemple).
Zabaras *et al.* [Zabaras 95] ont utilisé cette méthode pour l'optimisation de flux de chaleur en solidification. Néanmoins, il n'existe que peu d'applications de cet algorithme en mise en forme.

Méthode de Newton (méthode d'ordre 2) : considérons l'approximation quadratique q de Φ au voisinage d'un point p_k :

$$q(p) = \Phi(p_k) + \nabla\Phi(p_k).(p - p_k) + \frac{1}{2}(p - p_k).\nabla^2\Phi(p_k)(p - p_k) \tag{1.9}$$

On choisit alors p_{k+1} qui minimise q (l'optimum existe et est unique si $\nabla^2\Phi(p_k)$ est défini positif) et qui est défini par $\nabla q(p_{k+1})=0$. Ce qui conduit au système linéaire :

$$(p_{k+1} - p_k) = -\left(\nabla^2\Phi(p_k)\right)^{-1}\nabla\Phi(p_k) \tag{1.10}$$

Cette méthode est reconnue pour être une des plus efficaces [Bonnans 85]. Mais le principal inconvénient réside dans le calcul des dérivées secondes de Φ qui s'avère le plus souvent coûteux et très difficile à réaliser. Un certain nombre d'algorithmes se proposent ainsi de contourner cette difficulté en utilisant des approximations de la matrice hessienne. On peut mentionner le cas particulier où Φ peut s'écrire sous forme de moindres carrés. On obtient alors une approximation du hessien en ne considérant que les produits des gradients. Cette approximation, qui est à la base des algorithmes de Gauss-Newton ou Levenber-Marquardt [Minkowycz 88], est largement utilisée en identification de paramètres rhéologiques [Gavrus96] [Ghouati 98a].

Méthodes de quasi-Newton (ou méthodes à métrique variable): il s'agit de construire itérativement une matrice H, définie positive, qui approche le hessien de Φ (ou son inverse). La correction qui permet d'obtenir H_k à partir de H_{k-1} est de la forme :

$$H_k = H_{k-1} + \delta_{k-1} \tag{1.11}$$

Différentes formules de correction (1.11) faisant intervenir H_{k-1}, $\nabla\Phi(p_k)$, $\nabla\Phi(p_{k-1})$, p_k et p_{k-1} ont été proposées (on peut citer par exemple la méthode DFP [Fletcher 63]). L'une des plus

efficaces sur de nombreux problèmes est l'algorithme BFGS [Broyden 70]. L'inverse du hessien de Φ est actualisé à chaque itération k à partir des valeurs connues H_{k-1}, $\nabla\Phi(p_k)$ et p_k :

$$H_k = H_{k-1} + \left[I + \frac{\gamma_k \otimes (H_{k-1}\delta_k)}{\delta_k . \gamma_k} \right] \frac{\delta_k \otimes \delta_k}{\delta_k . \gamma_k} - \frac{\delta_k \otimes (H_{k-1}\gamma_k) + (H_{k-1}\gamma_k) \otimes \delta_k}{\delta_k . \gamma_k} \quad (1.12)$$

avec :

$$\delta_k = p_k - p_{k-1} \quad (1.13)$$

$$\gamma_k = \nabla\Phi(p_k) - \nabla\Phi(p_{k-1}) \quad (1.14)$$

La direction de descente est déterminée par :

$$d_k = -H_k \nabla\Phi(p_k) \quad (1.15)$$

Le nouveau point p_{k+1}, est finalement obtenu après une étape de minimisation unidimensionnelle : $p_{k+1} = p_k + \lambda d_k$.

Si H_k est définie positive, alors la direction d_k est assurée d'être une direction de descente (condition (1.6)). Si H_0 est définie positive (le plus souvent $H_0=I$), la positivité est assurée à chaque itération si la condition (1.16) est vérifiée.

$$\delta_k . \gamma_k < 0 \quad (1.16)$$

Appliqué à une forme quadratique, cet algorithme permet d'obtenir l'inverse du hessien de la fonction en, au plus, n itérations. De plus il est peu sensible aux imprécisions de la procédure de recherche linéaire. Ceci permet de s'affranchir d'une recherche linéaire exacte dont la résolution est souvent aussi coûteuse que celle du problème initial. Néanmoins, la convergence globale n'est assurée que si l'algorithme est périodiquement réinitialisé. Pour un problème d'extrusion, Kusiak *et al.* [Kusiak 89] ont ainsi mis en évidence l'efficacité des méthodes de quasi-Newton en comparant une méthode DFP, un algorithme du gradient et une méthode d'ordre 0. De par sa très large diffusion [Noiret 96, Bourdin 98, Vieilledent 98a, Zhao 97a], la méthode BFGS semble la plus adaptée aux problèmes d'optimisation en mise en forme.

1.2.1.2 *Techniques de recherche linéaire*

Comme nous venons de le voir, la plupart des algorithmes nécessitent la résolution du problème intermédiaire (1.7). Il existe de nombreuses méthodes de recherche linéaire pour le résoudre. Cette étape constitue souvent la partie la plus coûteuse en temps de calcul, car il implique une évaluation de la fonction Φ à chaque itération. Les méthodes de Newton ou de la sécante, ne sont que rarement utilisées. Les méthodes les plus répandues sont la méthode de la section d'or [Morris 82] [Chung 98], la méthode de Brent [Chen 95] et l'interpolation polynomiale. Nous nous intéresserons ici à cette dernière, et plus particulièrement à la méthode d'interpolation parabolique et de Davidon.

Interpolation parabolique [Press 92]: à partir de trois points λ_1, λ_2 et λ_3 vérifiant :

$$0 < \lambda_1 < \lambda_2 < \lambda_3 \quad \text{et} \quad f(\lambda_1) > f(\lambda_2) \text{ et } f(\lambda_3) > f(\lambda_2) \tag{1.17}$$

la méthode de l'interpolation parabolique consiste à choisir le quatrième point λ_4 comme le minimum du polynôme de degré 2 qui passe par les points (λ_i, f(λ_i)) i=1, ..3. Les trois points qui « encadrent » (Figure 1.5) le minimum parmi les quatre disponibles sont conservés et le processus est recommencé. Cette méthode, qui a une vitesse de convergence super linéaire, nécessite toutefois deux évaluations supplémentaires avant d'être effective (*f(0)* étant connu initialement). Lorsque les valeurs des dérivées sont connues, il est alors possible d'utiliser une interpolation cubique à deux points ou méthode de Davidon.

Méthode de Davidon [Schittkowski 80] *:* à partir de deux points λ_1 et λ_2, on génère un nouveau point en considérant le minimum de la cubique P_3 qui passe par les points (λ_1, f(λ_1)) et (λ_2, f(λ_2)) et qui a pour pente en ces points les valeurs des dérivées de *f* en λ_1 et λ_2 (Figure 1.5). Ce minimum est obtenu en annulant la dérivée de P_3. Cette méthode possède aussi une vitesse de convergence super linéaire.

Remarquons que d_k étant une direction de descente, la dérivée de *f* en 0 est de signe négatif :

$$\frac{df}{d\lambda}(0) = d_k . \nabla \Phi(p_k) < 0 \tag{1.18}$$

ce qui assure la décroissance de *f* à l'origine. Pour les deux méthodes, le point initial λ_1 est défini par $\lambda_1 = 0$. Le second point λ_2 est généralement obtenu en effectuant un déplacement arbitraire : $\lambda_2 = \beta$. Deux cas peuvent se présenter :

- $f(\lambda_2) < f(\lambda_1)$: on pourra alors effectuer successivement le calcul de $f(\lambda_i)$ avec

$$\lambda_i = (i - 1)\beta \quad \forall i > 2 \tag{1.19}$$

  jusqu'à ce que $f(\lambda_i) < f(\lambda_{i+1})$ et que la condition (1.17) soit vérifiée.
- $f(\lambda_2) > f(\lambda_1)$: dans le cas de l'interpolation parabolique, il faut alors déterminer un nouveau point $\lambda_3 \in [\lambda_1, \lambda_2]$, par exemple en choisissant le point milieu, ou si les dérivées sont disponibles, en appliquant une itération de la méthode de Davidon.

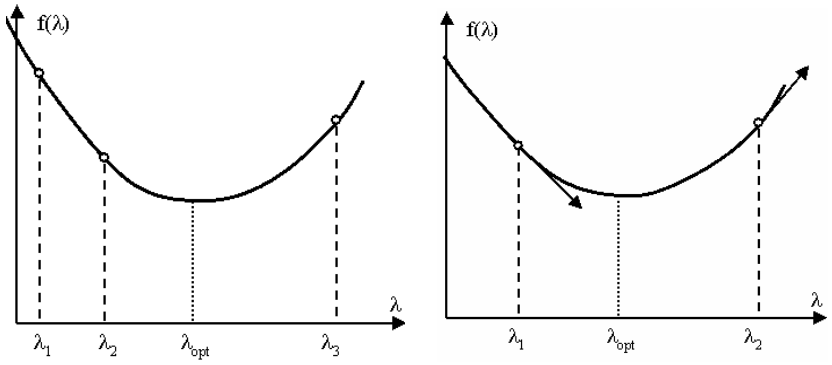


Figure 1.5 : *Interpolation quadratique et méthode de Davidon.*

La méthode de Davidon permet donc d'économiser un calcul de fonction par rapport à une méthode quadratique.

Critères d'arrêt : Les méthodes décrites visent à obtenir une approximation plus ou moins précise de l'optimum (qui peut n'être que local). Si on fait une recherche linéaire exacte, la procédure est arrêtée lorsque l'intervalle d'encadrement de la solution est d'une taille inférieure à une valeur prescrite. En fait, la convergence de certains algorithmes (en particulier l'algorithme BFGS, cf. 1.2.1.1) n'est pas assujettie à une recherche linéaire exacte. Plusieurs critères d'arrêt basés sur des conditions moins restrictives ont été proposés : règle de Goldstein ou d'Armijo [Armijo 66], lorsque le gradient de la fonction est trop coûteux à calculer à chaque itération, ou bien règle de Wolfe [Wolfe 71]. Ce dernier propose d'arrêter la recherche lorsque les deux conditions suivantes sont satisfaites :

$$f(\lambda) \leq f(0) + m_1 \lambda \frac{df}{d\lambda}(0) = f(0) + m_1 \lambda d_k . \nabla \Phi(p_k) \tag{1.20}$$

$$\frac{df}{d\lambda}(\lambda) \geq m_2 \frac{df}{d\lambda}(0) \tag{1.21}$$

où les coefficients m_1 et m_2 vérifient : $0 < m_1 < m_2 < 1$. La règle de Wolfe est bien adaptée à l'algorithme BFGS car (1.21) permet de vérifier la condition (1.16) et ainsi d'assurer la définie positivité des matrices successives de quasi-Newton.

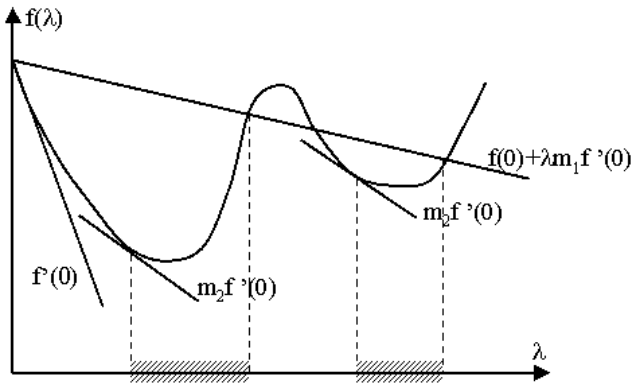


Figure 1.6 : *Intervalles satisfaisant la règle de Wolfe.*

1.2.2 Minimisation sous contraintes : Méthodes primales

Dans le cas des contraintes non linéaires d'inégalité, l'approche la plus naturelle est la méthode des directions réalisables proposée par Zoutendijk [Zoutendijk 60]. A partir d'un point satisfaisant les contraintes (ceci constitue en soit une hypothèse souvent difficile à vérifier), on recherche une direction de déplacement *d*, telle qu'un petit déplacement dans cette direction génère un nouveau point respectant encore les contraintes et diminuant Φ. Le choix le plus courant pour *d* est obtenu à partir d'une double linéarisation (fonction et contraintes saturées au point), se ramenant ainsi à un problème de programmation linéaire [Morris 82]. La convergence de la méthode n'est pas garantie, et la mise en œuvre souvent

difficile. Plusieurs méthodes plus performantes ont été proposées, dont les principales sont rappelées dans [Gill 81]. Notons que les méthodes primales sont souvent délicates à mettre en œuvre et que le choix de telle ou telle méthode est souvent dicté par le type de problème à résoudre. Certains auteurs proposent ainsi d'utiliser successivement plusieurs algorithmes différents suivant des séquences prédéfinies [Patnaik 97].

Nous nous intéresserons ici plus particulièrement à deux méthodes parmi les plus connues : la méthode du gradient réduit généralisé et la méthode de programmation quadratique.

Méthode du gradient réduit généralisé : On suppose que le problème (1.3) n'est soumis qu'à des contraintes d'égalité. On notera le vecteur h des contraintes par :

$$h(p)=\left[h_1(p),\ldots,h_{m_e}(p)\right]^T \tag{1.22}$$

On supposera que l'on dispose d'un point p_k vérifiant $h(p_k)=0$, et que $\{\nabla h_i(p_k),\ i=1,\ldots,\ m_e\}$ est une famille de vecteurs linéairement indépendants. Il existe alors une matrice $\nabla_B h$, inversible de dimension $m_e x m_e$, et une matrice $\nabla_N h$ $m_e x(n-m_e)$, telles que :

$$\nabla h(p_k)=\begin{bmatrix}\nabla h_1^T(p_k)\\ \vdots \\ \nabla h_{m_e}^T(p_k)\end{bmatrix}=\left[\nabla_B h,\nabla_N h\right] \tag{1.23}$$

Le vecteur p peut s'écrire :

$$p^T=[p_1,\ldots,p_{m_e},p_{m_e+1},\ldots,p_n]^T=[p_B,p_N]^T \tag{1.24}$$

faisant ainsi apparaître les variables dépendantes p_B (variables de base) et les variables indépendantes p_N. On utilise alors le théorème des fonctions implicites [Bonnans 85] pour se ramener à un problème à $n-m_e$ inconnues, en écrivant

$$p_N(p_B) \quad \text{tel que} \quad h(p_N,p_B)=0 \tag{1.25}$$

La projection du gradient de Φ sur l'espace des variables indépendantes (appelé gradient réduit généralisé) est donnée par :

$$\nabla_N\Phi(p_N)=\frac{\partial\Phi}{\partial p_N}(p_N,p_B(p_N))+\left(\frac{\partial p_B}{\partial p_N}\right)^T\frac{\partial\Phi}{\partial p_B}(p_N,p_B(p_N)) \tag{1.26}$$

Le vecteur $\partial p_B/\partial p_N$ est obtenu en dérivant la contrainte :

$$\left(\frac{\partial p_B}{\partial p_N}\right)^T=-\left(\nabla_B h\right)^{-1}\nabla_N h \tag{1.27}$$

où $\nabla_N h$ désigne la matrice $(n-m_e)xm_e$ complémentaire de ∇h. On obtient finalement la projection de $\nabla \Phi$ sous la forme :

$$\nabla_N \Phi(p_N) = \nabla_N \Phi(p) - (\nabla_B h)^{-1} \nabla_N h \ \nabla_B \Phi(p) \tag{1.28}$$

Ce gradient permet de calculer une direction de descente d_N par une des différentes méthodes de minimisation sans contrainte présentées au §1.2.1.1 en raisonnant sur les variables indépendantes. On a alors les itérations suivantes :

$$p_N^{k+1} = p_N^k + \lambda\, d_N \tag{1.29}$$

Le problème est de déterminer p_B^{k+1}. On a une prédiction au premier ordre donnée par :

$$\hat{p}_B^{k+1} = p_B^k - (\nabla_B h)^{-1} \nabla_N h\, d_N \tag{1.30}$$

A cause de la courbure des contraintes, $h(p_N^{k+1}, \hat{p}_B^{k+1})$ est en général non nul. Il faut alors résoudre les m_e équations non linéaires à m_e inconnues :

$$h_i(p_N^{k+1}, y_B) = 0 \quad y_B \in \mathbb{R}^{m_e} \tag{1.31}$$

Cet algorithme est considéré comme un des plus performants lorsqu'il est utilisé en combinaison avec une méthode de quasi-Newton de type BFGS [Abadie 79]. Néanmoins, l'étape (1.31) dite de « restauration de faisabilité » constitue la principale difficulté de cette méthode. Nous verrons toutefois au §3.3.2, que cette méthode revêt un intérêt particulier dans le cas de l'optimisation des procédés de forgeage en matrices fermées.

Méthode de programmation quadratique : l'idée de départ est de résoudre directement le système (1.4). Cette méthode appartient à la catégorie des méthodes primales-duales car elles opèrent sur l'espace dual des multiplicateurs et l'espace des variables primales. On supposera que seules les contraintes d'égalité sont considérées. Sous ces hypothèses, et en considérant la fonction de Lagrange L :

$$L(p,\lambda) = \Phi(p) + \sum_{i=1}^{m_e} \lambda_i h_i(p) \tag{1.32}$$

le problème (1.4) revient à résoudre le système à $n+ m_e$ inconnues (p,λ) et à $n+ m_e$ équations :

$$\nabla \Phi(p) + \sum_{i=1}^{m_e} \lambda_i \nabla h_i(p) = 0 \tag{1.33}$$

$$h_i(p) = 0 \quad i=1,\ldots,m_e \tag{1.34}$$

Si on utilise une méthode de Newton pour déterminer une meilleure approximation (p_{k+1}, λ_{k+1}) à l'itération suivante, le système linéaire que l'on doit alors résoudre est alors donné par :

$$\begin{bmatrix} \nabla_p^2 L(p_k,\lambda_k) & \nabla h^T(p_k) \\ \nabla h(p_k) & 0 \end{bmatrix} \begin{bmatrix} p_{k+1}-p_k \\ \lambda_{k+1}-\lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla_p L(p_k,\lambda_k) \\ -h(p_k) \end{bmatrix} \tag{1.35}$$

Le principe de la méthode de programmation quadratique est de remarquer que le vecteur correction $d_k=p_{k+1}-p_k$, solution de (1.35), est aussi solution du problème quadratique sous contraintes linéaires (1.36) [Powell 78]

$$\begin{cases} \underset{d\in\mathbb{R}^n}{Min}\left(\frac{1}{2}d.\nabla_p^2 L(p_k,\lambda_k)d + d.\nabla\Phi(p_k)\right) \\ \nabla h(p_k)d + h(p_k) = 0 \end{cases} \tag{1.36}$$

La résolution de (1.36) peut être réalisée, par exemple, par une méthode de gradient réduit. Dans le cas des contraintes d'inégalités, on montre [Powell 78] qu'il suffit de considérer dans (1.36) une linéarisation des contraintes de la forme :

$$\nabla c(p_k)d + c(p_k) \le 0 \tag{1.37}$$

Les méthodes quasi-Newtoniennes sont généralement utilisées pour construire une approximation de la matrice hessienne de L intervenant dans (1.36). Cet algorithme possède alors une vitesse de convergence super-linéaire au voisinage de la solution. Un certain nombre d'améliorations ont été proposées [Gao 97], en particulier en ce qui concerne les méthodes de recherche linéaire [Bonnans 85].

Cette méthode semble largement diffusée dans le domaine des procédés de mise en forme. On la trouve en effet aussi bien appliquée aux problèmes d'optimisation en extrusion [Ulysse 99, Mihelič 98], en fonderie [Ebrahimi 97], en mise en forme de plaques [Ghouati 99], ou encore en forgeage [Chung 98].

1.2.3 Minimisation sous contraintes : Méthodes de transformation

Les méthodes de transformation s'appliquent préférentiellement dans le cas des contraintes d'inégalité. Les plus connues d'entre elles sont :

- les méthodes de pénalisation (intérieure ou extérieure).
- les méthodes basées sur l'utilisation des multiplicateurs de Lagrange.
- les méthodes d'Usawa.

Ces méthodes présentent deux avantages sur les méthodes primales : elles sont souvent de mise en œuvre plus aisée, et leur domaine d'application est, généralement, plus étendu. Elles présentent de plus, la propriété de ne pas nécessiter un point initial admissible (vérifiant les contraintes). Nous ne présenterons ici que la méthode du Lagrangien augmenté, qui combine l'utilisation des multiplicateurs de Lagrange avec les fonctions de pénalisation. Le but est d'obtenir un problème sans contrainte, qui ne soit pas mal conditionné, comme dans le cas d'une méthode de pénalisation (lorsque le coefficient de pénalisation tend vers $+\infty$). Pour le problème général (1.3), la forme la plus courante de la fonction du Lagrangien augmenté est donnée par [Haftka 92]:

$$L(p,r,\lambda,\mu) = \Phi(p) + \sum_{i=1}^{m_e}\left(rh_i^2(p) + \lambda_i h_i(p)\right) + \sum_{i=1}^{m_i} l_i(p,\mu,r) \tag{1.38}$$

avec

$$l_i(p,r,\mu) = \begin{cases} \mu_i c_i(p) + rc_i^2(p) & si \quad c_i(p) \geq -\dfrac{\mu_i}{2r} \\ -\dfrac{\mu_i^2}{4r} & sinon \end{cases} \tag{1.39}$$

La minimisation de (1.38) par rapport à p est généralement réalisée grâce à une méthode de type gradient, décrite au §1.2.1. Le déroulement des étapes est alors le suivant : on se place à une itération k, pour $\lambda^{(k)}$, $\mu^{(k)}$ et $r^{(k)}$ fixés. La minimisation sans contrainte fournit une nouvelle approximation p_{k+1}. Les multiplicateurs de Lagrange, ainsi que le coefficient de pénalisation sont actualisés par :

$$\begin{cases} \lambda_i^{(k+1)} = \lambda_i^{(k)} + 2r^{(k)} h_i(p_{k+1}) \quad i = 1,\ldots,m_e \\ \mu_i^{(k+1)} = \mu_i^{(k)} + 2r^{(k)} Max\left[c_i(p_{k+1}), -\dfrac{\mu_i^{(k)}}{2r^{(k)}}\right] \quad i = 1,\ldots,m_i \\ r^{(k+1)} = \alpha\, r^{(k)} \quad \text{avec} \quad \alpha > 1 \end{cases} \tag{1.40}$$

Il existe de nombreuses variantes des formules (1.38), (1.39) et (1.40). On pourra par exemple consulter [Minoux 83], [Arora 91] ou [Fletcher 75]. Dans le cas d'une résolution de (1.38) à l'aide d'une méthode de quasi-Newton, Powell [Powell 78] propose d'utiliser l'approximation du hessien de la fonction Lagrangienne augmentée pour les formules de réactualisation (1.40).

Cet algorithme, a été utilisé en particulier par : Maniatty *et al.* [Maniatty 97] en extrusion, et Ghouati *et al.* [Ghouati 98b] pour optimiser la géométrie d'une entaille d'éprouvette, afin de minimiser la porosité finale après traction uniaxiale.

1.2.4 Algorithmes d'ordre 0 et méthodes probabilistes

Il s'agit de méthodes ne nécessitant pas le calcul du gradient de Φ. On cherche alors à générer un nouveau point, plus proche de l'optimum, uniquement à partir des valeurs de Φ pour un certain nombre de points dans l'espace des paramètres p,.

1.2.4.1 *L'algorithme du simplex* :

On appelle simplex de 3^n, un jeu de $n+1$ points linéairement indépendants. A chaque itération, un nouveau simplex est généré en éliminant le point correspondant à la plus forte valeur de Φ, et en déterminant un nouveau point à l'aide de l'opération de réflexion. La position de ce point est ensuite modifiée par contraction ou expansion suivant la valeur de Φ en ce point. Lorsque l'on est proche de la valeur optimale et qu'aucune des opérations

précédentes n'est satisfaisante, le simplex est contracté suivant toutes les directions, en utilisant l'opération de rétrécissement, réduisant ainsi la zone de recherche [Nelder 65]. La Figure 1.7 montre le principe des opérations pour n=2, et un simplex tel que $\Phi(x_h) > \Phi(x_m) > \Phi(x_l)$. L'isobarycentre $\bar{x}$ est défini par : $\bar{x}=(x_m+x_l)/2$.

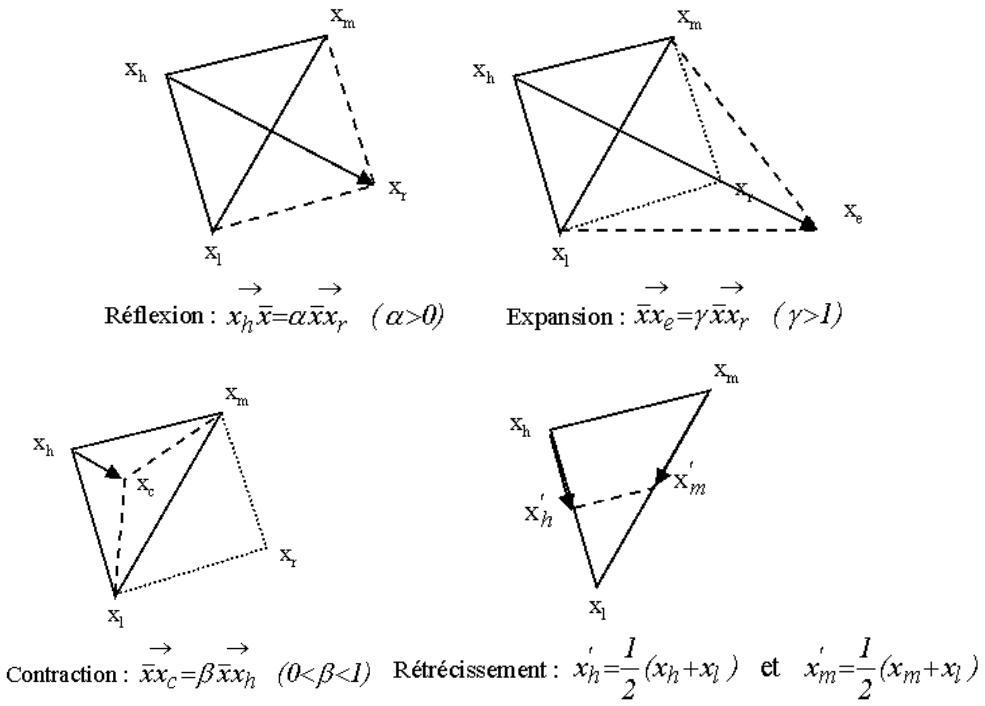


Figure 1.7 : *Opérations de base sur un simplex (n=2).*

Cet algorithme a été utilisé par Ohata *et al* [Ohata 98] pour l'optimisation d'un procédé 3D de mise en forme de plaques en deux opérations. On cherche alors les valeurs de trois paramètres, définissant les rayons et les hauteurs de la préforme, tels que l'épaisseur soit homogène en fin de procédé. La valeur optimale de *p* a été déterminée au prix de 389 simulations. Les auteurs précisent qu'il s'agit d'un minimum global sur le domaine admissible des paramètres, en comparant ce résultat avec le minimum obtenu par une méthode de grille fine. D'autres applications de cet algorithme ont été proposées. Coupez *et al.*[Coupez 99] l'utilisent par exemple pour l'optimisation du profil du champ de vitesse dans un procédé d'injection 3D. Notons qu'ici, seulement deux paramètres définissant les angles d'injection dans la pièce sont considérés. La convergence est alors réalisée en 12 itérations.

La méthode du simplex semble efficace tant que le nombre de paramètres est petit [Kusiak 89]. Lorsque le nombre de paramètres est supérieur à trois, elle semble mal adaptée du point de vue du coût. Nakamashi *et al.* [Nakamashi 98] ont ainsi proposé une étude comparative entre la méthode du simplex et la méthode heuristique du recuit simulé[2] [Aarts 89], sur le même cas que Ohata *et al*., mais pour sept paramètres. Le coût est tellement élevé que les auteurs proposent d'éliminer cinq paramètres par une méthode de Monte Carlo (nécessitant 100 simulations). On peut alors noter que la méthode de recuit simulé n'a nécessité « que » 50 simulations, contre 201 pour le simplex.

[2] En anglais : « Simulated annealing method»

L'algorithme du simplex, conçu à la base pour la minimisation de fonction sans contrainte, peut être étendu aux problèmes sous contraintes (1.3), en utilisant par exemple une technique de pénalisation.

1.2.4.2 *Les algorithmes génétiques* :

Un p-uplet (population *P*) de points expérimentaux (individus) est initialisé aléatoirement. Cette population évolue suivant un schéma de type Darwinien : les plus adaptés (en terme de valeurs de Φ) survivent et se reproduisent, jusqu'à faire apparaître les optima de Φ [Goldberg89, Michalewicz 96]. Les étapes principales de cette évolution sont :

Sélection : les individus les plus performants de *P* sont dupliqués, et forment une nouvelle population *P'* de même taille que *P*.

Croisement : les valeurs des paramètres caractérisant un point sont écrites sous forme binaire dans un vecteur formant un chromosome. Des couples d'individus choisis dans *P'* échangent un certain nombre de bits, donnant ainsi naissance à deux nouveaux individus. On génère alors une nouvelle population *P''* de même taille.

Mutation : chaque bit de chaque individu a une certaine probabilité d'être inversé.

Ce type d'algorithme permet d'obtenir un optimum global de la fonction coût, mais nécessite en contrepartie un nombre élevé d'évaluations de la fonctionnelle. Il existe toutefois des versions à population réduite[3] [Krishnakumar 89] dont les vitesses de convergence sont plus élevées.

Chung *et al.* [Chung 97] ont utilisé cet algorithme pour l'optimisation de filières d'extrusion. La forme de la filière est décrite à l'aide d'une courbe spline présentée sur la Figure 1.8. Les auteurs proposent de résoudre indépendamment trois problèmes : minimisation de la force d'extrusion, homogénéisation des déformations sur le produit extrudé et minimisation du pic de pression en entrée de filière, à l'aide d'un algorithme à population réduite (4 individus).

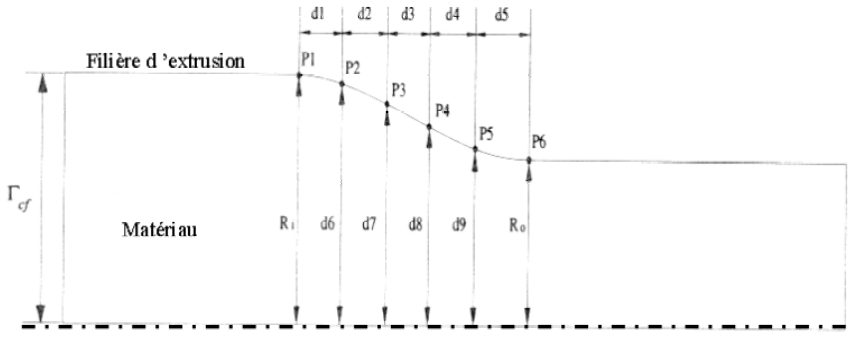


Figure 1.8 : *Paramétrisation de la filière d'extrusion [Chung 97].*

[3] Ce type d'algorithme est appelé 'Micro Genetic algorithm'.

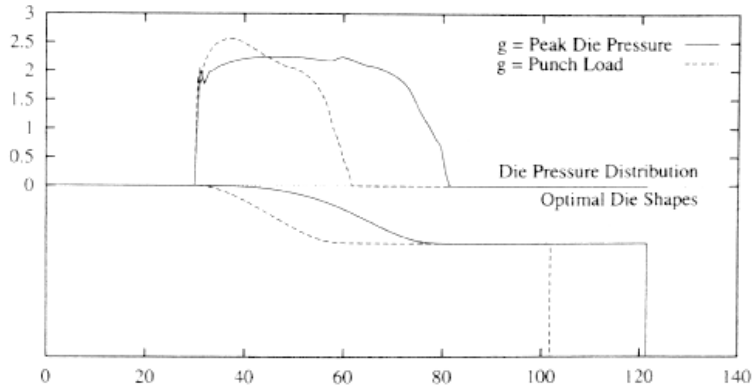


Figure 1.9 : *Formes optimales et distribution des pressions le long de la filière (--- : fonction coût = force d'extrusion), (— : fonction coût = pic de pression) [Chung 97].*

Ce type de méthode présente toutefois l'avantage de pouvoir travailler sur n'importe quel espace de recherche, continu ou discret. Roy *et al.* [Roy 97] ont ainsi tiré partie de cette propriété en étudiant le problème suivant : trouver le rapport hauteur/rayon du lopin initial, les paramètres géométriques (des angles essentiellement) des outils successifs ainsi que le nombre optimal de passes afin de minimiser le risque de rupture. Ce dernier paramètre prenant en effet des valeurs entières, la résolution de ce problème de conception avec une méthode de type gradient est impossible.

Les algorithmes génétiques, ainsi que d'autres algorithmes probabilistes (recuit simulé par exemple) semblent efficaces, et la prise en compte des contraintes peut se faire par une méthode de pénalisation. Néanmoins, les algorithmes génétiques utilisent certains paramètres qui doivent être fixés par l'utilisateur (taille de population, probabilité d'application des opérateurs génétiques, etc.), dont la mise au point ne peut se faire que par essais-erreurs. Mais le talon d'Achille de ces méthodes reste évidemment le coût en temps de calcul. Les exemples étudiés nécessitent en effet un nombre de simulations supérieur à 200. Ceci rend la méthode inenvisageable pour des problèmes complexes, où le calcul de la fonction est coûteux. Certes, certains travaux ont été effectués en vue d'utiliser une architecture parallèle [Talbi 96] qui permet de réduire le temps d'obtention d'une solution, mais elle ne fait que distribuer le temps de calcul sans réellement le réduire.

D'autres méthodes d'ordre 0, bâties à partir des méthodes de type gradient ont été proposées. Citons par exemple l'algorithme de Powell [Powell 64], basé sur le principe du gradient conjugué.

1.2.5 Méthodes statistiques

Le principe des méthodes statistiques consiste à approcher la fonction Φ (ainsi que les contraintes) sur l'espace des paramètres d'optimisation. On définit un plan d'expérience : pour chaque composante i du vecteur paramètre p, on définit M_i valeurs p_i^1, …, p_i^{Mi}. Parmi toutes les combinaisons possibles, on sélectionne alors N valeurs du vecteur paramètre p que l'on notera p^1, …, p^N. Le vecteur réponse y de la fonction Φ est calculé à partir des N valeurs de Φ par :

(1.41) $$y=\left[\Phi(p^1),\cdots,\Phi(p^N)\right]^T$$

on peut alors définir une approximation Φ_a de la fonction Φ sous la forme :

(1.42) $$\Phi_a(p)=\sum_{i=1}^{K}\alpha_i g_i(p)=\alpha.g(p) \qquad \forall p\in\mathbb{R}^n$$

où les fonctions prédéfinies g_i sont des monômes de p.

En règle générale, le nombre de points « expérimentaux » doit être supérieur au nombre de monômes utilisés : $N>K$. La détermination du vecteur inconnu α est réalisée par une minimisation au sens des moindres carrés de la fonction f :

(1.43) $$f(\alpha)=\sum_{i=1}^{N}\left[\Phi(p^i)-\Phi_a(p^i)\right]^2=\sum_{i=1}^{N}\left[\Phi(p^i)-\alpha.g(p^i)\right]^2$$

Ce qui conduit à la résolution du système :

(1.44) $$\alpha=\left(G^T G\right)^{-1} G y$$

où la matrice G (NxK) est formée des vecteurs $g(p_1)$, …, $g(p_N)$.

L'approximation des contraintes (lorsqu'elles existent) est réalisée de manière identique. Le problème général (1.3) est alors remplacé par le problème approché :

(1.45) $$\begin{cases} Min\,\Phi_a(p) \\ c_{a\,i}(p)\le 0 \;\;\forall\;\; i=1,..,m_i \\ h_{a\,i}(p)=0 \;\;\forall\;\; i=1,..,m_e \\ p\in\mathbb{R}^n \end{cases}$$

où toutes les fonctions s'expriment sous formes polynomiales. L'ensemble des méthodes présentées dans les sections précédentes peuvent s'appliquer, en remarquant que les gradients des fonctions d'approximation sont donnés par :

(1.46) $$\frac{\partial\Phi_a}{\partial p_j}(p)=\sum_{i=1}^{K}\alpha_i\frac{\partial g_i}{\partial p_j}(p) \qquad \forall p\in\mathbb{R}^n \quad j=1,\cdots,n$$

Les approximations des contraintes et leurs dérivées sont obtenues par des formules similaires. Quelque soit l'algorithme retenu, le temps de résolution de (1.45) est négligeable devant une seule simulation du procédé. Il existe d'autres formules d'interpolation pour (1.42). Lapokov *et al.* [Lapokov 97] utilisent par exemple des courbes splines cubiques.

Le choix des combinaisons à tester, leur nombre, ainsi que les modes d'interpolation, constituent les éléments déterminant de ces méthodes. La qualité de la solution obtenue est en grande partie fonction de ces paramètres. De nombreux ouvrages de référence ont été dédiés à

l'établissement de plans d'expérience répondant au mieux à ces problèmes [Schimmerling 98, Box 87]. Ces méthodes sont bien adaptées lorsque le nombre de paramètres est petit, ou lorsque leurs intervalles de définition sont réduits. Elles seront donc plutôt bien adaptées aux problèmes d'optimisation (faibles variations des paramètres), qu'aux problèmes de conception. Néanmoins, le nombre de simulations nécessaires à une bonne approximation de la fonction reste généralement élevé, ce qui constitue la principale limitation de ces méthodes. Comme pour les algorithmes génétiques, certains auteurs [Kopp 95] se sont attachés à utiliser des techniques de parallélisation pour réduire le temps d'évaluation des $\Phi(p^i)$.

Tout comme la méthode du simplex, ces techniques sont relativement indépendantes du problème d'optimisation et sont, de plus, simple à mettre en œuvre. Ces propriétés leur ont donc assuré une très large diffusion pour de nombreux problèmes d'optimisation en mise en forme. [Lapokov 98, Ghouati 99, Di Lorenzo 98, Santos 98, Liou 97]

1.3 Calcul des dérivées

Nous avons vu que dans le cadre des méthodes inverses, la majorité des algorithmes efficaces utilisent les dérivées de la fonctionnelle par rapport aux paramètres. Cette étude est généralement nommée *analyse de sensibilité*.

En un point fixe de l'espace x, la variation d'une variable $u(x,p)$ en fonction de p est définie par sa dérivée partielle $\partial u/\partial p$ par rapport à p. Néanmoins, dans la majorité des références dédiées à l'optimisation de forme, la sensibilité est définie par la dérivée totale :

$$\overset{\circ}{u} = \frac{du}{dp} = \frac{\partial u}{\partial p} + \frac{\partial u}{\partial x}\frac{dx}{dp} \tag{1.47}$$

D'un point de vue physique, il est en effet plus intéressant de connaître la variation d'une variable en un point matériel du domaine.

En optimisation de forme, l'analyse de sensibilité constitue l'étape la plus délicate et souvent la plus coûteuse en temps de développement. Les différentes méthodes sont généralement classées en trois groupes : les méthodes numériques, les méthodes semi-analytiques et les méthodes analytiques. Dans la dernière catégorie, plusieurs approches sont envisageables suivant que l'on calcule les dérivées par différentiation du modèle discret, ou bien par différentiation des équations continues, puis par discrétisation du problème de sensibilité. Le choix d'une méthode d'analyse est conditionné par le compromis coût-précision, les habitudes du laboratoire et le type de problème d'optimisation à résoudre

1.3.1 Méthodes numériques

Cette méthode, encore appelée « méthode des différences finies », consiste à approcher chaque composante du gradient de Φ en p par la formule :

$$\nabla_i \Phi \approx \frac{\Phi(p + \Delta p_i) - \Phi(p)}{\Delta p} \quad \text{avec} \quad \Delta p_i^T = [0\ 0 \ldots 0\ \Delta p\ 0 \ldots 0] \tag{1.48}$$

Le vecteur Δp_i a toutes ses composantes nulles, exceptée la i^{ème} qui est égale à la valeur de la perturbation Δp. Cette méthode permet donc de calculer les gradients de la fonctionnelle au prix de peu d'efforts de développement. En contrepartie, elle présente deux inconvénients majeurs. La méthode est coûteuse, car elle nécessite au moins $n+1$ calculs de Φ (pour un vecteur paramètre à n composantes). De plus, la précision de la dérivée dépend de la valeur de la perturbation Δp et de la précision des calculs de Φ. Il est possible d'augmenter la précision en utilisant un schéma centré :

$$\nabla_i \Phi \approx \frac{\Phi(p + \Delta p_i) - \Phi(p - \Delta p_i)}{2\Delta p} \tag{1.49}$$

Notons que le nombre de calculs nécessaires est alors égal à $2n$. D'un point de vue théorique, les approximations (1.48) ou (1.49) sont d'autant meilleures que la perturbation Δp est petite. Néanmoins, Δp ne peut pas être arbitrairement petite. Les erreurs d'arrondi et de discrétisation doivent être prises en compte. Si la variation Δp est trop faible, la différence entre $\Phi(p+\Delta p)$ et $\Phi(p)$ est de l'ordre de l'erreur de discrétisation, la valeur du gradient perd ainsi toute signification. A l'inverse, Δp ne peut pas être choisi trop grand, sous peine d'imprécision.

On trouve des applications de cette méthode en forgeage, pour l'optimisation de préformes [Becker 89] ou de la vitesse des outils [Boër 85], ainsi que dans d'autres domaines comme la mise en forme de plaques [Jensen 98] [Bourdin 98].

Les problèmes de précision ont clairement été mis en lumière par Mihelic *et al.* [Mihelic 98] lors de la résolution du problème d'étirage suivant : déterminer la forme de filière permettant de minimiser la force d'étirage. La forme de la filière est décrite par un polynôme du deuxième degré, dont les trois coefficients sont les paramètres d'optimisation.

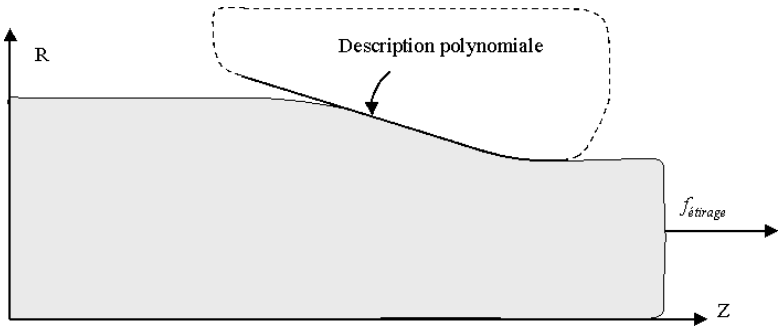


Figure 1.10 : *Etirage d'une barre métallique.*

Ces paramètres sont soumis à deux contraintes géométriques afin de conserver un diamètre minimal dans toutes les sections de la filière. La résolution du problème sous contraintes est effectuée par une méthode de programmation quadratique. Les gradients de la fonction coût et des contraintes sont calculés par un schéma aux différences finies.

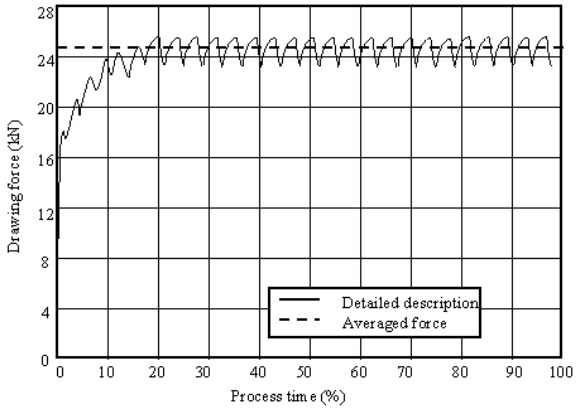


Figure 1.11: *Oscillations de la force d'étirage dues aux erreurs de discrétisation [Mihelic 98]*

On observe alors des oscillations de la force d'étirage (Figure 1.11) dues aux erreurs de discrétisation de la méthode éléments finis. Le calcul de la fonction coût par la méthode des différences finies devient alors impossible. Les auteurs proposent alors de diminuer la taille de maille et le pas de temps de façon importante, pour réduire ces oscillations, et de considérer une forme moyennée de la fonction coût [Mihelic 98].

L'imprécision des dérivées, et les temps de calcul élevés lorsque le nombre de paramètres devient important, sont très pénalisants. Dans le cas du forgeage, où les remaillages sont fréquents, il faut en plus considérer une nouvelle source d'imprécision liée au transport des variables, de l'ancien maillage vers le nouveau. Dans un cas d'écrasement d'un lopin cylindrique par un outil paramétré à l'aide d'une courbe spline, Bălan montre [Bălan 96] qu'au bout de seulement deux remaillages, les erreurs dues au transport sont supérieures aux variations de la fonction étudiée. L'approximation par différences finies n'ayant plus aucun sens, il est alors indispensable de calculer les sensibilités analytiquement.

1.3.2 Calcul analytique des dérivées : Méthodes variationnelles

Nous avons regroupé sous la dénomination « variationnelle » plusieurs méthodes d'analyse de sensibilité couramment rencontrées [Cardoso 88] [Zolesio 79]. Il est possible de classer ces méthodes suivant deux approches : les méthodes « globales », où le problème direct est différentié sous sa forme variationnelle (ou intégrale) [Pironneau 84], et les méthodes « locales », dans lesquelles la sensibilité de la fonction coût est obtenue par différentiation des équations continues régissant le problème direct [Ghouali 97]. Dans le cadre des procédés évolutifs, ces méthodes semblent difficiles, ou du moins délicates, à mettre en œuvre si on en juge par le nombre réduit de travaux leur étant dédiés. Nous présentons ici deux tentatives sur ce sujet.

Ghouali *et al.* [Ghouali 98] considèrent comme fonction coût l'énergie de mise en forme :

$$\Phi = \int_{t=t_0}^{t=t_{fin}} \int_{\Omega_t} \sigma : \dot{\varepsilon}\, dw\, dt \tag{1.50}$$

La différentiation de Φ par rapport aux paramètres de forme s'écrit :

$$\overset{\circ}{\Phi} = \int_{t=t_0}^{t=t_{fin}} \int_{\Omega_t} \left(\frac{\partial \sigma}{\partial p} : \dot{\varepsilon} + \sigma : \frac{\partial \dot{\varepsilon}}{\partial p} \right) dw\, dt + \int_{t=t_0}^{t=t_{fin}} \int_{\partial\Omega_t} \left(\sigma : \dot{\varepsilon}\, v^s . n \right) ds\, dt \tag{1.51}$$

où v^s désigne la vitesse de déplacement de la frontière $\partial\Omega_t$ due à la variation des paramètres de forme p. Afin de déterminer les dérivées $\partial\sigma/\partial p$, $\partial\dot{\varepsilon}/\partial p$ et le terme $v^s.n$ intervenant dans (1.51), on différentie les équations constitutives du problème direct ainsi que les conditions aux limites. On aboutit alors à un système d'équations locales, d'inconnues $\partial\sigma/\partial p$, $\partial v/\partial p$ et $v^s.n$, qui est résolu en discrétisant la forme variationnelle correspondante. La difficulté inhérente à cette méthode, provient des conditions aux limites auxquelles sont soumises les équations dérivées. Ces conditions font en effet intervenir les contraintes et leurs gradients [Ghouali 96] sur le bord du domaine. La résolution du problème direct permet de calculer σ sur la frontière, mais pas son gradient. Ce dernier n'est généralement pas défini, à moins d'utiliser des interpolations éléments finis C^1, dont la faible précision semble handicapante pour la méthode.

Afin de surmonter ce problème, Zabaras *et al.* [Zabaras 93] ont proposé d'utiliser la méthode de paramétrisation de domaine développée en mécanique des structures [Cardoso 88]. Cette dernière consiste à utiliser une configuration de référence Ω_0 sur laquelle se fait la résolution du problème de sensibilité. Les équations continues sont écrites sur Ω_0 en utilisant la jacobienne de la transformation ψ qui permet à tout instant de passer de la configuration déformée à la configuration de référence. Ω_0 étant indépendante des paramètres, on différentie ensuite les équations continues sur cette configuration. Le problème de sensibilité associé à cette différentiation est mis sous forme variationnelle, puis résolu par une méthode d'éléments finis. Dans cette formulation Lagrangienne totale, la variable principale est la sensibilité du gradient de la transformation ψ. Dans le cadre de la mise en forme, cette méthode a été appliquée à l'optimisation de forme de filières en extrusion [Badrinarayanan 96, Antúnez 96] et à l'optimisation du lopin initial en forgeage. Dans le dernier cas, le but est de déterminer la préforme qui élimine le bombé final après écrasement entre tas plats [Badrinarayanan 95]. La surface du lopin est décrite par des courbes splines et le matériau a un comportement viscoplastique et hyper-élastique. Les figures 1.12-1.13 montrent les différentes préformes optimales obtenues pour deux rapports d'écrasement différents.

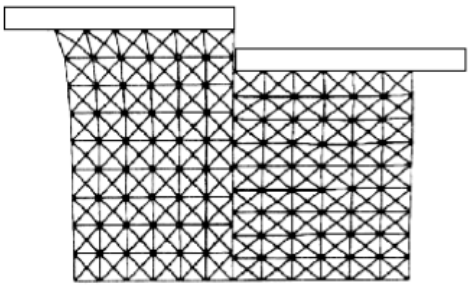


Figure 1.12 : *Préforme optimale et pièce forgée pour un écrasement de 17% [Badrinarayanan 95].*

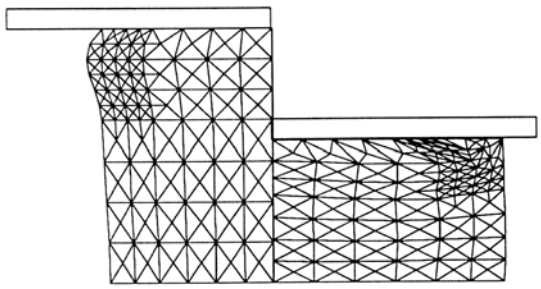


Figure 1.13 : *Préforme optimale et pièce forgée pour un écrasement de 44% [Badrinarayanan 95].*

1.3.3 Calcul analytique des dérivées : Méthodes discrètes

Les méthodes présentées dans cette section permettent de calculer le gradient de la fonction coût en différentiant les équations discrètes du modèle direct. Pour des problèmes stationnaires, des tests numériques montrent [Tortorelli 94] que sous certaines hypothèses, les approches discrètes et variationnelles sont équivalentes. La généralisation de cette propriété aux problèmes évolutifs semble plus difficile, en particulier à cause du traitement du contact. L'analyse de sensibilité par une approche discrète est souvent considérée comme étant plus consistante avec le problème direct. La fonction Φ n'est en effet accessible qu'au travers de sa valeur discrète $\tilde{\Phi}$, et différentier les équations discrètes permet, du moins d'un point de vue numérique, de calculer précisément les dérivées de la fonction coût $\tilde{\Phi}$ étudiée. L'approche discrète a été largement diffusée en optimisation structurale durant les deux dernières décennies.

Tout comme pour les méthodes variationnelles, la différentiation des équations discrètes peut être traitée de manière directe ou par la méthode de l'état adjoint, selon que l'on désire ou non obtenir les sensibilités de $\tilde{\Phi}$, en introduisant les calculs intermédiaires des gradients des variables d'état du problème direct.

Nous allons examiner les deux méthodes qui ont été utilisées pour l'optimisation de forme de filière dans un procédé de filage par Kusiak *et al.* [Kusiak 89] et Chen *et al.* [Chen 95]. Si les calculs de sensibilité obtenus avec l'une ou l'autre des méthodes sont équivalents, en terme de précision [Maniatty 96], les coûts en temps de calcul sont différents suivant le nombre de paramètres et de fonctions (ou de contraintes).

1.3.3.1 *Cas d'un problème stationnaire*

Le problème étant stationnaire et la loi de comportement viscoplastique, la fonctionnelle discrète $\tilde{\Phi}$ s'exprime en fonction du champ nodal des vitesses V, du vecteur des coordonnées du maillage X et du vecteur des paramètres p :

$$\tilde{\Phi}(p) = \tilde{\Phi}(X(p), V(p), p) \tag{1.52}$$

Dans le cas particulier du filage, la frontière du maillage en contact avec la partie à optimiser de la filière, est décrite par des fonctions polynomiales, dont les coefficients sont les paramètres de forme. La formule générale (1.52) correspond à une écriture où l'on considère aussi des paramètres de procédé (si seuls les paramètres géométriques devaient être considérés, $\tilde{\Phi}$ ne serait plus fonction de p qu'au travers des variables X et V).
En différentiant (1.52) par rapport à p :

$$\overset{\circ}{\tilde{\Phi}}(p) = \frac{\partial \tilde{\Phi}}{\partial X}\overset{\circ}{X} + \frac{\partial \tilde{\Phi}}{\partial V}\overset{\circ}{V} + \frac{\partial \tilde{\Phi}}{\partial p} \tag{1.53}$$

La sensibilité $\overset{\circ}{X}$ (cf. équation (1.47) est obtenue en dérivant l'expression polynomiale de la frontière. Les termes $\partial\tilde{\Phi}/\partial V$, $\partial\tilde{\Phi}/\partial X$ et $\partial\tilde{\Phi}/\partial p$ sont obtenus directement à partir de l'expression discrète de Φ. La seule inconnue est donc la sensibilité $\overset{\circ}{V}$ de la vitesse

La méthode de différentiation directe consiste à calculer explicitement $\overset{\circ}{V}$ en dérivant les équations discrètes du problème direct dont V est solution. Pour simplifier, on notera R ce système d'équations en se plaçant dans le cadre d'un comportement viscoplastique où l'incompressibilité est imposée par pénalisation :

$$R(p) = R(X(p), V(p), p) = 0 \tag{1.54}$$

En différentiant (1.54), $\overset{\circ}{V}$ apparaît comme la solution du système :

$$\overset{\circ}{V} = -\left[\frac{\partial R}{\partial V}\right]^{-1}\left(\frac{\partial R}{\partial X}\overset{\circ}{X} + \frac{\partial R}{\partial p}\right) \tag{1.55}$$

(1.55) revient à inverser un système linéaire pour chaque paramètre, tous ces systèmes ayant la même matrice $\partial R/\partial V$.

La méthode adjointe consiste à éliminer $\overset{\circ}{V}$ de l'équation (1.53) en considérant la fonction de Lagrange $\tilde{\Phi}_\lambda$ et le multiplicateur de Lagrange λ associé à (1.54) :

$$\tilde{\Phi}_\lambda = \tilde{\Phi} + R(X(p), V(p), p).\lambda \tag{1.56}$$

$\tilde{\Phi}_\lambda$ est égale à $\tilde{\Phi}$ pour V solution. En différentiant (1.56) et en regroupant les termes, on obtient la sensibilité de $\tilde{\Phi}_\lambda$ sous la forme :

$$\overset{\circ}{\tilde{\Phi}}_\lambda = \frac{\partial \tilde{\Phi}}{\partial p} + \frac{\partial R}{\partial p}\lambda + \left(\frac{\partial \tilde{\Phi}}{\partial X} + \frac{\partial R}{\partial X}\lambda\right)\overset{\circ}{X} + \left(\frac{\partial \tilde{\Phi}}{\partial V} + \frac{\partial R}{\partial V}\lambda\right)\overset{\circ}{V} = \frac{\partial \tilde{\Phi}}{\partial p} + \frac{\partial R}{\partial p}\lambda + \left(\frac{\partial \tilde{\Phi}}{\partial X} + \frac{\partial R}{\partial X}\lambda\right)\overset{\circ}{X} \tag{1.57}$$

où le multiplicateur λ est solution du système :

$$\frac{\partial R}{\partial V}\lambda = -\frac{\partial \tilde{\Phi}}{\partial V} \tag{1.58}$$

Le terme $\overset{\circ}{V}$ disparaît des équations, et $\overset{\circ}{\tilde{\Phi}} = \overset{\circ}{\tilde{\Phi}}_\lambda$ pour V solution. Le calcul de la sensibilité de $\tilde{\Phi}$ est donc assujetti au calcul du vecteur λ, solution de (1.58). Il y a donc autant de systèmes linéaires à résoudre que de fonctions coût ou de contraintes. Cette méthode devient donc très attrayante lorsque le nombre de paramètres est supérieur au nombre de fonctions à dériver, ce qui est généralement le cas.

1.3.3.2 *Extension aux problèmes instationnaires*

La méthode adjointe est généralement appliquée aux problèmes non évolutifs [Ulysse 99, Maniatty 97]. Dans le cadre de l'élastoplasticité, Tortorelli *et al.* [Tortorelli 94] ont proposé une extension de cette méthode aux cas où le comportement dépend de l'histoire de la déformation. Cette dépendance est exprimée en introduisant dans l'équation résiduelle d'équilibre, le champ de déplacement à l'incrément précédent. Dans la pratique, cette écriture semble délicate. Il nous semble plus réaliste d'exprimer l'histoire de la déformation au travers

de la variable *X*. L'équation résiduelle d'équilibre s'écrira alors à tout instant *i* :

$$R_i(V^i(p), X^i(p), p) = 0 \quad i = 0, \cdots, N-1 \tag{1.59}$$

Les équations sont sensiblement plus complexes, aussi, nous nous limiterons à une fonctionnelle qui dépend exclusivement de la configuration finale, obtenue après *N* incréments de calcul :

$$\tilde{\Phi}(p) = \tilde{\Phi}(X^N(p), p) \tag{1.60}$$

A tout instant *i*, la variable X^i s'écrit donc comme une fonction de l'ensemble des champs de vitesse calculés aux incréments précédents :

$$X^i = X^0 + \Psi^i(V^0, \cdots, V^{i-1}) \tag{1.61}$$

Pour éliminer les sensibilités $\overset{\circ}{V}{}^i$, il faut alors introduire dans la fonction de Lagrange toutes les contraintes (1.59) correspondant à tous les incréments de calcul :

$$\tilde{\Phi}_\lambda = \tilde{\Phi} + \sum_{i=0}^{N-1} R_i . \lambda_i \tag{1.62}$$

En différentiant (1.62) et en tenant compte de (1.59)-(1.60) :

$$\begin{aligned} \overset{\circ}{\tilde{\Phi}}_\lambda = & \frac{\partial \tilde{\Phi}}{\partial p} + \frac{\partial \tilde{\Phi}}{\partial X^N}\left(\overset{\circ}{X}{}^0 + \sum_{i=0}^{N-1} \frac{\partial \Psi^N}{\partial V^i} \overset{\circ}{V}{}^i \right) + \left(\frac{\partial R_0}{\partial V^0} \overset{\circ}{V}{}^0 + \frac{\partial R_0}{\partial X^0} \overset{\circ}{X}{}^0 \right) \\ & \sum_{i=1}^{N-1} \left(\frac{\partial R_i}{\partial V^i} \overset{\circ}{V}{}^i + \frac{\partial R_i}{\partial X^i} \left(\overset{\circ}{X}{}^0 + \sum_{k=0}^{i-1} \frac{\partial \Psi^i}{\partial V^k} \overset{\circ}{V}{}^k \right) \overset{\circ}{X}{}^i + \frac{\partial R_i}{\partial p} \right) . \lambda_i + \sum_{i=0}^{N-1} \frac{\partial R_i}{\partial p} . \lambda_i \end{aligned} \tag{1.63}$$

Si on regroupe les termes en $\overset{\circ}{V}{}^i$ dans l'équation précédente :

$$\begin{aligned} \overset{\circ}{\tilde{\Phi}}_\lambda = & \frac{\partial \tilde{\Phi}}{\partial p} + \left(\frac{\partial \tilde{\Phi}}{\partial X^N} + \sum_{i=0}^{N-1} \frac{\partial R_i}{\partial X^i} \lambda_i \right) \overset{\circ}{X}{}^0 \\ & + \sum_{i=0}^{N-1} \left(\frac{\partial \tilde{\Phi}}{\partial X^N} \frac{\partial \Psi^N}{\partial V^i} + \frac{\partial R_i}{\partial V^i} \lambda_i + \sum_{k=i+1}^{N-1} \left(\frac{\partial R_k}{\partial X^k} \frac{\partial \Psi^k}{\partial V^i} \lambda_k \right) \right) \overset{\circ}{V}{}^i + \sum_{i=0}^{N-1} \frac{\partial R_i}{\partial p} . \lambda_i \end{aligned} \tag{1.64}$$

Les multiplicateurs de Lagrange sont donc solutions des systèmes :

$$(1.65)\qquad \begin{cases} \dfrac{\partial R_{N-1}}{\partial V^{N-1}}\lambda_{N-1} = -\dfrac{\partial \tilde{\Phi}}{\partial X^N}\dfrac{\partial \Psi^N}{\partial V^{N-1}} \\ \dfrac{\partial R_i}{\partial V^i}\lambda_i = -\dfrac{\partial \tilde{\Phi}}{\partial X^N}\dfrac{\partial \Psi^N}{\partial V^i} - \sum_{k=i+1}^{N-1}\dfrac{\partial R_k}{\partial X^k}\dfrac{\partial \Psi^k}{\partial V^i}\lambda_k \qquad i = N-2,\cdots,0 \end{cases}$$

Si on analyse le coût de la méthode, on voit que l'on doit résoudre un système linéaire par incrément de calcul. Les matrices de ces systèmes sont les matrices hessiennes déjà calculées pour la résolution du problème direct. Mais la résolution de (1.65) se fait en descendant : le système en λ_i fait intervenir λ_{N-1}, …, λ_{i+1}. Ceci implique de stocker toutes les matrices $\partial R_i/\partial V$ et $\partial R_i/\partial X$, jusqu'à la fin de la simulation. Dans le cas du forgeage, le nombre d'incréments est souvent élevé, ce qui rend cette méthode prohibitive du point de vue des ressources de mémoire nécessaires. Une alternative consiste à recalculer ces matrices à partir des informations de reprise des calculs (variables nodales X et V, conditions aux limites, etc.), stockées à tous les incréments de la simulation. Mais le gain de place mémoire qui en résulte est alors compensé par un coût de calcul plus élevé. A cause des temps de calcul et de factorisation des matrices manquantes, cette méthode s'avère beaucoup trop coûteuse si les systèmes linéaires sont résolus par une méthode directe. Elle ne peut être raisonnablement envisagée que lorsque un solveur itératif est utilisé et que le nombre de paramètre est grand.

L'extension aux procédés instationnaires de l'analyse de sensibilité par différentiation directe des équations discrètes a été proposée par Chenot et Fourment [Chenot 93a]. Le principe est de calculer $\mathring{X}$ et $\mathring{V}$ à tous les incréments de temps. Cette méthode est présentée en détail au chapitre 4. Comme dans le paragraphe 1.3.3.1, le calcul de $\mathring{V}$ équivaut à la résolution d'un système linéaire par paramètre d'optimisation (ayant tous la même matrice), et ceci pour chaque incrément. Contrairement à la méthode adjointe, le fait de connaître explicitement $\mathring{X}$ et $\mathring{V}$ à tout instant permet de calculer les sensibilités des variables principales du problème direct qui découlent de X et V. Ces informations permettent, pour un procédé donné, de mesurer quantitativement l'influence de certains paramètres, et d'en déduire dans un deuxième temps ceux dont il est important de déterminer la valeur optimale.

1.3.4 Méthodes semi-analytiques

Ces méthodes ont été initiées en optimisation structurale [Haftka 92]. Elles peuvent être vues comme une voie médiane entre l'analyse de sensibilité analytique et numérique. Comme pour les méthodes analytiques discrètes, le principe est de calculer la dérivée de $\tilde{\Phi}$ en différentiant la formulation discrétisée du problème mécanique. Mais certaines dérivées sont calculées par une méthode de différences finies.

Joun *et al.* [Joun 91] proposent par exemple, de calculer l'expression (1.53), en utilisant les approximations :

$$(1.66)\qquad \frac{\partial \tilde{\Phi}}{\partial X}\mathring{X}_i = \frac{\tilde{\Phi}(X(p+\Delta p_i),p) - \tilde{\Phi}(X(p),p)}{\Delta p_i} \qquad i = 1,\cdots,n$$

$$\frac{\partial \tilde{\Phi}}{\partial p_i} = \frac{\tilde{\Phi}(X(p), p + \Delta p_i) - \tilde{\Phi}(X(p), p)}{\Delta p_i} \qquad i = 1, \cdots, n \tag{1.67}$$

et les formules identiques pour les termes $\partial R/\partial X$ et $\partial R/\partial p$ intervenant dans (1.55). Les termes $\acute{c}\tilde{\Phi}/\partial V$ et $\partial R/\partial V$ sont calculés analytiquement. La sensibilité de $\tilde{\Phi}$ est finalement obtenue en utilisant une méthode adjointe.

Cette méthode implique donc $n+1$ calculs, à V fixé, du vecteur résidu R et de la fonctionnelle. Ce qui est beaucoup moins coûteux que les $n+1$ simulations éléments finis nécessaires à une méthode numérique. L'intérêt de cette approche découle des avantages liés aux méthodes numériques : simplicité de mise en œuvre et possibilité de considérer des paramètres et des lois de comportement variées. Dans le cas des procédés stationnaires, l'inconvénient majeur réside dans le choix de la perturbation Δp.

L'extension aux problèmes instationnaires a été proposée par Chung *et al.* [Chung 98]. Les auteurs renoncent à l'utilisation d'une méthode adjointe et se basent sur une approche directe. La prise en compte de l'évolution du contact, entre deux instants de calcul, introduit certaines difficultés. Pour tout incrément de temps, le calcul des termes de sensibilité par différence finies, suppose en effet que les conditions aux limites soient identiques dans l'état de référence, et dans l'état perturbé obtenu pour une variation Δp des paramètres. Dans le cas contraire, une résolution complète est alors nécessaire pour calculer toutes les variables d'état dans la configuration perturbée. Les formules d'actualisation des sensibilités nodales qui en découlent sont sensiblement plus compliquées [Chung 98].

La méthode semble toutefois pouvoir être simplifiée, en considérant la dérivation analytique du contact, telle que nous la présenterons par exemple au §4.4 Le problème du remaillage n'a pas été abordé par les auteurs. Il est essentiel que les sensibilités nodales, obtenues par un schéma aux différences finies, soient calculées sur le même maillage. Or la définition d'un état perturbé, obtenu par une modification Δp des paramètres géométriques, peut entraîner une dégénérescence des éléments, et par conséquent, des maillages différents pour l'état de référence et l'état perturbé.

1.3.5 Les outils informatiques pour l'analyse de sensibilité.

Les logiciels de différentiation automatique sont des outils informatiques permettant, à partir du code calculant une fonction, d'obtenir un nouveau code calculant les dérivées partielles exactes de cette fonction par rapport à ses arguments (paramètres). Considérons une fonction :

$$\begin{aligned} f : \quad & \mathbb{R} \to \mathbb{R}^m \\ & x \mapsto [f_1(x), \ldots, f_m(x)] \end{aligned} \tag{1. 68}$$

Le but est de construire un nouveau code qui calcule $\partial f_i / \partial x_j$ ($i=1,..,m$; $j=1,..,n$), où x_j désigne les variables indépendantes sur lesquelles opère f. La fonction f est alors décomposée en fonctions élémentaires dont les dérivées sont connues (+, -, *, /, *sin, exp, ...)*.

Les arguments de sortie du nouveau code sont des combinaisons linéaires des dérivées

partielles $\partial f_i/\partial x_j$, dont les coefficients sont les variables d'entrée. L'ensemble des dérivées partielles est ainsi obtenu par appels successifs du code dérivé pour différentes valeurs des arguments d'entrée (généralement 0 ou 1). On distingue deux méthodes pour générer le code dérivé : le mode direct et le mode inverse. Le premier consiste à dériver, pas à pas, chaque instruction intervenant dans le calcul des f_i. Le principe du mode inverse est similaire à celui de l'état adjoint. Le calcul des dérivées s'effectue alors en ordre inverse de l'ordre d'exécution du code original [Rostaing 93].

Ces méthodes diffèrent par leur coût d'utilisation, tant au niveau de la place mémoire, que du temps de calcul. Pour le mode direct, le nombre de calculs effectués par le code dérivé, est proportionnel au nombre de variables d'entrée. Cette méthode s'applique donc préférentiellement, lorsque l'on souhaite dériver un nombre élevé de fonctions par rapport à un petit nombre de paramètres. Pour le mode inverse, le nombre de calcul est indépendant du nombre de paramètres par rapport auxquels on dérive. Ceci rend la méthode attractive lorsque le nombre de paramètre est élevé. Par contre, le nombre de variables intermédiaires qu'il est nécessaire de stocker, peut devenir critique en gestion de place mémoire, si le nombre d'instructions pour le calcul de f est important. Des techniques de différentiation automatique ont été proposées afin de calculer les dérivées d'ordre supérieur d'une fonction [Guillaume 93, Navarrina 98]. Cette technique s'avère par exemple intéressante, lors de la phase de recherche linéaire pour la minimisation unidimensionnelle de la fonction coût (cf. §1.2.1.2). La fonction est alors approchée dans la direction de descente par un développement de Taylor [Mader 95] :

$$\tilde{\Phi}(p_k + \lambda d_k) = \tilde{\Phi}(p_k) + \sum_{j=1}^{m} \frac{\lambda^j}{j!} D^j \tilde{\Phi}(p_k).d_k^j + o\left(\|\lambda d_k\|^j\right) \tag{1.69}$$

L'utilisation de ces outils nécessite néanmoins certaines précautions. Le choix du mode utilisé pour dériver dépend du problème que l'on cherche à résoudre. Lorsque le nombre de fonctions est égal au nombre de paramètres (par exemple le calcul de $\partial R/\partial X$ où R et X désignent deux vecteurs de taille 2xNombre de nœuds), le choix de la méthode devient plus délicat et doit s'appuyer sur la complexité des calculs. De plus, le code dérivé peut être appelé un grand nombre de fois (2xNombre de nœuds dans le cas précédent), il est alors nécessaire « d'épurer » le code dérivé afin d'enlever tout calcul intermédiaire inutile. Suivant les cas, ce travail peut s'avérer aussi long que les tests de vérifications, qui constituent la partie la plus fastidieuse d'un calcul analytique.

Chapitre 2

Simulation du procédé de forgeage

Nous présentons dans ce chapitre la formulation et les équations générales du problème de mise en forme par forgeage, ainsi que la description des algorithmes de résolution implémentés dans le code de simulation FORGE2® développé au CEMEF.

2.1 Formulation du problème

En se plaçant à l'échelle macroscopique, le matériau est considéré comme homogène et isotrope. Sa déformation est caractérisée par une partie irréversible (déformation plastique) et une partie réversible (déformation élastique). Dans le cadre de la mise en forme des métaux à chaud, la composante élastique de la déformation est faible au regard de la composante plastique. De plus, nous n'effectuons pas de calcul des contraintes résiduelles, et nous nous intéressons seulement à la simulation de l'écoulement pendant le procédé de forgeage. Aussi, nous négligeons la contribution élastique à la déformation et nous utiliserons une loi de comportement viscoplastique. Nous faisons, de plus, l'hypothèse que les phénomènes d'échanges thermiques avec les outils et l'air ambiant n'influencent pas significativement l'écoulement. Nous nous limitons, par conséquent, à une approche isotherme du procédé. Pour plus de détails, le lecteur pourra consulter [Cescutti 89] pour la formulation mécanique et [Soyris 90] pour l'aspect thermodynamique dans un cadre plus général.

2.1.1. Les équations du problème continu

L'état des contraintes du matériau est défini par le tenseur des contraintes de Cauchy σ, qui est décomposé en une partie déviatorique s représentant le cisaillement, et une partie sphérique représentant la pression hydrostatique p.

$$s = \sigma + pI \quad \text{avec} \quad p = -\frac{1}{3} Tra(\sigma) \tag{2.1}$$

où I désigne la matrice identité. En considérant les hypothèses précédentes, nous utilisons donc une loi de comportement viscoplastique de Norton-Hoff :

$$s = \frac{2\bar{\sigma}}{3\dot{\bar{\varepsilon}}} \dot{\varepsilon} \tag{2.2}$$

$$\bar{\sigma} = K\sqrt{3}\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^m \tag{2.3}$$

La contrainte équivalente $\bar{\sigma}$ s'exprime ainsi en fonction de la consistance K du matériau, du coefficient de sensibilité à la vitesse de déformation m et de la vitesse de déformation

équivalente $\dot{\bar{\varepsilon}}$.

(2.4) $$\dot{\bar{\varepsilon}} = \sqrt{\frac{2}{3}\dot{\varepsilon} : \dot{\varepsilon}}$$

(2.5) $$\dot{\varepsilon} = \frac{1}{2}\left(grad(v) + grad^T(v)\right)$$

où v désigne le champ de vitesse. A tout instant t du procédé, la valeur de la déformation équivalente (ou cumulée) est obtenue par intégration temporelle de (2.4) :

(2.6) $$\bar{\varepsilon} = \int_{t=0}^{t} \dot{\bar{\varepsilon}}\, dt$$

L'écrouissage du matériau est introduit par la dépendance de la consistance K à la déformation :

(2.7) $$K = K_0\left(\bar{\varepsilon} + \bar{\varepsilon}_0\right)^n e^{\beta / T}$$

La température T est supposée constante. De manière plus générale, pour la loi de Norton-Hoff, le déviateur des contraintes dérive du pseudo-potentiel conjugué de dissipation :

(2.8) $$s = \frac{\partial \varphi^c}{\partial \dot{\varepsilon}} \text{ où } \varphi^c = \frac{K}{m+1}\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^{m+1}$$

Dans la plupart des procédés de forgeage, les forces d'inertie et de pesanteur sont négligeables devant les forces dues à la viscosité. L'équation d'équilibre du matériau est donc définie à un instant donné par :

(2.9) $$\mathrm{div}\sigma = 0$$

La loi de comportement est complétée par la condition d'incompressibilité du matériau :

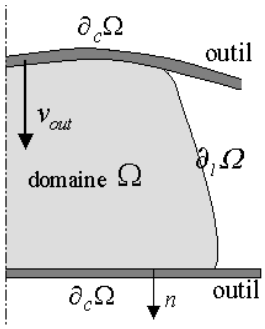


Figure 2.1 : *Conditions aux limites.*

(2.10) $$\mathrm{div}v = 0$$

Ces équations sont soumises aux conditions aux limites suivantes :
- condition de surface libre :

(2.11) $$\sigma.n = 0 \text{ sur } \partial_l\Omega$$

- condition de non-pénétration avec les outils :

(2.12) $$\begin{cases} (v - v_{out}).n \le 0 \text{ et } (\sigma n).n \le 0 \text{ sur } \partial_c\Omega \\ \text{avec } \left[(v - v_{out}).n\right]\left[\sigma n.n\right] = 0 \end{cases}$$

où v_{out} désigne la vitesse de l'outillage, n la normale extérieure à la matière et $\partial_c\Omega$ (respectivement $\partial_l\Omega$) la surface de contact (respectivement la surface libre). Nous utilisons une loi de frottement qui exprime l'existence d'un potentiel dissipatif convexe χ, dont dérive le vecteur cission de frottement τ :

$$\tau = \sigma n - [(\sigma n).n]n = -\frac{\partial \chi(v)}{\partial v} \tag{2.13}$$

Les procédés de forgeage étant caractérisés par de très fortes pressions, nous utilisons en première approximation un potentiel de Norton :

$$\chi(v) = \frac{\alpha_f K_f}{p_f + 1} \left\| \Delta v_g \right\|^{p_f + 1} \tag{2.14}$$

où Δv_g désigne la vitesse relative de glissement :

$$\Delta v_g = v - v_{out} - [(v - v_{out}).n]n \tag{2.15}$$

Le vecteur cission de frottement à l'interface s'exprime alors par :

$$\tau = -\alpha_f K_f \left\| \Delta v_g \right\|^{p_f} \frac{\Delta v_g}{\left\| \Delta v_g \right\|} \tag{2.16}$$

Par la suite, nous considèrerons $K=K_f$ et $m=p_f$. Le frottement est alors fonction d'un seul paramètre α_f déterminé de manière expérimentale.

2.1.2 Formulation faible

Afin de résoudre le système d'équations précédent, l'équation d'équilibre est écrite sous forme faible. Cette formulation est obtenue en considérant une description Eulérienne, où l'inconnue est le champ des vitesses :

$$\text{trouver } v \in V_{div}^{ca} \text{ tel que } \forall v^* \in V_{div}^{ca0} : \int_\Omega \sigma : \dot{\varepsilon}^* dw - \int_{\partial_c \Omega} \tau . v^* ds = 0 \tag{2.17}$$

V_{div}^{ca} désigne l'espace vectoriel des champs de vitesses incompressibles et cinématiquement admissibles (i.e. : vérifiant la condition de bord (2.12)), et V_{div}^{ca0} l'ensemble des champs incompressibles et cinématiquement admissibles à zéro (i.e. vérifiant la condition $v.n \leq 0$ sur $\partial_c\Omega$).

L'existence et l'unicité de la solution sur l'espace V_{div}^{ca} sont assurées en remarquant que l'équation (2.17) est équivalente [Germain 85] à la condition de stationnarité de la fonctionnelle strictement convexe $\Psi(v)$:

(2.18) $$\Psi(v)=\int_{\Omega}\varphi^{c}(v)dw+\int_{\partial_c\Omega}\chi(v)ds=\int_{\Omega}\frac{K}{m+1}\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^{m+1}dw+\int_{\partial_c\Omega}\frac{\alpha_f K}{p+1}\left\|\Delta v_g\right\|^{p+1}ds$$

Formulation vitesse-pression

Dans la pratique, la construction des champs de vitesse incompressibles V_{div}^{ca} et V_{div}^{ca0} s'avère difficile. On préfèrera utiliser l'espace V^{ca} (respectivement V^{ca0}) des champs admissibles (respectivement admissibles à zéro) sans autre restriction. On reprend alors les équations (2.9) et (2.10), que l'on écrit sous leur forme faible : trouver $v\in V^{ca}$ et $p\in P$ tel que :

(2.19) $$\begin{cases}\forall v^{*}\in V^{ca0} & \displaystyle\int_{\Omega}s:\dot{\varepsilon}^{*}dw-\int_{\partial_c\Omega}\tau.v^{*}ds-\int_{\Omega}p\,\mathrm{div}v^{*}dw=0\\ \forall p^{*}\in P & \displaystyle\int_{\Omega}p^{*}\,\mathrm{div}v\,dw=0\end{cases}$$

On introduit la pression comme variable supplémentaire. $P=L^2(\Omega)$ désigne l'espace des pressions et des fonctions tests pour l'équation d'incompressibilité.

2.2 Discrétisation

Nous nous limitons à la simulation des procédés axisymétriques (symétrie axiale repérée par l'axe vertical *z*). Le problème ainsi formulé est bidimensionnel. Le cas échéant, un plan de symétrie supplémentaire pourra être considéré.

2.2.1 Discrétisation spatiale : méthode éléments finis

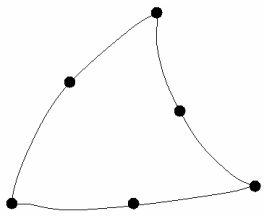


Figure 2.2 : *Elément triangulaire quadratique*

La discrétisation consiste à projeter le champ continu *v* sur un espace discret $\tilde{v}$. Pour ce faire, le domaine Ω représentant la pièce, est discrétisé à l'aide d'éléments quasi isoparamétriques P2-P0. La vitesse est interpolée de manière quadratique et continue. La pression est discontinue et constante par élément. Dans le cadre des problèmes axisymétriques, nous employons une interpolation spatiale dans l'espace (r^2,z) et une interpolation des vitesses dans l'espace (rv_r, v_z). Cette méthode permet de calculer plus précisément la divergence du champ de vitesse nécessaire pour imposer l'incompressibilité [Abouaf 87] :

(2.20) $$\tilde{r}=\sqrt{\sum_{k=1}^{Nbnoe}N^{k}R_{k}^{2}}\ ,\ \tilde{z}=\sum_{k=1}^{Nbnoe}N^{k}Z_{k}$$

Les champs de vitesse et de pression discrets sont donnés par :

$$\widetilde{r}\widetilde{v}_r = \sum_{k=1}^{Nbnoe} N^k R_k V_k^r \ , \ \widetilde{v}_z = \sum_{k=1}^{Nbnoe} N^k V_k^z \quad \text{et} \quad \widetilde{p} = \sum_{e=1}^{Nbelt} \overline{N}^e P_e \tag{2.21}$$

où k est un nœud du maillage (nombre de nœuds = *Nbnoe,* nombre d'éléments = *Nbelt*) de vecteur coordonnée $X_k=(R_k, Z_k)$ et ayant $V_k=(V_k^r, V_k^z)$ pour vecteur vitesse. Les vecteurs nodaux globaux des vitesses et des coordonnées du maillage sont notés X et V. N^k désigne la fonction d'interpolation au nœud k et $\overline{N}^e$ la fonction d'interpolation de la pression sur l'élément e. A partir de (2.5), on définit pour tout nœud k du maillage l'opérateur B tel que :

$$\widetilde{\dot{\varepsilon}} = \sum_{i=1}^{i=2Nbnoe} \sum_{k=1} B^{ik} V_k^i \tag{2.22}$$

On définit les sous espaces $\widetilde{V}^{ca}$, $\widetilde{V}^{ca0}$ et $\widetilde{P}$ de dimension finie, engendrés par les fonctions de base discrétisant les vitesses et les pressions. Conformément à la méthode Galerkin, les fonctions d'interpolation N^k et $\overline{N}^e$ (constantes par élément pour la triangulation P2P0) sont utilisées comme fonctions tests v^* et p^*. Le problème (2.19) s'écrit alors sous forme discrète:

$$\begin{cases} \forall k = 1,..Nbnoe; i = 1,2 \\ \\ \displaystyle\int_{\Omega} 2K\left(\sqrt{3}\widetilde{\dot{\varepsilon}}\right)^{m-1} : B^{ik} dw - \int_{\partial_c \Omega} \alpha_f K \left\| \Delta \widetilde{v}_g \right\|^{m-1} \left(\Delta \widetilde{v}_g\right)_i . N^k ds - \sum_e P_e \int_{\Omega_e} Tra(B^{ik}) dw = 0 \\ \\ \forall e = 1, \cdots, Nbelt \quad \displaystyle\int_{\Omega_e} \mathrm{div}\widetilde{v} dw = 0 \end{cases} \tag{2.23}$$

Afin de satisfaire la condition de compatibilité de Brezzi-Babuska, par commodité, on peut définir l'opérateur divergence discrète [Fortin 82], qui remplace l'opérateur divergence au moment de la discrétisation, et qui s'écrit pour les éléments P2P0 considérés :

$$\forall e = 1, \cdots, Nbelt \ : \ \mathrm{d\widetilde{i}v^e} v = \frac{1}{V^e} \int_{\widetilde{\Omega}_e} \mathrm{div}\widetilde{v} dw \tag{2.24}$$

V^e désignant le volume de l'élément $\widetilde{\Omega}_e$ du maillage.

Cette approche permet de résoudre l'équation d'incompressibilité avec la même précision que l'équation d'équilibre mécanique, mais au prix de l'augmentation du nombre de variables. La méthode retenue consiste, à partir de (2.23), à éliminer la pression par pénalisation de l'équation d'incompressibilité. Le problème à résoudre s'écrit : trouver X et V tels que :

$$(2.25)\quad \begin{cases} \forall k = 1,..Nbnoe, \forall i = 1,2 : R^{k,i}(X,V) = R_c^{k,i}(X,V) + R_f^{k,i}(X,V) + R_p^{k,i}(X,V) = 0 \\ \forall\, e = 1,\cdots, Nbelt \quad P_e = -\dfrac{\rho}{V^e}\displaystyle\int_{\Omega_e} div\tilde{v}\, dw \\ \text{et vérifiant les conditions aux limites (2.12)} \end{cases}$$

où

$$R_c^{k,i}(X,V) = \int_{\tilde{\Omega}} 2K\left(\sqrt{3}\dot{\varepsilon}\right)^{m-1} \dot{\varepsilon} : B_i^k\, dw$$

$$(2.26)\quad R_f^{k,i}(X,V) = \int_{\partial_c\tilde{\Omega}} \alpha_f K \left\| \Delta\tilde{v}_g \right\|^{p-1} \left(\Delta\tilde{v}_g\right)_i N^k\, ds$$

$$R_p^{k,i}(X,V) = \rho \sum_{e=1}^{Nbelt} \frac{1}{V^e} \int_{\tilde{\Omega}_e} \mathrm{div}\tilde{v}\, dw \int_{\tilde{\Omega}_e} trace(B_i^k)\, dw$$

Par abus de notation, nous omettrons par la suite l'indice *i* des deux directions de l'espace. R^k désigne alors le vecteur résidu du problème éléments finis écrit au nœud *k*.

2.2.2 Discrétisation temporelle

L'intervalle de temps [0, t_f] du procédé est discrétisé en plusieurs pas de temps. L'écoulement du matériau entre deux incréments de temps successifs *t* et $t+\Delta t$, est donné par un schéma d'intégration temporelle explicite :

$$(2.27)\quad \begin{cases} X_k^{t+\Delta t} = X_k^t + \Delta t V_k^t \\ \bar{\varepsilon}_k^{t+\Delta t} = \bar{\varepsilon}_k^t + \Delta t \dot{\bar{\varepsilon}}_k \end{cases} \quad \forall k = 1,.. Nbnoe$$

Ce schéma revient à faire l'hypothèse que le champ de vitesse reste constant durant l'incrément de temps Δt. Nous verrons par la suite les erreurs que peut entraîner une telle approximation. L'écriture (2.27) n'est en réalité vraie que pour les nœuds dont la condition de contact n'évolue pas entre les instants *t* et $t+\Delta t$. Le cas contraire est étudié dans la section suivante.

2.2.3 Traitement des conditions aux limites

On se place à un incrément de temps donné t_i. On suppose que l'ensemble des nœuds frontaliers $\partial_c\Omega_t$ en contact avec les outils est connu. Au niveau discret, la condition de contact unilatéral (2.12) n'est vérifiée que par les nœuds frontière du maillage :

$$(2.28)\quad \forall k \in \partial_c\tilde{\Omega}_t \qquad V_k . n_k = V_{out} . n_k$$

où n_k désigne la normale intérieure à l'outillage. L'équation (2.28) revient à imposer la composante normale de la vitesse. Elle est éliminée de l'équation d'équilibre, qui est alors écrite seulement suivant la direction t_k, tangente à l'outil :

(2.29) $$\forall k \in \partial_c \widetilde{\Omega} \qquad R^k . t_k = 0$$

Le résidu est ainsi exprimé dans la base locale définie par la tangente et la normale à l'outil. Cette écriture est généralisée à tous les nœuds du maillage, en considérant :

(2.30) $$\forall k \in \widetilde{\Omega} \qquad Q_k R^k (V) = 0$$

(2.31) $$\forall k \in \partial_c \widetilde{\Omega} \qquad (V_k - V_{out}).n_k = 0$$

où la matrice orthonormée de changement de base Q_k, est définie par :

(2.32) $$Q_k = \begin{cases} \begin{bmatrix} t_k^r & t_k^z \\ n_k^r & n_k^z \end{bmatrix} & si \;\; k \in \partial_c \widetilde{\Omega} \\ I & sinon \end{cases}$$

Nous noterons par la suite Q la matrice diagonale par blocs (2x2), de dimension (2x*Nbnoe*, 2x*Nbnoe*), obtenue par assemblage des matrices locales de rotation Q_k.

2.2.4 Résolution du système

L'équation non linéaire (2.30) est résolue par un algorithme de Newton-Raphson [Cescutti 89]. Le principe consiste à se ramener, à chaque itération *(i)*, à la résolution d'un problème linéaire tangent. On cherche alors une correction $\Delta V^{(i)}$ à apporter à $V^{(i)}$ telle que :

(2.33) $$QR(V^{(i)} + \Delta V^{(i)}) = 0$$

qui est approchée à l'ordre 1 par :

(2.34) $$Q \frac{\partial R}{\partial V}(V^{(i)}) \Delta V^{(i)} = Q \frac{\partial R}{\partial V}(V^{(i)}) Q^T Q \Delta V^{(i)} = -QR(V^{(i)})$$

Ce qui conduit à la résolution du système linéaire :

(2.35) $$\Delta \breve{V}^{(i)} = -\left[\frac{\partial \breve{R}}{\partial V}\right]^{-1} QR \quad \text{avec} \quad \Delta \breve{V} = Q \Delta V \quad \text{et} \quad \frac{\partial \breve{R}}{\partial V} = Q \frac{\partial R}{\partial V} Q^T$$

La résolution de (2.35) est suivie d'une étape de recherche linéaire (interpolation quadratique), afin de minimiser la norme du résidu :

(2. 36) $$\underset{\lambda}{Min}\left\| R(V^{(i)}+\lambda\Delta V^{(i)}\right\|$$

Lors de l'élimination de (2.34), il convient de s'assurer que la deuxième condition de contact unilatéral est respectée, i.e. : que la contrainte normale de contact est compressive :

(2.37) $$(\tilde{\sigma}n).n < 0$$

La résolution du problème de contact peut être présentée de la manière suivante : après calcul du champ de vitesse, on teste[4] la condition (2.37) pour chaque nœud de $\partial_c\tilde{\Omega}$ (en contact). Si cette condition n'est pas vérifiée, le nœud k est relâché et la surface de contact $\partial_c\tilde{\Omega}$ est modifiée en conséquence. Un nouveau champ de vitesse est alors calculé. Cette séquence est répétée itérativement jusqu'à ce que $\partial_c\tilde{\Omega}$ n'évolue plus. La convergence de cet algorithme est assurée par les fortes pressions généralement mises en jeu dans les procédés de forgeage.

2.2.5 Aspect instationnaire

Une fois le champ de vitesse déterminé, il faut actualiser les positions géométriques des nœuds du maillage en utilisant (2.27). On est alors amené à corriger les erreurs dues au schéma explicite d'actualisation. Ces erreurs sont de deux types :

- un nœud pénètre à l'intérieur d'un outil (Figure 2.3)
- un nœud de $\partial_c\tilde{\Omega}$, bien qu'ayant une condition de contact compressive, quitte le contact avec l'outillage (Figure 2.4)

Une solution simple consiste à projeter le nœud orthogonalement sur la surface des outils. Pour éviter la pénétration des nœuds frontaliers à l'intérieur des outils, il est possible d'ajuster le pas de temps Δt, de telle façon qu'au plus un seul nœud, arrive en contact à chaque incrément de calcul.

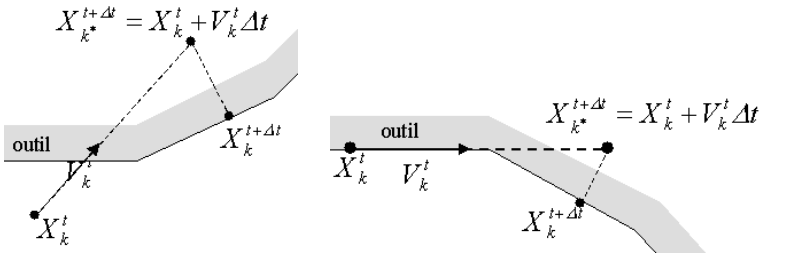


Figure 2.3 : *Pénétration d'un nœud frontalier*

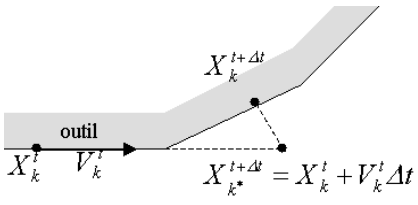


Figure 2.4 : *Perte de contact due au schéma explicite*

[4] La condition examinée dans la pratique ne porte pas sur la distribution de contrainte normale, mais sur la valeur du résidu nodal dans la direction normale qui a le même signe que cette dernière.

2.2.6 Les pertes de volume

L'incompressibilité du matériau est assurée par la contribution du terme R_p dans l'équation d'équilibre (2.26). Le champ de vitesse ainsi déterminé est incompressible au temps t considéré. Toutefois, comme ce champ est supposé constant pendant l'incrément de temps Δt et que l'on utilise une formulation de type Lagrangien réactualisé, la transformation de $\widetilde{\Omega}_t$ à $\widetilde{\Omega}_{t+\Delta t}$ n'est pas incompressible. Les variations de volume générées seront d'autant plus importantes que le pas de temps est grand. Dans le cas d'un écrasement entre tas plats, on montre que la perte de volume est proportionnelle au carré du pas de temps [Bellet 90]. Diminuer suffisamment le pas de temps risquerait de rendre le temps de calcul prohibitif. Il faut, de plus, considérer les variations de volume lors de la projection des nœuds en contact. Selon les conditions, ces variations de volume peuvent atteindre de 1% à 2%. Pour résoudre ce problème, Coupez [Coupez 91] a proposé d'écrire l'équation de conservation de la masse de manière incrémentale. Si on considère un élément de volume ω contenu dans Ω, on cherche le champ de vitesse v, tel qu'après actualisation par un schéma explicite, le volume soit conservé entre les temps t et $t+\Delta t$; soit $\omega_t = \omega_{t+\Delta t}$. La conservation de la masse s'écrit :

$$\forall \omega_t \subset \Omega_t \quad \int_{\omega_t} \rho(x,t)dw = \int_{\omega_{t+\Delta t}} \rho(x,t+\Delta t)dw \tag{2.38}$$

où ρ désigne la masse volumique du matériau. En considérant le jacobien $\bar{J}_t^{t+\Delta t}$ de la transformation de t à $t+\Delta t$:

$$\forall \omega_t \subset \Omega_t \quad \int_{\omega_t} \rho(x,t)dw = \int_{\omega_t} \bar{J}_t^{t+\Delta t} \rho(x,t+\Delta t)dw \tag{2.39}$$

En tenant compte du schéma explicite utilisé, ce jacobien s'écrit

$$\bar{J}_t^{t+\Delta t} = dét\left(\frac{\partial x_{t+\Delta t}}{\partial x_t}\right) = 1 + \mathrm{div} v_t \Delta t + dét\left(grad(v_t)\right)\Delta t^2 \tag{2.40}$$

Le matériau étant incompressible et homogène au cours du temps, $\rho(x,t) = \rho(x,t+\Delta t)$, l'équation (2.39) devient alors :

$$\forall \omega_t \subset \Omega_t \quad \int_{\omega_t} \left(\mathrm{div} v_t + dét\left(grad(v_t)\right)\Delta t\right)dw = 0 \tag{2.41}$$

La non-linéarité de cette équation en v peut poser des problèmes de convergence. De plus, cette formulation ne tient pas compte des variations de volume dues aux projections des nœuds en contact. On préfère donc utiliser une formulation linéarisée de l'incompressibilité incrémentale, en considérant les variations numériques de volume comme une variation locale de la densité numérique du matériau. Cette densité est calculée en écrivant la conservation de la masse entre les instants $t=0$ et t :

(2.42) $$\forall \omega_t \subset \Omega_t \quad \int_{\omega_t} \rho(x,t)dw = \int_{\omega_0} \rho(x,0)dw = \int_{\omega_t} \bar{J}_0^t \rho(x,0)dw$$

En considérant (2.38) et (2.42), on obtient :

(2.43) $$\bar{J}_0^t \rho_0 = \bar{J}_t^{t+\Delta t} \rho_{t+\Delta t}$$

où $\rho_0=\rho(x,0)$ est la densité numérique initiale du matériau. En écrivant maintenant, que la densité à l'instant $t+\Delta t$ doit être égale à 1 et en considérant l'approximation au premier ordre du jacobien $\bar{J}_t^{t+\Delta t}$:

(2.44) $$\bar{J}_0^t \rho_0 = 1 + \mathrm{div} v_t \Delta t \Rightarrow \mathrm{div} v_t = \frac{\bar{J}_0^t \rho_0 - 1}{\Delta t}$$

Cette équation est intégrée sur tous les éléments du maillage :

(2.45) $$\int_{\tilde{\Omega}_e} \mathrm{div}\tilde{v} dw + \frac{1}{\Delta t} \int_{\tilde{\Omega}_e} \left(1 - \bar{J}_0^t \rho_0\right) dw = 0$$

En notant V^e le volume d'un élément actuel e et V_0^e le volume de ce même élément sur la configuration initiale non déformée :

(2.46) $$V^e = \int_{\tilde{\Omega}_e} dw \quad \text{et} \quad V_0^e = \int_{\tilde{\Omega}_e} \bar{J}_0^t dw$$

l'équation (2.45) devient :

(2.47) $$\forall e \qquad \int_{\tilde{\Omega}_e} \mathrm{div}\tilde{v} dw + \frac{1}{\Delta t}\left(V^e - \rho_0 V_0^e\right) = 0$$

Cette équation est ensuite introduite dans la résolution du problème mécanique à travers le terme de pénalisation :

(2.48) $$R_p^k(X,V) = \rho \sum_{e=1}^{Nbelt} \frac{1}{V^e} \left(\int_{\tilde{\Omega}_e} \mathrm{div}\tilde{v} dw + \frac{1}{\Delta t}\left(V^e - \rho_0 V_0^e\right) \right) \int_{\tilde{\Omega}_e} Trace(B^k) dw$$

Pour être complet, il convient de prendre en compte les variations de volume lors des remaillages. En effet, si avant remaillage, un élément grossier pénètre la surface d'un outil (Figure 2.5), des nœuds vont être ajoutés sur la description quadratique des cotés. Ces nouveaux nœuds vont ensuite être projetés sur la surface de l'outil, d'où une variation du volume de la pièce.

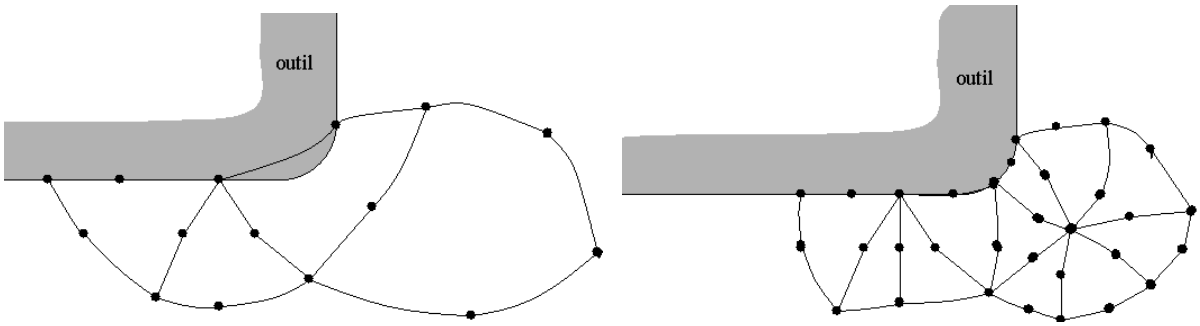


Figure 2.5 : *Pénétration d'un élément avant remaillage.*

Faute de mieux, Miles [Miles 95] propose alors de calculer la densité du nouveau maillage, de telle sorte que la masse reste constante, soit :

$$\rho_0 = \frac{V(\Omega_{initial})}{V(\Omega_{remaillage})} \tag{2.49}$$

Sur le maillage initial, la densité ρ_0 vaut 1. Après chaque remaillage, elle est donnée par (2.49). Les variations de volumes sont rattrapées sur plusieurs incréments. Cette méthode nécessite de stocker les volumes des éléments du maillage non déformé, et d'actualiser ρ_0 après chaque remaillage du domaine.

2.3 Lissage des variables nodales et remaillage

2.3.1 Lissage des variables nodales

Lors de la résolution mécanique, le tenseur des vitesses de déformation $\tilde{\dot{\varepsilon}}$ intervient sous forme intégrale (2.26). La contribution de ce dernier est calculée aux points d'intégration, à partir des vitesses nodales et des gradients des fonctions de base. Néanmoins, le champ $\tilde{\dot{\varepsilon}}$ obtenu après résolution éléments finis est d'un ordre inférieur à l'interpolation en vitesse. Ce champ est ainsi discontinu pour des éléments C^0. Les variables secondaires de la simulation (taille de grain, endommagement,…) sont supposées continues : elles sont interpolées à partir de valeurs nodales. Ceci implique, par exemple, la connaissance des valeurs nodales de $\bar{\varepsilon}$ et par conséquent de $\dot{\varepsilon}$. Il est alors nécessaire d'effectuer un lissage. Plusieurs techniques de lissage sont envisageables [Fourment 92]. La procédure de lissage utilisée dans le logiciel FORGE2® est basée sur la méthode proposée par Liszka et Orkisz [Liszka 80]. L'idée consiste à utiliser le maillage éléments finis pour définir une grille locale de différences finies. Ceci afin de calculer la dérivée d'un champ nodal quelconque *F*. Pour chaque nœud *k* du maillage, on sélectionne un certain nombre de nœuds voisins $\vee(k)$. Pour tout nœud *i* appartenant à ce voisinage, on effectue alors un développement de Taylor à l'ordre 2 du champ *F* [Fourment 92] :

$$\forall i \in \vee(k) \qquad F_i = F_k + gradF_k . \Delta x_{ik} + \frac{1}{2} \Delta x_{ik} . H(F_k) \Delta x_{ik} + O(\Delta x_{ik}^3) \tag{2.50}$$

(2.51) $$\text{avec } \Delta x_{ik} = \begin{bmatrix} R_i - R_k \\ Z_i - Z_k \end{bmatrix} \text{ et } H(F_k) = grad(gradF_k)$$

En notant :

(2.52) $$\Delta F_i^k = F_i - F_k - gradF_k.\Delta x_{ik} - \frac{1}{2}\Delta x_{ik}.H(F_k)\Delta x_{ik}$$

On cherche alors les solutions $grad(F_k)$ et $H(F_k)$ qui minimisent la fonctionnelle π_k sur l'ensemble des voisins de k :

(2.53) $$\pi_k = \sum_{i \in V(k)} \frac{\left(\Delta F_i^k\right)^2}{\left(\Delta x_{ik}\right)^6}$$

Cette minimisation conduit à la résolution d'un système linéaire :

(2.54) $$A(X_k, X_{V(k)}) \begin{bmatrix} grad(F_k) \\ H(F_k) \end{bmatrix} = b(X_k, X_{V(k)}, F_k, F_{V(k)})$$

où A est une matrice 5x5. $F_{V(k)}$ désigne les valeurs de F aux nœuds $X_{V(k)}$ du voisinage. Ce système est résolu par une méthode directe. Le gradient de la vitesse (et par conséquent le tenseur des vitesses de déformation $\dot{\tilde{\varepsilon}}$) est obtenu en considérant, pour un champ F, les composantes V^r et V^z du champ de vitesse. De la même manière, $grad\bar{\bar{\varepsilon}}$ est calculé en prenant la déformation équivalente $\bar{\bar{\varepsilon}}$ pour F.

2.3.2 Remaillage et transport des variables sur le nouveau maillage

2.3.2.1 *Remaillage*

Dans la formulation Lagrangienne réactualisée que nous utilisons, le maillage est considéré comme "matériel", car il suit la matière lors de sa déformation. Ceci implique que la qualité des éléments appartenant à des zones fortement déformées peut se dégrader rapidement. La simulation de la majorité des procédés de forgeage implique donc le recourt à une procédure de remaillage automatique.

Le remailleur automatique 2D de FORGE2® [Coupez 91], permet de générer automatiquement des maillages triangulaires. Cette procédure se décompose en 7 étapes distinctes :

- ♦ surdiscrétisation de la frontière de l'ancien maillage : on ajoute des points supplémentaires aux nœuds frontaliers de l'ancien maillage (au total 7 points de surdicrétisation par côté d'élément frontière : Figure 2.6). Ces points, appartenant au côté quadratique de l'élément, correspondent à une description plus fine de la frontière, et permettent de linéariser la frontière quadratique. Si besoin, ces nouveaux points sont projetés sur l'outillage.

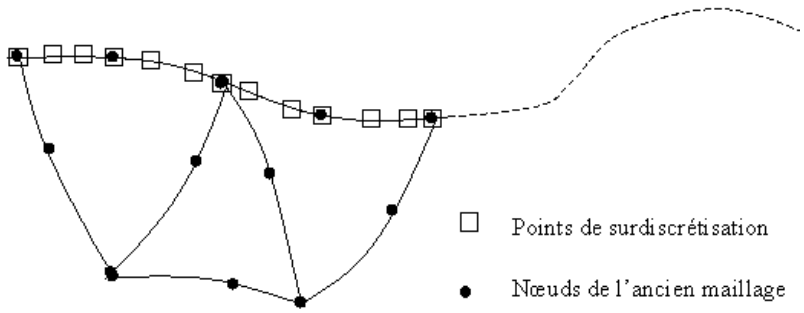


Figure 2.6 : *Surdiscrétisation de la frontière.*

- ♦optimisation de la frontière : on supprime de la description de la frontière certains points de surdiscrétisation. La frontière décrite par les points restants doit vérifier un certain nombre de critères : précision, courbure maximale et homogénéité entre les segments successifs.
- ♦triangulation : on réalise une triangulation du nuage de points frontière ainsi obtenu par un algorithme de Delaunay.
- ♦reconnaissance de la frontière et suppression des triangles extérieurs au domaine à mailler.
- ♦ajout des nœuds internes : on réalise une première ébauche de maillage par une triangulation à l'aide d'éléments à 3 nœuds.
- ♦régularisation et amélioration topologique par inversion des diagonales
- ♦ajout des nœuds au milieu de chaque arête, pour former des éléments triangulaires quadratiques.

Il existe différents critères de déclenchement du remaillage :
- mesure du taux de pénétration du maillage à l'intérieur des outils,
- mesure de la courbure des côtés frontières,
- dégénérescence des éléments ou des côtés frontières (jacobien de la transformation sur l'élément de référence négatif ou nul),
- interpénétration de la matière (défaut de repli : Figure 2.7).

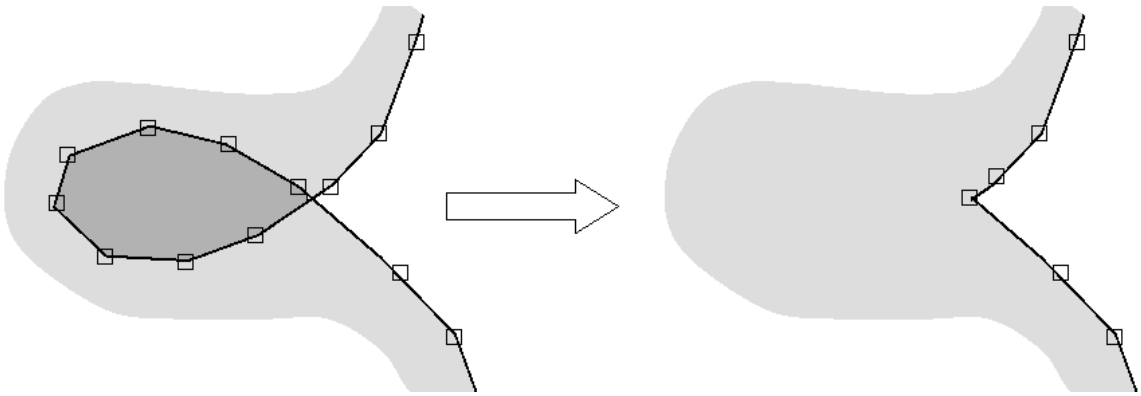


Figure 2.7 : *Suppression d'un repli de matière.*

Dans le dernier cas, il est impératif de modifier la description de la frontière. En effet, la procédure de remaillage ne peut pas générer un maillage à partir d'une frontière se

chevauchant. Ainsi, si l'un des précédents critères de qualité a été violé, ou si le rapport du volume de la boucle de repli au volume de la pièce, est supérieur à une valeur prescrite c_{repli} (2.55), le remaillage est déclenché. Il alors effectué suivant la procédure présentée auparavant.

$$\frac{V(\Omega_{repli})}{V(\Omega_t)} \geq c_{repli} \Rightarrow \textit{déclenchement du remaillage} \tag{2.55}$$

Cette procédure de détection des boucles de repli (ou cycles) teste de manière systématique, pour tout segment, l'existence d'une éventuelle intersection avec un autre segment de la frontière surdiscrétisée. Les points appartenant à la boucle de repli sont éliminés de la description de la frontière. L'élimination de surface de repli introduit une perte de matière. A moins qu'il n'ait lieu dans la bavure, un repli de matière conduit au rejet de la séquence de forgeage. La poursuite des calculs a pour seul objet de permettre l'analyse complète de cette solution pour l'optimisation de forme. L'écoulement solution (sans repli) n'est donc pas influencé par cette approximation.

2.3.2.2 *Transport des variables nodales*

Une fois le nouveau maillage créé, il convient de "transporter", de l'ancien maillage vers le nouveau, les valeurs intégrées dans le temps, et donc dépendantes de l'histoire de la déformation (ex. : la déformation équivalente). Plusieurs méthodes de transport existent afin de résoudre ce problème. On peut par exemple calculer la valeur de la variable à transporter en chaque nœud du nouveau maillage, en l'interpolant au point géométrique qu'il occupait sur l'ancien maillage. Cette méthode, par "inversion de paramètres", quoique peu coûteuse, n'a pas été retenue à cause de son caractère local qui peut aboutir à des champs dégradés [Tronel 93].

Il est possible d'envisager des méthodes plus globales, pour lesquelles on sélectionne, pour tout nœud du nouveau maillage, un certain nombre de voisins topologiques appartenant à l'ancien maillage (Figure 2.8) :

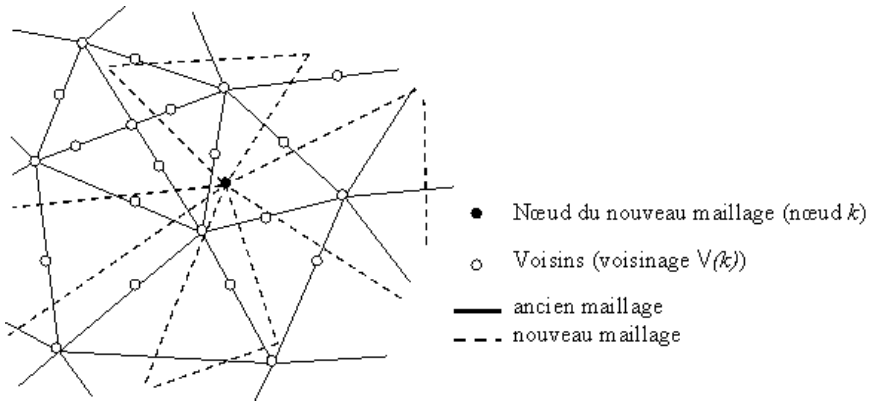


Figure 2.8 : *Sélection des voisins.*

Cette sélection effectuée, une première méthode consiste à calculer la valeur d'un champ quelconque au nœud du nouveau maillage, comme une combinaison linéaire des valeurs nodales de l'ancien maillage. Les coefficients de pondération peuvent être introduits, comme par exemple l'inverse du carré de la distance entre le nœud et son voisin. La méthode est alors

appelée "transport par proximité". Néanmoins, des expériences numériques montrent que la méthode de transport basée sur un lissage de type différences finies s'avère plus précise [Bălan 96]. Le principe est similaire à celui utilisé pour le calcul des vitesses de déformation $\tilde{\dot{\varepsilon}}$ (cf. §2.3.1). Cette fois, on cherche à calculer la valeur du champ *f* au nœud *k* du nouveau maillage (symbolisé par ● sur la Figure 2.8) à partir des valeurs de ce même champ *f* sur le voisinage $\vee(k)$ des nœuds de l'ancien maillage (symbolisés par ○). On calcule alors les solutions f_k, $gradf_k$ et $H(f_k)$, minimisant la fonctionnelle π_k (2.53) sur l'ensemble des voisins $\vee(k)$.

2.4 Algorithme de résolution

L'algorithme de résolution du problème de forgeage n'est abordé ici que de manière schématique et simplifiée, afin de présenter l'ordre dans lequel sont effectuées les opérations. Pour plus de détails, on pourra se référer à [Cescutti 89], ou (en particulier) à [Soyris 90] pour la résolution du problème thermique.

Pour la détermination d'un champ de vitesse solution, on se place à un incrément *t* de calcul. L'équation non-linéaire (2.30), ayant pour conditions aux limites (2.31), est résolue en vitesse par une méthode de Newton-Raphson. L'algorithme de résolution du champ de vitesse peut être résumé de la manière suivante :

Initialisations :

Choix d'un champ de vitesse initial $V^{(0)}$ incompressible (lorsqu'il ne s'agit pas du premier incrément de calcul, on choisit la solution de l'incrément précédent)

Tant que la convergence vers la solution n'est pas réalisée :

- ♦ calculer $QR(V^{(n)})$
- ♦ calculer $\frac{\partial \breve{R}}{\partial V} = Q\frac{\partial R}{\partial V}\left(V^{(n)}\right)Q^T$
- ♦ imposer les conditions (2.31)
- ♦ résoudre le système linéaire : $\Delta\breve{V} = Q\Delta V = -\left[\frac{\partial \breve{R}}{\partial V}\right]^{-1} QR$
- ♦ recherche linéaire : $\underset{\lambda>0}{Min}\left\|QR\left(V^{(n)} + \lambda\Delta V\right)\right\|$
- ♦ $V^{(n+1)} = V^{(n)} + \lambda\Delta V$: évaluer la convergence $\left\|R\left(V^{(n)} + \lambda_{min}\Delta V\right)\right\|$
- ♦ étude des conditions de contact (2.37) : s'il y a des changements, alors

  modification de $\partial_c\tilde{\Omega}$

  convergence non réalisée

  fin étude

fin tant que

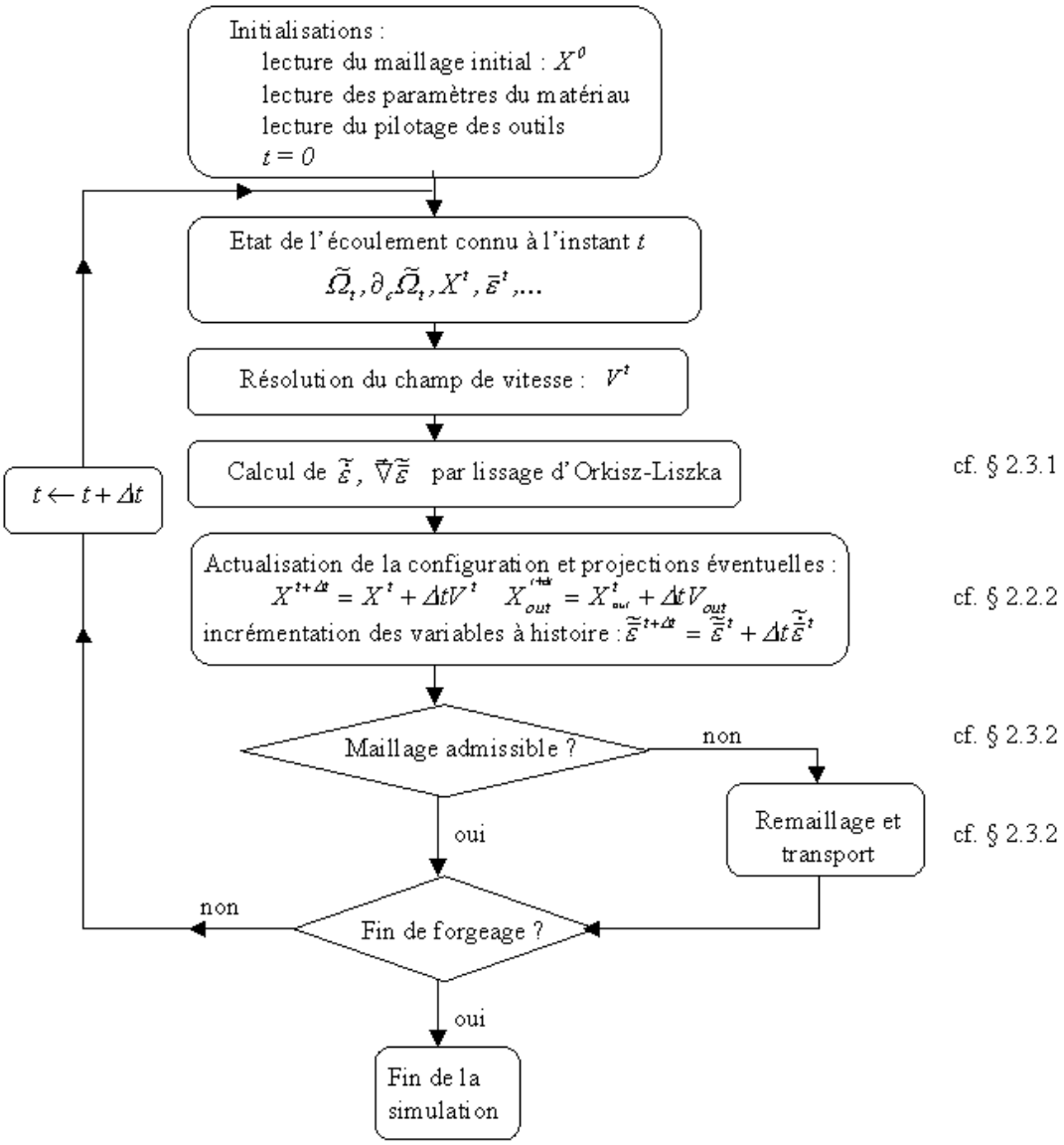


Figure 2.9 : *Processus de résolution du problème de forgeage.*

Chapitre 3

Optimisation de forme en forgeage

Dans la plupart des procédés de forgeage, il est impossible de réaliser la pièce désirée en une seule opération. Le procédé peut en effet s'avérer trop coûteux, voire même impossible, pour de nombreuses raisons : forces nécessaires pour la mise en forme sont trop importantes, forgeage direct provoquant des défauts majeurs tels que l'apparition de replis, etc. Il existe aussi des contraintes d'ordre géométrique, qui rendent souvent plusieurs opérations nécessaires. Par conséquent, le forgeron utilise en général plusieurs passes intermédiaires de forgeage. Bien que l'extension à un plus grand nombre d'opérations ne pose pas de problèmes de principe (sinon de temps de calcul), nous nous limiterons dans cette étude, aux gammes axisymétriques de forgeage en deux passes. La forme des matrices de finition est généralement fixe, déterminée par la géométrie de la pièce à réaliser (Figure 3.1). Nous verrons toutefois (cf. chapitre 5) qu'il est possible d'optimiser la matrice de finition, par exemple pour déterminer la position des freins de bavures. Dans le cas général, si la séquence comprend plusieurs opérations de préformage, on optimisera le dernier outil de préforme. Le lopin de départ est alors issu de la passe précédente. Dans le cas où la gamme ne comprend que deux opérations (préformage et finition), le lopin initial est de forme simple : le plus souvent cylindrique.

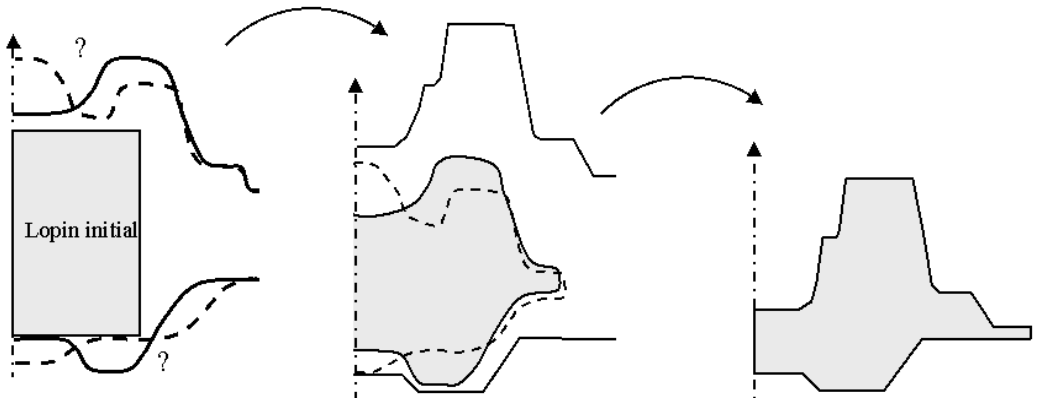


Figure 3.1 : *Problème de conception pour une gamme de forgeage axisymétrique en deux passes.*

Le problème du forgeron est le suivant : déterminer la meilleure géométrie des outils de préforme, permettant d'obtenir une pièce répondant aux exigences de fabrication. Ces dernières sont de natures diverses suivant les domaines industriels. La réalisation de pièces pour le secteur automobile est souvent synonyme de grandes séries et de cadences élevées. Les coûts de fabrication des matrices étant importants, il devient alors primordial, d'un point de vue économique, d'augmenter la durée de vie des outils. A l'inverse, les pièces forgées

destinées à l'industrie aéronautique sont souvent réalisées en petites séries. Il s'agit de pièces de sécurité où la qualité métallurgique est prépondérante. L'éventail des problèmes (et des attentes) est donc vaste, et notre objectif est de développer un logiciel d'aide à la conception qui puisse répondre à tous. Nous présenterons au §3.1 les critères retenus et les formulations mathématiques qui en découlent.

A ce niveau, on peut faire une distinction entre le problème de conception et le problème d'optimisation. Dans le premier cas, on ne dispose que de peu d'idées sur la direction à suivre. De nombreux paramètres sont à déterminer. Pour les formes des matrices intermédiaires, les options envisageables sont très différentes. Elles peuvent être proches de la géométrie de la pièce finale. L'essentiel de la déformation est alors réalisé durant l'opération de préformage. La dernière opération apparaît alors, comme une étape de « finition ». A l'inverse, elles peuvent être très éloignées de la géométrie finale, avec des formes plus douces. La préforme pouvant être vue comme une « ébauche ». Répondre à la question : *quelle préforme utiliser ?* implique donc le choix d'une méthode de description des contours qui permette d'explorer des formes variées avec un nombre réduit de paramètres. La méthode retenue, présentée au §3.2., est basée sur l'utilisation de courbes Bsplines.

Le problème d'optimisation est différent : les lignes directrices de conception sont déjà connues et on cherche à optimiser la gamme, en modifiant légèrement la géométrie de l'outil de préforme, les températures initiales, le type de lubrifiant, etc. Nous verrons (cf. chapitre 5) au travers des quelques problèmes industriels d'optimisation, qu'il est toutefois possible d'aborder ce problème de façon efficace avec les outils que nous avons mis en place.

3.1 Les critères d'optimisation

Parmi tous les critères pouvant être considérés lors de la conception d'une gamme de forgeage, les plus important sont ceux dont la violation entraîne le rejet de la pièce. Dans le cadre de ce travail, nous avons ainsi été amené à étudier :

· deux critères liés à l'écoulement de la matière lors de la mise en forme :
  - le remplissage correct de la matrice de finition
  - l'absence de replis de matière lors des différentes étapes de la déformation
· des critères de qualité métallurgique de la pièce finale :
  - taille de grain homogène ou supérieure à une certaine valeur minimale.
· un critère portant sur le procédé :
  - l'énergie totale de mise en forme

Le dernier critère, proposé par Fourment *et al*. [Fourment 96a], intervient surtout pour régulariser une des fonctions précédentes. Il permet alors de choisir une solution à énergie minimale, parmi les solutions possibles.

L'étude et l'expression sous forme mathématique, des critères d'écoulement, de qualité ou de procédés font l'objet de ce chapitre. Néanmoins, la méthode de calcul et de dérivation mise en œuvre n'est pas restrictive aux fonctions étudiés. D'autres critères pourraient également être considérés.

3.1.1. Remplissage de la matrice finale

L'étude de ce critère a été effectuée par Bălan *et al.* [Bălan 96]. Nous rappelons ici brièvement la définition et le calcul de la fonctionnelle associée. La géométrie de la pièce à forger est décrite à l'aide d'un contour imposé $\partial\Omega_{imp}$. En général, ce contour ne tient pas compte de la bavure mais seulement de la partie utile de la pièce. En fin de procédé, on compare ce contour à la frontière de la pièce forgée $\partial\Omega_{fin}$. La fonction coût remplissage Φ_r est égale au volume axisymétrique séparant les deux contours, c'est à dire au volume non rempli (Figure 3.2) :

(3.1) $$\Phi_r = \int_{\Omega_{prescrit}/\Omega_{fin}} dw$$

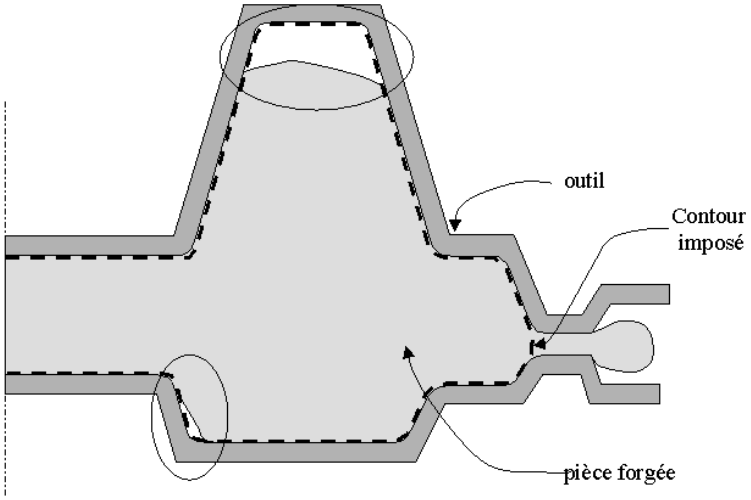


Figure 3.2 : *Définition du critère de remplissage.*

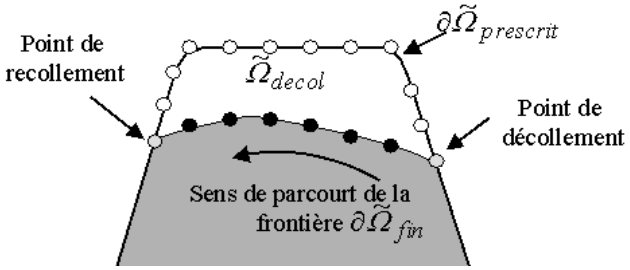


Figure 3.3 : *Calcul du volume non rempli.*

Au niveau discret, les zones non remplies sont obtenues en considérant une surdiscrétisation de la frontière, identique à celle utilisée lors de la procédure de remaillage (cf. §2.3.2). On parcourt cette frontière, en recherchant les points de « décollement » et de « recollement » de la matière sur $\partial\Omega_{prescrit}$ (Figure 3.3). Le volume non rempli est finalement obtenu en appliquant le théorème de la divergence sur $\tilde{\Omega}_{decol}$. La forme discrète s'exprime alors simplement par :

(3.2) $$\tilde{\Phi}_r = \frac{\pi}{3} \sum_{i \in \partial\tilde{\Omega}_{decol}} (Z_{i+1} - Z_i)\left(R_{i+1}^2 + R_i^2 + R_i R_{i+1}\right)$$

Le calcul du critère de remplissage peut aussi être abordé par des méthodes de projection [Zhao 97a]. Néanmoins, ce type de fonction s'avère imprécis si l'écart entre les deux contours est important. Il est préférable d'effectuer un calcul de volume précis, par la méthode (3.2) [Bălan 96].

3.1.2. Energie de mise en forme

Cette fonctionnelle a été introduite par Fourment *et al.* [Fourment 96a] afin de régulariser la fonction coût remplissage. Elle représente l'énergie dépensée pendant les deux passes de forgeage, pour déformer la matière de la configuration Ω_{t_0} à la configuration $\Omega_{t_{fin}}$:

$$\Phi_e = \int_{t=t_0}^{t=t_{fin}} \left(\int_{\Omega_t} K\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^{m+1} dw + \int_{\partial_c \Omega_t} \alpha_f K \left\| \Delta v_t \right\|^{p+1} ds \right) dt \tag{3.3}$$

La forme discrète de (3.3) est donnée par :

$$\tilde{\Phi}_e = \sum_{i=1}^{Nbinc} \left(\int_{\tilde{\Omega}_i} K\left(\sqrt{3}\tilde{\dot{\bar{\varepsilon}}}\right)^{m+1} dw + \int_{\partial_c \tilde{\Omega}_i} \alpha_f K \left\| \Delta \tilde{v}_t \right\|^{p+1} ds \right) \Delta t_i = \sum_{i=1}^{Nbinc} \left(\int_{\tilde{\Omega}_i} \tilde{\Phi}^i_{e1} dw + \int_{\partial_c \tilde{\Omega}_i} \tilde{\Phi}^i_{e2} ds \right) \Delta t_i \tag{3.4}$$

Nbinc désigne le nombre total d'incréments de calcul nécessaires à la simulation et Δt_i les pas de temps de ces incréments.

Il est possible de définir un critère lié à la force maximale de forgeage :

$$\Phi_{force} = \underset{t}{Max} \left\{ \frac{\int_{\Omega_t} K\left(\sqrt{3}\dot{\bar{\varepsilon}}\right)^{m+1} dw + \int_{\partial_c \Omega_t} \alpha_f K \left\| \Delta v_t \right\|^{p+1} ds}{\left\| V_{out,t} \right\|} \right\} \tag{3.5}$$

où $\left\| V_{out,t} \right\|$ désigne la norme de la vitesse de l'outillage à l'instant *t*. Ce critère peut être utilisé pour différentes applications, par exemple si l'on désire optimiser le procédé de manière à forger la pièce désirée sur une presse de faible capacité. Néanmoins, (3.5) n'est pas différentiable. On peut alors considérer, en particulier lorsque la vitesse d'outillage est constante, que le critère d'énergie fournit une bonne approximation de (3.5), comme le montrent les résultats obtenus dans [Bălan 96].

3.1.3 Critères métallurgiques

Afin d'obtenir des pièces possédant de bonnes propriétés mécaniques, par exemple une bonne tenue en fatigue, il est important que la taille de grain soit petite et homogène dans la pièce finale. Des lois d'évolution de la taille de grain, *S*, peuvent être introduites dans le

logiciel de simulation FORGE2® [Kusiak 94, Tronel 93]. La variable *S*, qui dépend de l'histoire thermomécanique de la déformation, est calculée à partir de sa dérivée temporelle $\dot{S}$. Toutefois, ces lois ont souvent un champ d'application restreint (spécifique à un matériau et à des conditions de mise en forme données). Par exemple, dans le cadre du filage de l'aluminium à chaud, Smelser *et al.* [Smelser 86] proposent :

$$\dot{S} = a_1 \exp(a_2 Z^p)\dot{\bar{\varepsilon}}\left(1 - \frac{S}{a_3 Z^r}\right) \tag{3.6}$$

où *Z* désigne le paramètre de Zener-Hollomon : $Z=\dot{\bar{\varepsilon}}\exp(\beta/T)$, a_1, a_2 et a_3 sont des constantes. La valeur de *S* à l'instant *t* du procédé, est donnée par :

$$S^t = \int_{t=t_0}^{t} \dot{S}dt \tag{3.7}$$

La détermination des constantes intervenant dans les lois d'évolution est difficile. A ceci s'ajoutent des problèmes de confidentialité sur l'utilisation de ces valeurs, qui synthétisent toute la connaissance métallurgique acquise par un industriel. En nous appuyant sur la manière de procéder dans l'industrie, et en considérant que les variations de taille de grain sont comparables aux variations de $\bar{\varepsilon}$, nous avons préféré étudier une première approximation de ces critères de qualité, à partir de la déformation totale $\bar{\varepsilon}$. En nous appuyant sur les requêtes des forgerons du secteur aéronautique, on considère deux critères d'optimisation selon que l'on désire obtenir une pièce où la distribution de déformation est homogène, ou que l'on préfère maximiser la valeur des déformations dans certaines parties de la pièce.

3.1.3.1 *Homogénéisation des déformations*

Dans la pratique, afin d'estimer l'homogénéité des déformations sur la pièce forgée, certains forgerons calculent la différence $\bar{\varepsilon}_{max}$ - $\bar{\varepsilon}_{min}$ à partir des valeurs relevées en un certain nombre de points de mesure. Ce critère présente l'inconvénient d'être discontinu et non différentiable. De plus, il s'agit d'une mesure locale qui ne rend pas compte de la répartition de $\bar{\varepsilon}$ sur l'ensemble de la pièce. Afin de définir une mesure globale et continue, Chung *et al.* [Chung 98] préfèrent utiliser la fonctionnelle :

$$\tilde{\Phi} = \frac{1}{Nbelt} \sum_{e=1}^{Nbelt} \left(\bar{\varepsilon}_e - \bar{\varepsilon}_{moy}\right)^2 \tag{3.8}$$

où $\bar{\varepsilon}_{moy}$ désigne la valeur moyenne de $\bar{\varepsilon}$ sur la pièce finale et $\bar{\varepsilon}_e$ la valeur moyenne sur l'élément *e*. Cette formulation ne tient pas compte du caractère axisymétrique de la pièce et semble donc restreinte aux problèmes plans. Une autre forme de cette fonctionnelle a été proposée par Chung *et al.* [Chung 97], pour un problème d'extrusion, en considérant la forme intégrale :

(3.9) $$\Phi = \int_{\Omega_{fin}} \left(\bar{\varepsilon} - \bar{\varepsilon}^*\right)^2 dw \quad avec \quad \bar{\varepsilon}^* = \frac{1}{V_{\Omega_{fin}}} \int_{\Omega_{fin}} \bar{\varepsilon} dw$$

Toutefois, la valeur moyenne $\bar{\varepsilon}^*$ ainsi définie, dépend des paramètres de forme, ce qui risque de rendre le problème de minimisation instable.

Le critère que nous avons retenu repose sur le fait que les zones de fortes variations de $\bar{\varepsilon}$ (Figure 3.4) sont caractérisées par des valeurs élevées du gradient, $grad\bar{\varepsilon}$ (Figure 3.5). On considère la fonction coût suivante :

(3.10) $$\Phi_d^1 = \int_{\Omega_{fin}} \left\| grad\bar{\varepsilon} \right\| dw$$

Le critère ainsi défini a un caractère plus local que (3.9).

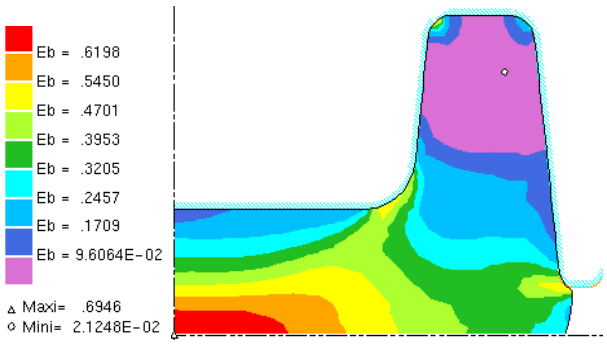


Figure 3.4 : *Isovaleurs de $\bar{\varepsilon}$ sur la pièce forgée*

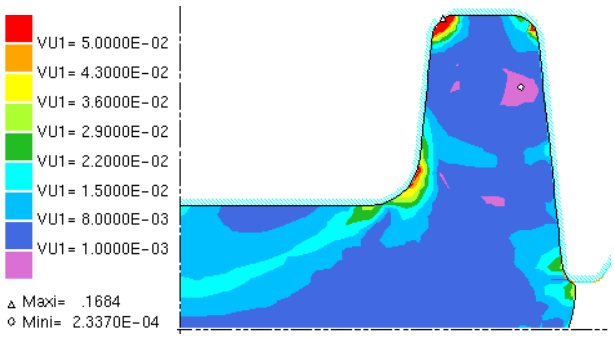


Figure 3.5 : *Isovaleurs de $\left\| grad\bar{\varepsilon} \right\|$*

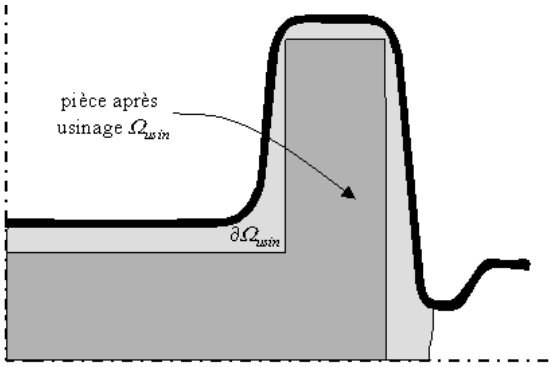


Figure 3.6 : *Pièce après usinage*

Lorsque la pièce est usinée après forgeage (Figure 3.6), l'étude est restreinte à la partie utile Ω_{usin}. La fonction coût est alors donnée par :

(3.11) $$\Phi_d^1 = \int_{\Omega_{usin}} \left\| grad\bar{\varepsilon} \right\| dw$$

Au niveau discret, l'intégration de (3.10) est obtenue à partir de l'interpolation des valeurs nodales de $\bar{\varepsilon}$:

$$\tilde{\bar{\varepsilon}} = \sum_k \bar{\varepsilon}^k N^k \;\Rightarrow\; grad\,\tilde{\bar{\varepsilon}} = \sum_k \bar{\varepsilon}^k\, grad\, N^k \tag{3.12}$$

La restriction (3.11) de l'intégration à Ω_{usin}, peut être calculée en ne prenant en compte que la contribution des points d'intégration appartenant à Ω_{usin}. Ou bien, en utilisant un maillage de la pièce usinée et en effectuant le transport des valeurs nodales $\bar{\varepsilon}$ sur le maillage Ω_{usin} (cf. §2.3.2). Les différences entre ces méthodes sont inférieures à 1%, pour une taille de maille classique, aussi bien pour la fonction que pour sa dérivée.

3.1.3.2 *Maximisations de* $\bar{\varepsilon}$

De faibles valeurs de $\bar{\varepsilon}$ sur la partie utile de la pièce, en fin de procédé, peuvent être la cause du rejet de celle-ci. En effet, elles peuvent traduire une porosité trop importante ou des déformations insuffisantes pour produire la taille de grain désirée. Le critère industriel s'exprime ici sous forme de maximisation :

$$\underset{p}{Max}\left\{ \underset{\Omega_{u\,sin}}{Min}\,\bar{\varepsilon}(p) \right\} \tag{3.13}$$

Un moyen simple de définir un critère d'optimisation différentiable et proportionnel aux faibles valeurs de la déformation, consiste à considérer la fonction coût :

$$\Phi_d^2 = \int_{\Omega_{fin}} f(\bar{\varepsilon})\, dw \tag{3.14}$$

où f est définie par :

$$f(\bar{\varepsilon}) = \begin{cases} 0 & si\;\; \bar{\varepsilon} > \bar{\varepsilon}_2 \\ a\left(\dfrac{1}{(\bar{\varepsilon} - \mu\bar{\varepsilon}_1)} + \dfrac{\bar{\varepsilon}}{(\bar{\varepsilon}_2 - \mu\bar{\varepsilon}_1)^2} - b \right) & si\;\; \bar{\varepsilon} \in [\bar{\varepsilon}_1, \bar{\varepsilon}_2] \\ c\bar{\varepsilon} + d & si\;\; \bar{\varepsilon} < \bar{\varepsilon}_1 \end{cases} \tag{3.15}$$

Les valeurs $\bar{\varepsilon}_1$ et $\bar{\varepsilon}_2$ sont fonction du procédé et fixées par l'utilisateur. $\bar{\varepsilon}_1$ représente la valeur minimale de laquelle on souhaite s'éloigner. $\bar{\varepsilon}_2$ est fonction de la valeur minimale que l'on souhaite atteindre. L'écriture (3.15) provient simplement de l'idée suivante : introduire un terme en $1/\bar{\varepsilon}$ au voisinage de $\bar{\varepsilon}_1$, pour rendre prépondérantes les zones de faibles déformations, tout en conservant une description C^2. Les coefficients a, b, c et d sont choisis tels que $f(\bar{\varepsilon}_1)=1$ et $f(\bar{\varepsilon}_2)=0$ (Figure 3.7). Le coefficient μ détermine la pente de f autour de la valeur critique $\bar{\varepsilon}_1$.

Il est impératif de calculer la fonction coût sur la totalité de la pièce utile. En effet, un calcul restreint aux zones où $\bar{\varepsilon}$ est initialement faible (zones d'étude) n'est pas satisfaisant : au cours des itérations d'optimisation, les zones faiblement déformées peuvent se déplacer significativement.

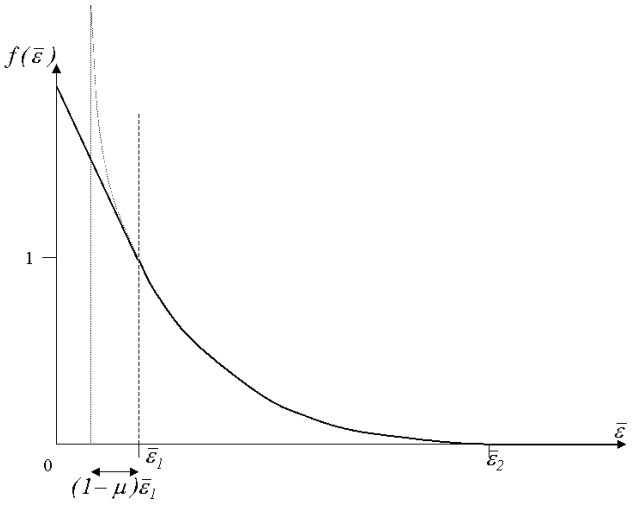


Figure 3.7 : *Graphe de f.*

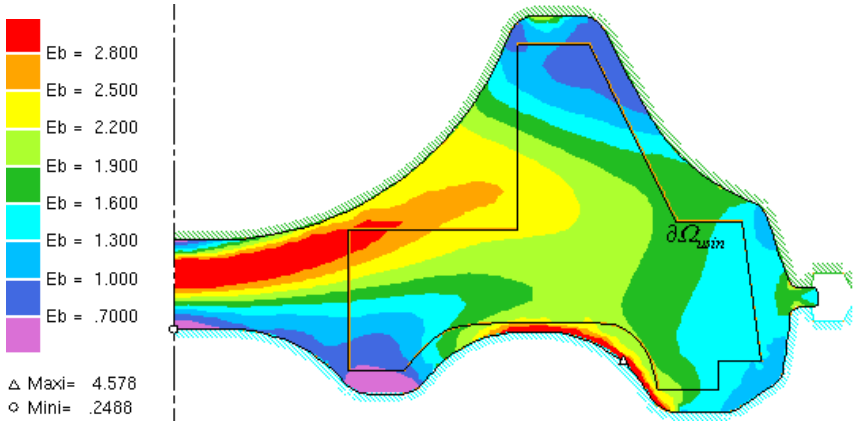


Figure 3.8 : *Isovaleurs de $\bar{\varepsilon}$ sur la pièce forgée.*

La Figure 3.8 donne la répartition de $\bar{\varepsilon}$ sur la pièce finale ainsi que la forme de l'usiné. La carte correspondante de *f* pour $\mu=0.7$, $\bar{\varepsilon}_1=0.7$ (valeur minimale de $\bar{\varepsilon}$ sur Ω_{usin}) et $\bar{\varepsilon}_2=1.9$ est présentée sur la Figure 3.9 :

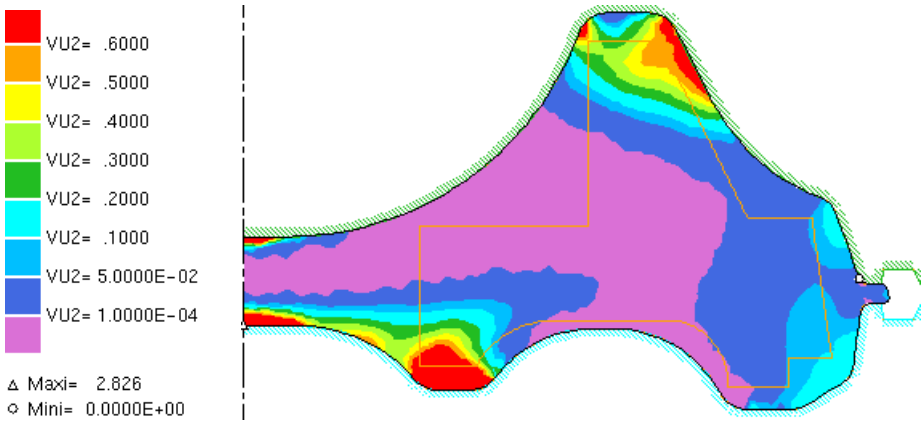


Figure 3.9 : *Isovaleurs de $f(\bar{\varepsilon})$*

Cette fonction permet donc de repérer les zones de faible déformation, tout en prenant en compte le volume des zones critiques.

3.1.4 Elimination des replis de matière

Un repli apparaît lorsque la frontière de Ω_t se chevauche. A la fin du procédé, la fissure ainsi créée représente une zone de faiblesse inacceptable. Ce défaut est synonyme du rebut de la pièce. Arentoft *et al.* [Arentoft 95] ont proposé une table de corrélations entre les différents types de replis et les paramètres, du procédé et du matériau, susceptibles d'influencer ces défauts. Ils montrent que la géométrie des outillages joue un rôle prépondérant.

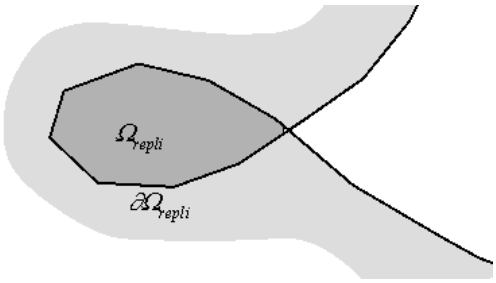


Figure 3.10 : *Défaut de repli.*

La définition d'une fonction coût relative à ce type de défaut est délicate. Elle doit en effet rendre compte du défaut physique et lui être proportionnelle, tout en évitant une trop grande dépendance au maillage ou aux autres paramètres numériques de la discrétisation (pas de temps, fréquence des remaillages,…). Une approche naturelle consiste à définir un critère de type géométrique, en considérant le volume de matière repliée. La durée de vie du défaut (entre l'apparition et la fermeture complète du repli) peut aussi être prise en compte, par exemple en considérant l'intégration dans le temps de ce volume. Malheureusement, notre méthode de traitement des replis (cf. §2.3.2) est incompatible avec un tel critère. En effet, lors d'un remaillage, le cycle de repli est purement et simplement supprimé. La valeur de la fonction coût est alors fonction du nombre et des différents instants de remaillage.

Pour s'affranchir de cette difficulté, on peut préférer suivre l'évolution de la zone de contact matière/matière pendant la déformation. A la fin du procédé, la fonction coût est alors proportionnelle à la surface $S_{défaut}$ du repli (Figure 3.11).

(3.16) $$\Phi_{repli} = S_{défaut}$$

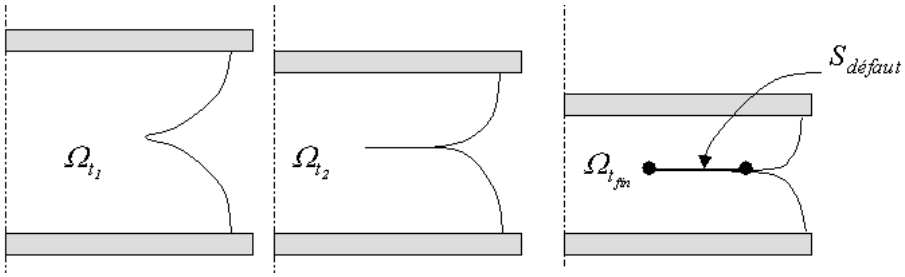


Figure 3.11 : *Suivi de la zone de repli.*

Comme la frontière du domaine est fréquemment remaillée, et que les replis sont éliminés à chaque remaillage, il s'avère nécessaire de calculer le déplacement des points qui

se trouvaient initialement sur la frontière. Pour cela, on suit le déplacement de certains points particuliers (points de marquage) X_{fr} par :

(3.17) $$X_{fr}^{t+\Delta t} = X_{fr}^{t} + \Delta t V_{fr}$$

où les vitesses V_{fr} sont obtenues par interpolation du champ de vitesse. En fin de forgeage, $S_{défaut}$ est calculé à partir des positions des points se trouvant à l'intérieur du domaine. L'absence de tout repli pendant la mise en forme est ainsi caractérisée par $S_{défaut} = 0$.

Ceci ne permet malheureusement pas de conclure que le procédé est correct. Certains paramètres de procédé tels que les conditions de frottement, ont aussi un rôle non négligeable sur l'apparition de ces défauts. Une modification de l'un d'eux peut, en effet, suffire à faire apparaître un défaut de repli. Il convient donc de définir une fonctionnelle qui permette de détecter un repli potentiel, même si ce dernier n'est pas réellement déclaré lors de la simulation.

Nous observons que dans une zone de repli, la matière est soumise à de forts cisaillements. Des valeurs anormalement élevées de $\dot{\bar{\varepsilon}}$ sont calculées, alors même que le repli n'est pas tout à fait formé, ni détecté. Dans le cas présenté sur la Figure 3.12, la vitesse de déformation dans la zone de repli est 10 fois supérieure à la valeur moyenne sur le reste de la surface libre.

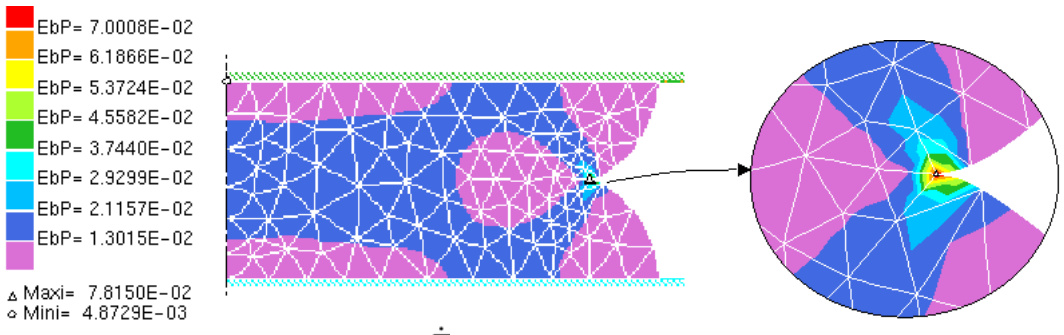


Figure 3.12 : *Isovaleurs de $\dot{\bar{\varepsilon}}$ lors de la formation d'un repli de matière.*

Nous choisissons donc de calculer l'intégrale de la vitesse de déformation sur la surface libre. Cette valeur doit être à son tour intégrée dans le temps, afin de tenir compte de la durée d'existence des défauts potentiels.

(3.18) $$\Phi_{rep} = \int_{t_0}^{t_{fin}} \left(\int_{\partial_1 \Omega_t} \dot{\bar{\varepsilon}} ds \right) dt$$

La forme discrète de (3.18) est donnée par :

(3.19) $$\tilde{\Phi}_{rep} = \sum_{i=1}^{Nbinc} \int_{\partial_1 \Omega_t} \dot{\bar{\varepsilon}} ds \, \Delta t_i$$

L'intégrale (3.19) peut être calculée de deux manières :

- soit en effectuant un calcul aux points d'intégration des côtés à partir des fonctions de

base 1D.

- soit en considérant les valeurs nodales de $\dot{\bar{\varepsilon}}$ obtenues par la méthode de lissage d'Orkisz-Liszka (cf. 2.3.1). L'expression (3.19) est alors approchée en pondérant ces valeurs nodales par les surfaces (axisymétriques) des côtés des éléments adjacents. Il faut noter que si la méthode de lissage est précise et robuste pour les nœuds internes du maillage, le choix des nœuds voisins pour les nœuds frontières en est le point sensible. Ce choix ne tient en effet pas compte de la courbure des côtés frontières et peut aboutir, dans des zones de repli, à des valeurs erronées (par prise en compte de la contribution d'un voisin situé de l'autre côté du repli).

Nous avons donc retenu la première méthode, moins dépendante du maillage. D'un point de vue pratique, cette intégration ne peut être menée classiquement comme une intégrale de bord car le calcul du gradient de la vitesse (dérivée directionnelle dans 3^2) nécessite la contribution des nœuds internes au domaine. On interpole ainsi $\tilde{\dot{\bar{\varepsilon}}}$ sur tout l'élément avec (2.22), mais seule la valeur au bord de la frontière libre est prise en compte dans le calcul de l'intégrale.

Le choix effectué comporte une part d'arbitraire. Des critères plus spécifiques à certains procédés peuvent également être envisagés. Le paragraphe suivant en est une illustration avec l'utilisation d'un critère géométrique.

3.1.5 Cas particulier des défauts de repli : défaut d'aspiration

Considérons le procédé industriel présenté sur la Figure 3.13. Le lopin initial est obtenu par écrasement d'un cylindre entre tas plats. Lors de la passe de finition, ce procédé présente un défaut d'aspiration de matière sur l'axe de symétrie.

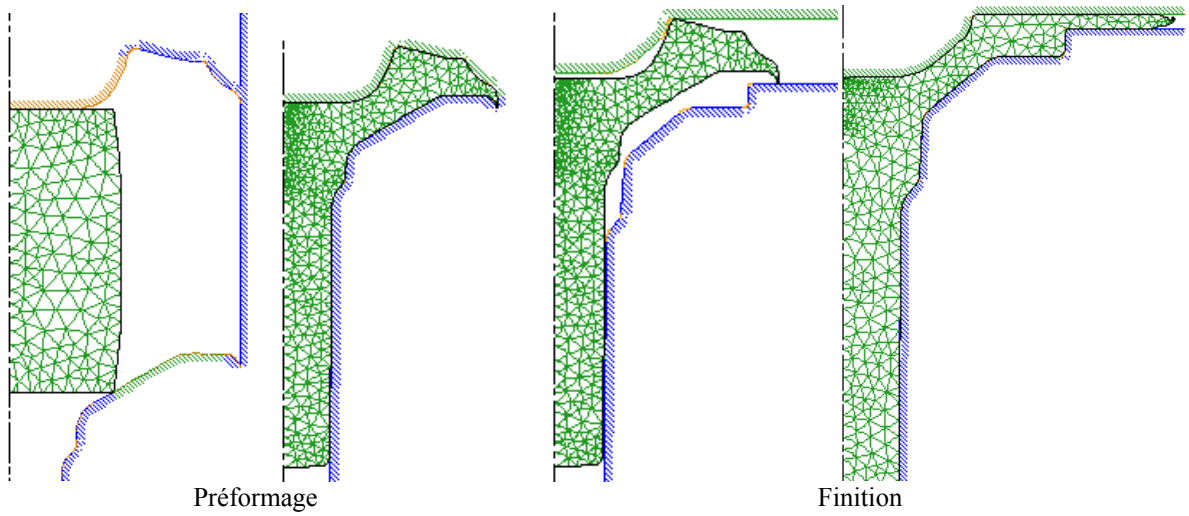


Figure 3.13 : *Forgeage d'une pièce automobile.*

Ce problème peut être abordé indifféremment par une fonction repli ou un critère de remplissage, puisqu'en fin de procédé le défaut d'aspiration se caractérise par une zone non remplie (Figure 3.14). Malheureusement, ce défaut n'a pu être simulé par le logiciel FORGE2. Plusieurs raisons peuvent être avancées, comme une mauvaise modélisation du

frottement, ou le fait que la géométrie finale soit faiblement 3D. Lors de la simulation, on observe toutefois un décollement de la matière sur l'axe (Figure 3.15), mais ce défaut disparaît dans les derniers instants de forgeage. On se trouve ainsi dans la situation précédente : le défaut existe, mais la simulation n'est pas assez précise pour le détecter.

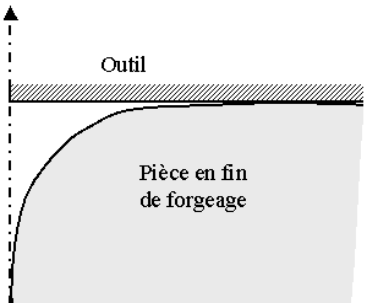


Figure 3.14 : *Défaut d'aspiration en fin de procédé.*

En considérant le phénomène de décollement comme une expression du défaut réel, on peut définir pour fonction coût, l'intégrale temporelle de la distance d_t (sur l'axe de symétrie) entre la matière et l'outillage [Fourment 99] (Figure 3.15) :

(3.20)
$$\Phi_{rep}^{geo} = \int_{t_0}^{t_{fin}} (d_t)dt$$

Une valeur nulle de la fonction coût correspond à un procédé pour lequel le phénomène de décollement a disparu. La préforme optimisée grâce à cette fonction (chapitre 5) est proche de celle mise en œuvre par les forgerons, et déterminée par une optimisation « manuelle » basée sur leur expérience.

Toutefois, sur ce même cas, il est intéressant de noter que le critère (3.18) a donné des résultats similaires. Il nous apparaît plus général et sera donc utilisé par la suite.

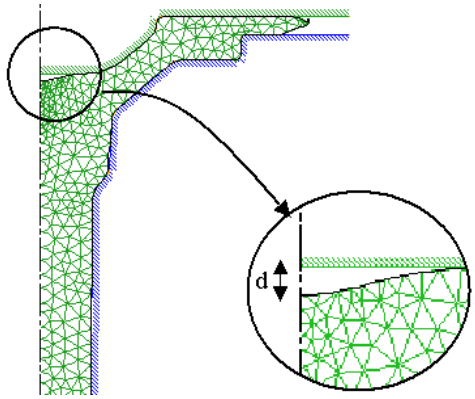


Figure 3.15 : *Défaut et définition d'un critère géométrique*.

3.2 Définition des paramètres d'optimisation

Les géométries des outils de forgeage sont généralement décrites à l'aide de segments de droite et d'arcs de cercle (Figure 3.16). Cette description est, en autre, dictée par des impératifs liés à la réalisation des matrices, par exemple par usinage.

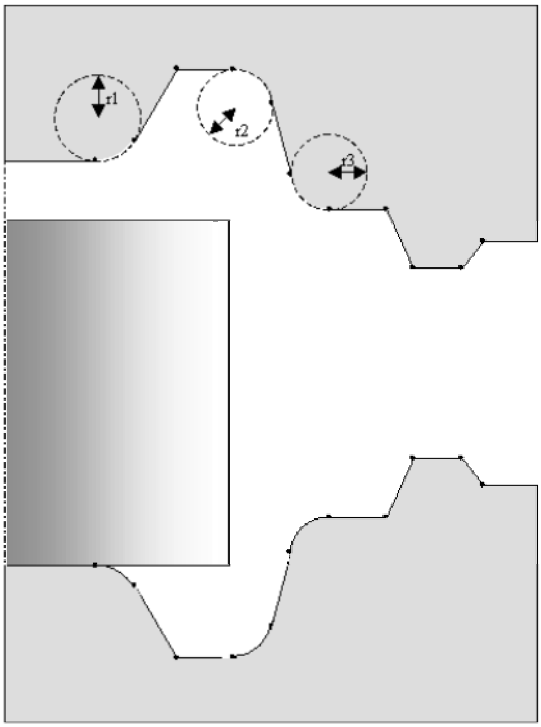


Figure 3.16 : *Description des outils par segments et arcs de cercle.*

En s'inspirant de cette description, une démarche naturelle consiste à retenir les coordonnées des points $X_i = (R_i, Z_i)$ définissant les segments, et les valeurs des rayons r_i définissant les raccordements (Figure 3.17), comme paramètres géométriques d'optimisation.

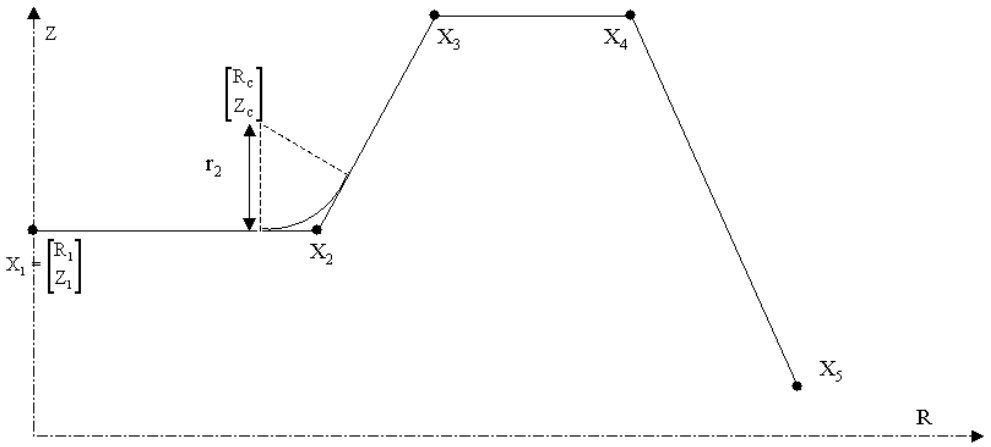


Figure 3.17 : *Paramètres définis à partir des extrémités des segments et des rayons de raccordement.*

Cette méthode est aisément compatible avec les outils classiques de CAO, mais elle présente certains inconvénients. Elle nécessite en effet l'introduction d'un nombre important de restrictions sur la variation des paramètres, afin de conserver une description cohérente. Le respect des conditions géométriques, pour que r_2 (Figure 3.17) puisse être un rayon de raccordement sur les segments $[X_1, X_2]$ et $[X_2, X_3]$, implique par exemple :

$$(3.21) \qquad \begin{aligned} 0 \le (R_c(r_2) - R_1)(R_2 - R_1) + (Z_c(r_2) - Z_1)(Z_2 - Z_1) \le \|X_1X_2\|^2 \\ 0 \le (R_c(r_2) - R_2)(R_3 - R_2) + (Z_c(r_2) - Z_2)(Z_3 - Z_2) \le \|X_2X_3\|^2 \end{aligned}$$

où R_c et Z_c désignent les coordonnées du centre du cercle de rayon r_2, tangent aux segments $[X_1\ X_2]$ et $[X_2\ X_3]$, et sont des fonctions linéaires en r_2. Au niveau de l'algorithme d'optimisation, ces restrictions sont interprétées comme des contraintes inégalitaires (linéaires ou constantes).

De plus, comme nous le verrons au §4.4, l'analyse de sensibilité nécessite d'avoir des surfaces dont la normale soit continue et différentiable. Or, au point de raccord entre un segment et un arc de cercle, la dérivée par rapport aux coordonnées du point sommet (ex. X_2 sur la Figure 3.17) est discontinue. Cette discontinuité provient du passage d'une courbure constante sur l'arc, à une courbure nulle sur le segment. Différents tests numériques ont montré que ces sauts de dérivées se traduisaient par de fortes incertitudes sur le calcul des dérivées globales du problème. La convergence de l'algorithme d'optimisation, devient alors très incertaine.

Les impératifs de différentiation nous imposent donc une description C^1 de la surface de l'outillage à optimiser. Les courbes polynomiales ont été utilisées pour la description d'une filière d'extrusion [Kusiak89, Chen 95]. Bălan *et al.*[Bălan 95] ont proposé d'utiliser des courbes splines pour décrire les contours des outils de forgeage à optimiser. Ces mêmes courbes sont couramment utilisées pour la description des filières d'extrusion 2D [Maniatty 97] ou 3D [Joun 98].

Les splines présentent en effet certains avantages : elles sont continues, d'un ordre de dérivabilité égal au degré d'interpolation, et ne nécessitent que peu de paramètres pour définir des géométries complexes. Elles sont construites à partir d'un ensemble de points caractéristiques définissant une courbe de « référence ». Néanmoins, le point faible des splines réside dans la dépendance globale de la courbe aux positions des points caractéristiques : le déplacement d'un point modifie l'ensemble de la courbe. Cette dépendance peut être source d'instabilités lors de l'optimisation. Afin de remédier à cet inconvénient, Zhao *et al.* [Zhao 97b] et Fourment *et al.* [Fourment 97] (et plus récemment [Chung 98]) ont proposé de décrire les contours d'outils à optimiser à l'aide de courbes Bsplines.

Nous ne présenterons que les équations générales et les points spécifiques à la paramétrisation des contours. Une description détaillée de la construction de ces courbes est présentée en annexe (cf. §A.1).

Le mode de construction est similaire à celui utilisé pour les courbes splines. On définit un contour polygonal comportant *Nbsom* sommets $C_1,\ldots,C_{Nbsom}$ (points caractéristiques). A

chaque sommet C_i correspond une fonction Bspline de base M_i. Tout point de la courbe Bspline d'ordre k est repéré par :

$$X_B(s) = \begin{bmatrix} r_B(s) \\ z_B(s) \end{bmatrix} = \sum_{i=1}^{Nbsom} M_{i,k}(s) C_i = \sum_{i=1}^{Nbsom} M_{i,k}(s) \begin{bmatrix} R_i \\ Z_i \end{bmatrix} \quad s \in [s_{min}, s_{max}] \tag{3.22}$$

Les fonctions de base $M_{i,k}$ sont C^{k-2} polynomiales de degré $k-1$ par morceaux et à support compact. La courbe est décrite en faisant varier l'abscisse s de s_{min} à s_{max}. Nous utiliserons par la suite des courbes Bsplines d'ordre 4 (degré 3), en omettant l'indice k dans (3.22). Ce type de courbe présente la propriété de variation locale : la modification d'un point de contrôle C_i n'influence que la portion de courbe correspondant à l'intervalle où la fonction de base M_i est non nulle.

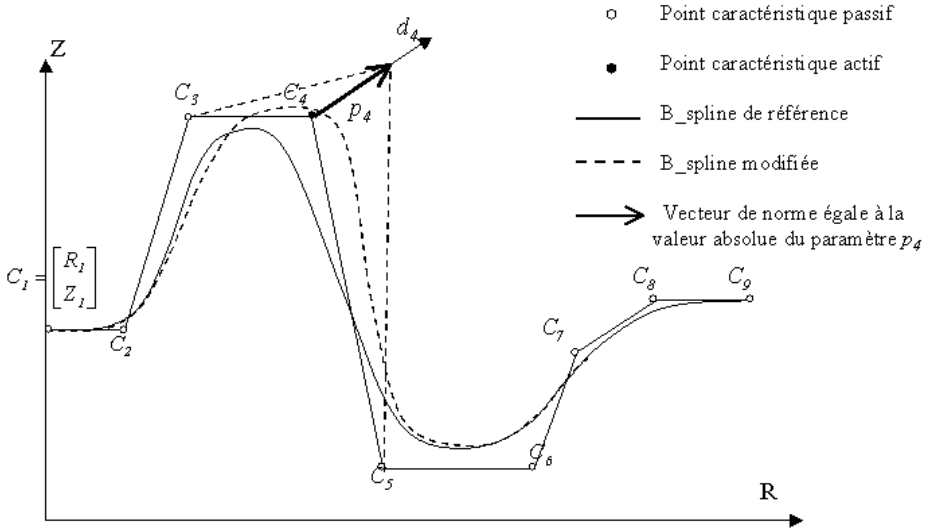


Figure 3.18 : *Définition d'un paramètre suivant une direction arbitraire*.

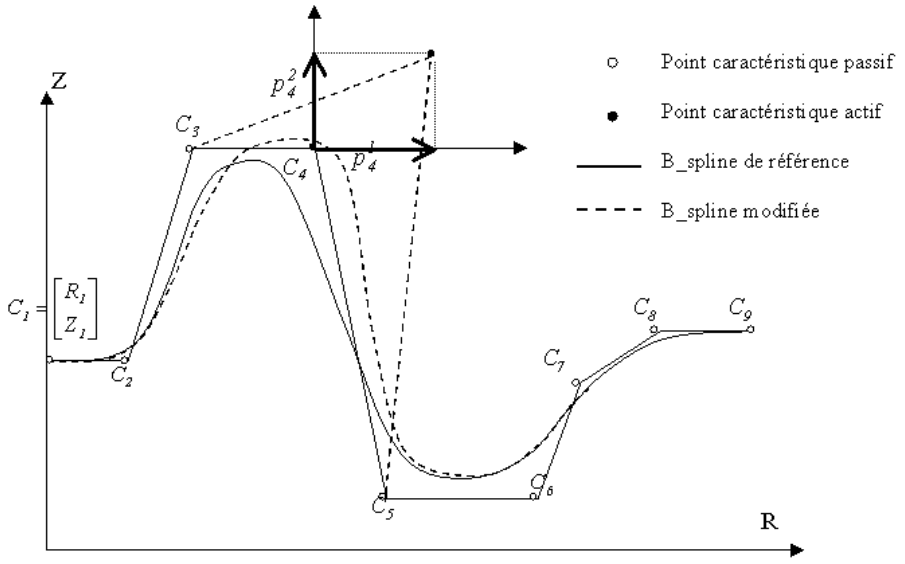


Figure 3.19 : *Point de contrôle actif suivant les deux directions de l'espace.*

La définition des paramètres d'optimisation se fait alors de manière simple et naturelle : à partir d'un contour polygonal initial, on définit une Bspline de référence. Les points caractéristiques « actifs » pourront être déplacés suivant une direction arbitraire *d* (Figure 3.18) ou bien suivant les deux directions de l'espace (Figure 3.19). Les paramètres d'optimisation *p* sont alors les valeurs de ces déplacements, en chaque point actif du polygone.

Cette méthode permet une grande liberté dans la description des formes. Deux cas particuliers (définition de points multiples et définition de points liés à un même paramètre) sont présentés en annexe à titre d'exemples (cf. A.2). Afin de simplifier les notations, nous supposerons par la suite que les points actifs sont mobiles suivant une seule direction arbitraire *d*. Par ailleurs, la tangente à la Bspline au point d'abscisse *s* s'exprime par :

$$(3.23)\quad t(s)=\begin{bmatrix} t_r(s) \\ t_z(s) \end{bmatrix}=\frac{\sum_{i=1}^{Nbsom}\frac{dM_i(s)}{ds}\begin{bmatrix} R_i \\ Z_i \end{bmatrix}}{\sqrt{\left(\sum_{i=1}^{Nbsom}\frac{dM_i(s)}{ds}R_i\right)^2+\left(\sum_{i=1}^{Nbsom}\frac{dM_i(s)}{ds}Z_i\right)^2}} \quad s\in[0,Nbsom-3]$$

La normale est obtenue en considérant le troisième vecteur unitaire 3D, *k* :

$$(3.24)\qquad n(s)=k\wedge t(s)$$

3.3 Algorithmes d'optimisation

3.3.1 Cas général

Le choix d'un algorithme d'optimisation n'est pas chose aisée. Comme nous l'avons vu dans le chapitre 1, il existe de nombreuses méthodes, avec ou sans contrainte. Dans le cas du forgeage, lorsque les maillages sont relativement fins et le nombre d'incréments de calcul élevé, le temps de simulation peut être supérieur à une heure. Chaque calcul de fonction équivaut à une simulation complète du procédé. Le coût de calcul de Φ est donc l'élément déterminant pour le choix d'un algorithme. La méthode retenue doit être robuste et capable d'atteindre un optimum en un nombre réduit de simulations. De plus, le nombre de paramètres ne doit pas avoir une influence trop importante sur son coût. Cette dernière condition permet d'éliminer les algorithmes probabilistes qui semblent, en effet, beaucoup trop gourmands en nombre d'évaluations de la fonction. Lorsque le nombre de paramètres est important, le coût de ces méthodes devient alors exorbitant. En conséquence, il semble plus raisonnable d'utiliser des algorithmes de minimisation d'ordre 1 utilisant le gradient de Φ. En dehors de toute considération sur la précision des dérivées, cette condition permet aussi d'éliminer le calcul des gradients par différences finies.

Dans le cas où la fonction coût n'est soumise à aucune contrainte, nous utiliserons un algorithme de type BFGS. Ce dernier est en effet largement utilisé en optimisation structurale, et semble être un des plus performants et robustes [Haftka 92]. Dans le cas plus général d'une

minimisation sous contraintes, il est plus difficile d'effectuer un choix. Deux méthodes semblent se distinguer [Schittkowski 94] : la méthode du Lagrangien augmenté et l'algorithme de programmation quadratique. Elles ont été programmées et sont l'objet d'un test comparatif dans le chapitre 5.

La recherche linéaire représente souvent la partie la plus coûteuse de l'algorithme d'optimisation (chaque itération de recherche linéaire est aussi coûteuse qu'une itération BFGS). Nous utilisons donc une recherche linéaire de type inexacte. Son caractère « économique », permet en effet de diminuer le nombre d'évaluations, et donc le nombre de simulations. Les critères de convergence sont définis par la règle de Wolfe, qui présente de plus, l'avantage d'assurer la définie positivité de matrice BFGS. Enfin, les dérivées de la fonction coût étant calculées à chaque simulation, il nous a paru intéressant d'utiliser une interpolation de Davidon qui permet d'obtenir une approximation plus précise pour un nombre de simulations inférieur.

La structure du code d'optimisation est schématisée sur la Figure 3.20. A partir d'une valeur initiale du vecteur p des paramètres, le module de paramétrisation génère une description discrète de l'outillage. L'analyse de sensibilité est menée en parallèle à la résolution du problème direct par le logiciel de simulation éléments finis FORGE2®. Les sorties de ce module sont la fonction coût et ses dérivées (ainsi que les contraintes et leurs gradients, le cas échéant). Ces valeurs servent de point de départ pour le calcul d'un nouveau jeu de paramètres.

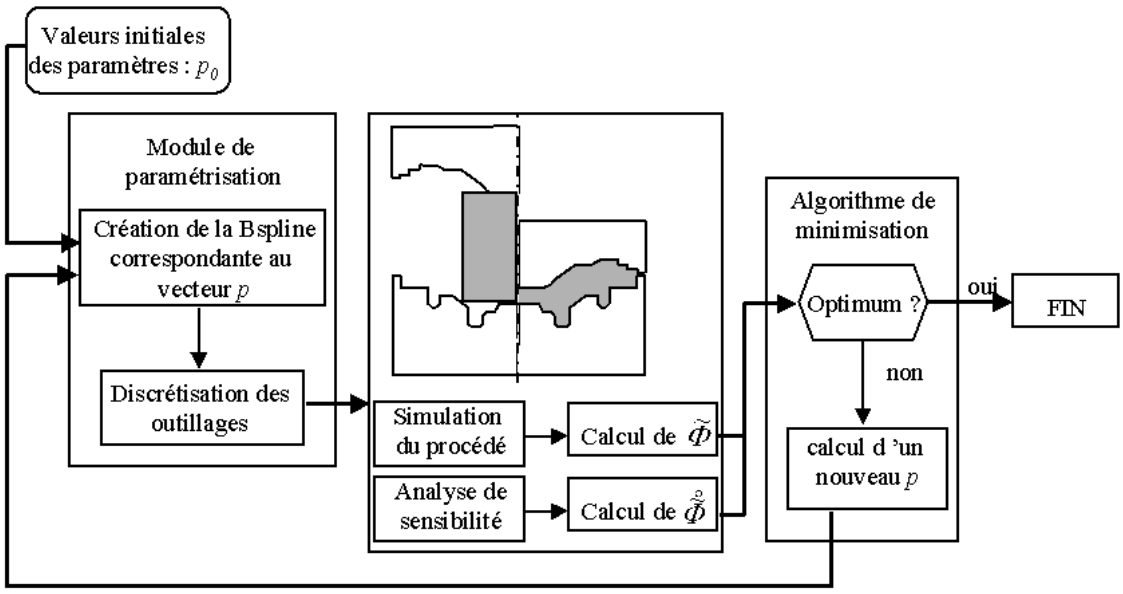


Figure 3.20 : *Structure du code d'optimisation.*

Dans la majorité des problèmes industriels abordés, nous n'avons considéré qu'une seule fonction coût, sans contrainte. Il existe toutefois un cas particulier de contrainte que l'on peut traiter par élimination, pour un procédé très courant : le forgeage en matrices fermées.

3.3.2 Cas particulier du forgeage en matrices fermées

Nous n'avons pas pour l'instant défini de restrictions sur la variation des paramètres de forme. Le déplacement des points de contrôle est donc libre, du moins dans certaines limites fixées pour l'algorithme d'optimisation, en particulier les rayons de recherche linéaire. En conséquence, le volume défini par les matrices en fin de procédé n'est pas constant. Ce type d'optimisation s'applique donc aux procédés en « matrices ouvertes », dont la Figure 3.21 est un exemple. On cherche ici à optimiser certaines parties de l'outil de préforme, dans un procédé comportant deux passes de forgeage. La matrice de finition est le négatif de la pièce à forger, enrichie par des freins de bavure, dans lesquels la matière peut s'écouler librement. Ces bavures jouent un rôle important en forgeage à chaud. Elles permettent en effet d'instaurer en fin de procédé une phase de mise en pression, durant laquelle les détails de la matrice de finition sont remplis.

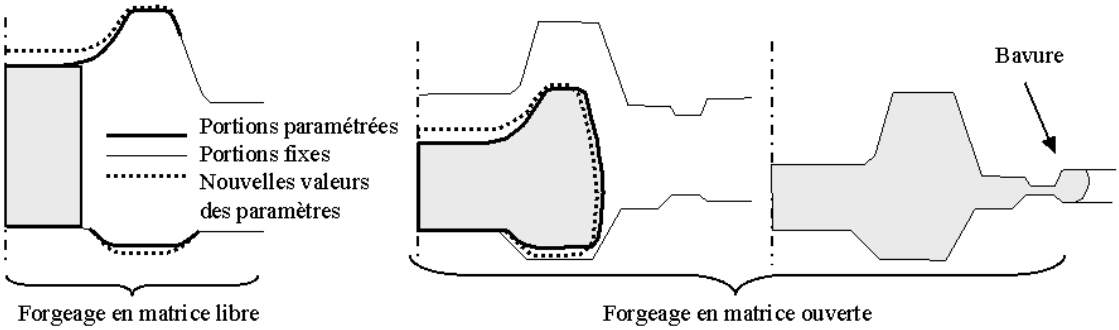


Figure 3.21 : *Optimisation d'une préforme pour un procédé en matrices ouvertes.*

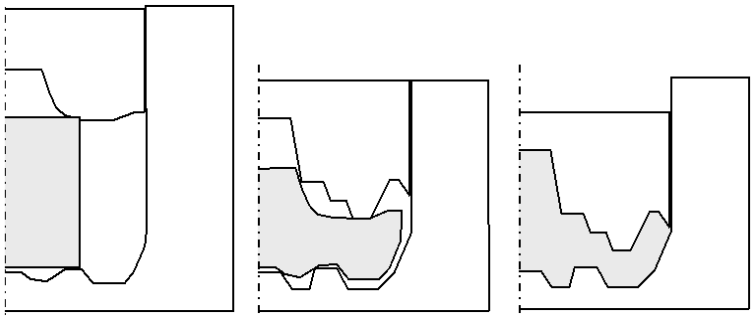


Figure 3.22 : *Procédé en matrices fermées : préformage et finition.*

Pour les procédés de forgeage en « matrices fermées » (Figure 3.22), le volume de toutes les préformes est identique à celui de la pièce. Ce qui implique donc une contrainte sur la forme des outils. Si on considère une des fonctions coût présentées au §3.1, le problème d'optimisation s'écrit sous la forme :

$$\begin{cases} \underset{p\in\mathbb{R}^n}{Min}\ \Phi(p) \\ h(p) = V(p) - V_0 = 0 \end{cases} \tag{3.25}$$

où V_0 désigne le volume du lopin, et $V(p)$ le volume en fin de forgeage, contenu entre les

outils de préforme. La contrainte *h* doit être respectée strictement. En contre partie, pour une valeur donnée de *p*, *h(p)* et *dh(p)/dp* sont faciles à calculer, et ne nécessitent aucune simulation du procédé. Ceci nous conduit à utiliser un algorithme du gradient réduit généralisé (cf. §1.2.2). Il n'y a ici qu'une contrainte, la base des variables dépendantes est donc de dimension 1. Pour simplifier, on choisit une composante particulière du vecteur *p* (on notera *i* cette composante). Sur l'espace des *n-1* variables indépendantes, la $k^{ième}$ composante de la projection de $\tilde{\tilde{\Phi}}$ est donnée par :

$$\left(\overset{\circ}{\Phi}_{(n-1)}\right)_k = \frac{d\Phi}{dp_k} - \frac{d\Phi}{dp_i}\left(\frac{dh}{dp_i}\right)^{-1}\left(\overset{\circ}{h}_{(n-1)}\right)_k \qquad \forall k \neq i \tag{3.26}$$

Le problème est ainsi résolu suivant les *n-1* composantes libres, par une méthode de minimisation sans contrainte. On en déduit alors une direction de descente $d_{(n-1)}$, suivant laquelle on minimise Φ par une technique de recherche linéaire. Le nouveau vecteur $p_{(n-1)}$ ainsi déterminé, est complété en recherchant la $i^{ième}$ composante p_i du vecteur *p* telle que :

$$h(p_{(n-1)}, p_i) = V(p_{(n-1)}, p_i) - V_0 = 0 \tag{3.27}$$

Cette équation non linéaire à une variable est résolue par une méthode de Newton.

Chapitre 4

Analyse de sensibilité

4.1 Dérivation des fonctions coût

On se propose de calculer les dérivées des fonctions coût du problème d'optimisation. On définit dans un premier temps Φ comme une fonction quelconque des variables d'état du problème mécanique :

$$\Phi = \int_{t_0}^{t_{fin}} \Phi_t dt = \int_{t_0}^{t_{fin}} \left(\int_{\Omega_t} \varphi_1(x, v, \dot{\varepsilon}, \bar{\varepsilon}, \sigma, \ldots) dw + \int_{\partial\Omega_t} \varphi_2(x, v, \dot{\varepsilon}, \bar{\varepsilon}, \sigma, \ldots) ds \right) dt \tag{4.1}$$

On se place ici dans le cadre général où la fonction est une intégrale temporelle, calculée tout au long de la simulation (par exemple, la fonction énergie (3.3)). Le cas d'une fonctionnelle calculée à un instant donné du procédé (par exemple la fonction remplissage qui n'est calculée qu'à la fin du procédé) est un cas particulier de (4.1).

Compte tenu de la formulation en Lagrangien réactualisé et de la loi de comportement utilisée, les grandeurs utilisées dans (4.1) s'expriment, au niveau discret, en fonction des seules variables nodales principales X, V et $\bar{\varepsilon}$. La forme discrète de (4.1) est donnée par :

$$\tilde{\Phi} = \sum_{i=1}^{Nbinc} \Delta t_i \tilde{\Phi}^i = \sum_{i=1}^{Nbinc} \Delta t_i \tilde{\Phi}^i(p, X_i, V_i, \bar{\varepsilon}_i) \tag{4.2}$$

où les contributions de chaque incrément de temps sont définies par :

$$\tilde{\Phi}^i = \int_{\tilde{\Omega}_{t_i}} \tilde{\varphi}_1(p, X_i, V_i, \bar{\varepsilon}_i) dw + \int_{\partial\tilde{\Omega}_{t_t}} \tilde{\varphi}_2(p, X_i, V_i, \bar{\varepsilon}_i) ds \tag{4.3}$$

Le vecteur paramètre p intervient explicitement dans (4.3) lorsqu'une des composantes de celui-ci définit un paramètre du procédé (vitesse des outils, coefficient de frottement, …). L'intégrale sur la configuration du maillage $\tilde{\Omega}_{t_i}$ à l'instant t_i, est calculée de manière habituelle, en effectuant un changement de variable entre l'espace réel et l'espace des éléments de référence (1D ou 2D) :

$$\tilde{\Phi}_i = \int_{\tilde{\Omega}_{ref}} \tilde{\varphi}_1 2\pi \tilde{r} \bar{J} dw_{ref} + \int_{\partial\tilde{\Omega}_{ref}} \tilde{\varphi}_2 2\pi \tilde{r} \bar{J}^s ds_{ref} \tag{4.4}$$

(4.5) $$dw_{ref} = d\xi d\eta \ , \ \ ds_{ref} = d\xi$$

(4.6) $$J = \begin{bmatrix} \frac{1}{2\tilde{r}} \sum_k R_k^2 \frac{\partial N^k}{\partial \xi} & \frac{1}{2\tilde{r}} \sum_k R_k^2 \frac{\partial N^k}{\partial \eta} \\ \sum_k Z_k \frac{\partial N^k}{\partial \xi} & \sum_k Z_k \frac{\partial N^k}{\partial \eta} \end{bmatrix} \quad \text{et} \quad J^s = \begin{bmatrix} \frac{1}{2\tilde{r}} \sum_k R_k^2 \frac{\partial N^k}{\partial \xi} \\ \sum_k Z_k \frac{\partial N^k}{\partial \xi} \end{bmatrix}$$

où $\bar{J}$ (respectivement $\bar{J}^s$) désigne le déterminant du jacobien J de passage des éléments de référence aux éléments réels 2D (respectivement la norme du jacobien 1D). Cette configuration étant indépendante des paramètres d'optimisation, les dérivations sont effectuées sur celle-ci.
En dérivant les expressions de chaque contribution de (4.2) on obtient :

(4.7) $$\overset{\circ}{\tilde{\Phi}}^i = \frac{d\tilde{\Phi}^i}{dp} = \frac{\partial \tilde{\Phi}^i}{\partial p} + \frac{\partial \tilde{\Phi}^i}{\partial \bar{\varepsilon}} \frac{d\bar{\varepsilon}}{dp} + \frac{\partial \tilde{\Phi}^i}{\partial X_i} \frac{dX_i}{dp} + \frac{\partial \tilde{\Phi}^i}{\partial V_i} \frac{dV_i}{dp} = \tilde{\Phi}^i_{,p} + \tilde{\Phi}^i_{,\bar{\varepsilon}} \overset{\circ}{\bar{\varepsilon}} + \tilde{\Phi}^i_{,X} \overset{\circ}{X}_i + \tilde{\Phi}^i_{,V} \overset{\circ}{V}_i$$

Les sensibilités $\overset{\circ}{V}$, $\overset{\circ}{X}$ et $\overset{\circ}{\bar{\varepsilon}}$ sont calculées de façon incrémentale, par différentiation directe des équations régissant le problème direct (cf. §2.2.4). Les termes aux dérivées partielles $\partial\tilde{\Phi}^i/\partial\bar{\varepsilon}$, $\partial\tilde{\Phi}^i/\partial X$, $\partial\tilde{\Phi}^i/\partial p$ et $\partial\tilde{\Phi}^i/\partial V$, sont calculés directement en dérivant les expressions définissant les fonctions coût utilisées, à partir de (4.3). Ces expressions peuvent être calculées, ou directement codées en utilisant un logiciel de différentiation automatique [Rostaing 93]. Nous présentons ici le calcul de ces dérivées pour les fonctions décrites au §3.1.

4.1.1 Fonction coût remplissage

La fonction remplissage (3.2) étant calculée sur le maillage en fin de simulation, elle ne dépend que des variables nodales au dernier incrément :

$$\frac{\partial \tilde{\Phi}_r}{\partial \bar{\varepsilon}} = 0 \quad \frac{\partial \tilde{\Phi}_r}{\partial V} = 0 \quad \frac{\partial \tilde{\Phi}_r}{\partial p} = 0$$

(4.8)

$$\frac{\partial \tilde{\Phi}_r}{\partial X_j} = \frac{\pi}{3} \begin{bmatrix} \left(Z_{j+1} - Z_j\right)\left(2R_j + R_j R_{j+1}\right) + \left(Z_j - Z_{j-1}\right)\left(2R_j + R_j R_{j-1}\right) \\ R_{j-1}^2 + R_{j-1} R_j - R_{j+1}^2 - R_j R_{j+1} \end{bmatrix}$$

où l'indice j décrit les points appartenant à la frontière du maillage, pour lesquels la distance au contour prescrit est non nulle (Figure 3.3).

4.1.2 Fonctions coût liées aux déformations

Il s'agit là encore d'une fonction ne dépendant que des variables nodales au dernier incrément de calcul. La fonction coût d'homogénéisation (3.11) s'écrit :

(4.9) $$\tilde{\Phi}_d^1 = \int_{\tilde{\Omega}_{ref}} \left\| grad\tilde{\bar{\varepsilon}} \right\| 2\pi \tilde{r} \bar{J} dw_{ref}$$

Compte tenu de l'interpolation des valeurs nodales de $\bar{\varepsilon}$:

(4.10) $$\tilde{\bar{\varepsilon}} = \sum_k \bar{\varepsilon}^k N^k$$

on obtient le gradient de la déformation totale :

(4.11) $$\frac{\partial \tilde{\bar{\varepsilon}}}{\partial \tilde{r}} = \sum_k \bar{\varepsilon}^k \frac{\partial N^k}{\partial \tilde{r}} \quad \text{et} \quad \frac{\partial \tilde{\bar{\varepsilon}}}{\partial \tilde{z}} = \sum_k \bar{\varepsilon}^k \frac{\partial N^k}{\partial \tilde{z}}$$

Les dérivées de la fonction coût par rapport à *X*, *V* et $\bar{\varepsilon}$ s'en déduisent :

(4.12) $$\frac{\partial \tilde{\Phi}_d^1}{\partial X} = \int_{\tilde{\Omega}_{ref}} \left(\frac{grad\tilde{\bar{\varepsilon}}}{\left\| grad\tilde{\bar{\varepsilon}} \right\|} . \frac{\partial grad\tilde{\bar{\varepsilon}}}{\partial X} \tilde{r} \bar{J} + \left\| grad\tilde{\bar{\varepsilon}} \right\| \frac{\partial \tilde{r} \bar{J}}{\partial X} \right) 2\pi dw_{ref}$$

(4.13) $$\frac{\partial \tilde{\Phi}_d^1}{\partial \bar{\varepsilon}} = \int_{\tilde{\Omega}_{ref}} \frac{grad\tilde{\bar{\varepsilon}}}{\left\| grad\tilde{\bar{\varepsilon}} \right\|} . \frac{\partial grad\tilde{\bar{\varepsilon}}}{\partial \bar{\boldsymbol{\varepsilon}}} 2\pi \tilde{r} \bar{J} dw_{ref} \quad \text{et} \quad \frac{\partial \tilde{\Phi}_d^1}{\partial V} = 0$$

avec

(4.14) $$\frac{\partial \frac{\partial \tilde{\bar{\varepsilon}}}{\partial \tilde{r}}}{\partial X} = \sum_k \bar{\varepsilon}^k \frac{\partial \frac{\partial N^k}{\partial \tilde{r}}}{\partial X} \quad , \quad \frac{\partial \frac{\partial \tilde{\bar{\varepsilon}}}{\partial \tilde{z}}}{\partial X} = \sum_k \bar{\varepsilon}^k \frac{\partial \frac{\partial N^k}{\partial \tilde{z}}}{\partial X}$$

(4.15) $$\frac{\partial \frac{\partial \tilde{\bar{\varepsilon}}}{\partial \tilde{r}}}{\partial \bar{\varepsilon}} = \sum_k \frac{\partial \bar{\varepsilon}^k}{\partial \bar{\varepsilon}} \frac{\partial N^k}{\partial \tilde{r}} \quad \text{et} \quad \frac{\partial \frac{\partial \tilde{\bar{\varepsilon}}}{\partial \tilde{z}}}{\partial \bar{\varepsilon}} = \sum_k \frac{\partial \bar{\varepsilon}^k}{\partial \bar{\varepsilon}} \frac{\partial N^k}{\partial \tilde{z}}$$

Le calcul des gradients des fonctions de base sur l'espace réel et leurs dérivées par rapport à *X*, ainsi que les dérivées des termes $2\pi\tilde{r}\bar{J}$ (ou $\bar{J}^s$) du changement de base, sont présentés plus en détail en annexe (cf. §A.3 et §A.4).

La dérivation de la deuxième fonction liée aux déformations est obtenue de manière similaire :

(4.16) $$\frac{\partial \tilde{\Phi}_d^2}{\partial X} = \int_{\tilde{\Omega}_{ref}} f(\bar{\varepsilon}) \frac{\partial \tilde{r} \bar{J}}{\partial X} 2\pi dw_{ref} \quad , \quad \frac{\partial \tilde{\Phi}_d^2}{\partial \bar{\varepsilon}} = \int_{\tilde{\Omega}_{ref}} f'(\tilde{\bar{\varepsilon}}) \frac{\partial \tilde{\bar{\varepsilon}}}{\partial \bar{\varepsilon}} 2\pi \tilde{r} \bar{J} dw_{ref} \quad \text{et} \quad \frac{\partial \tilde{\Phi}_d^2}{\partial V} = 0$$

4.1.3 Fonction coût énergie

Les contributions à la fonction coût énergie sont calculées à chaque incrément de temps. Pour alléger les notations, nous ne reprendrons que les termes $\tilde{\Phi}_{e1}$ et $\tilde{\Phi}_{e1}$, définis par (3.4) et ne faisant pas intervenir les termes de changement de base. On différentie alors chaque terme par rapport aux variables principales :

$$(4.17) \qquad \frac{\partial \tilde{\Phi}^i_{e1}}{\partial \bar{\varepsilon}} = \frac{\partial \tilde{K}}{\partial \bar{\varepsilon}} \left(\sqrt{3}\bar{\tilde{\varepsilon}}\right)^{m+1} \quad ; \quad \frac{\partial \tilde{\Phi}^i_{e2}}{\partial \bar{\varepsilon}} = \alpha_f \frac{\partial \tilde{K}}{\partial \bar{\varepsilon}} \left\| \Delta \tilde{v}_t \right\|^{p+1}$$

$$(4.18) \qquad \frac{\partial \tilde{\Phi}^i_{e1}}{\partial X} = \tilde{K}(m+1)\left(\sqrt{3}\bar{\tilde{\varepsilon}}\right)^{m-1} \tilde{\varepsilon} : \frac{\partial \tilde{\varepsilon}}{\partial X} \quad ; \quad \frac{\partial \tilde{\Phi}^i_{e2}}{\partial X} = \alpha_f \tilde{K}(p_f+1)\left\| \Delta \tilde{v}_t \right\|^{p-1} \Delta \tilde{v}_g . \frac{\partial \Delta \tilde{v}_g}{\partial X}$$

$$(4.19) \qquad \frac{\partial \tilde{\Phi}^i_{e1}}{\partial V} = \tilde{K}(m+1)\left(\sqrt{3}\bar{\tilde{\varepsilon}}\right)^{m-1} \tilde{\varepsilon} : B^k \quad ; \quad \frac{\partial \tilde{\Phi}^i_{e2}}{\partial V} = \alpha_f \tilde{K}(p_f+1)\left\| \Delta \tilde{v}_t \right\|^{p-1} \Delta \tilde{v}_g . \frac{\partial \Delta \tilde{v}_g}{\partial V}$$

La dépendance de la consistance K du matériau aux conditions thermomécaniques peut être de différentes natures [Gavrus 96]. Nous utilisons une loi d'écrouissage puissance associée à une loi d'Arrhénius en température (2.7) (supposée constante).
Les termes (4.17) sont alors directement calculés par différentiation de la forme discrète de (2.7) :

$$(4.20) \qquad \frac{\partial \tilde{K}}{\partial \bar{\varepsilon}} = \sum_k K_0 n \left(\bar{\tilde{\varepsilon}} + \bar{\varepsilon}_0\right)^{n-1} e^{\beta / T} \frac{\partial \bar{\tilde{\varepsilon}}}{\partial \bar{\varepsilon}} N^k$$

Le calcul des dérivées partielles intervenant dans (4.18) et (4.19) est effectué [Bălan 96] à partir des formules de discrétisation éléments finis (2.21)-(2.23), et des conditions aux limites (2.15). La prise en compte de l'influence de la déformation dans la sensibilité de $\tilde{\Phi}_e$ s'avère indispensable. La Figure 4.1 montre l'évolution de $\mathring{\tilde{\Phi}}_e$ dans le temps, lorsque l'on calcule ou non le terme $\partial \tilde{\Phi}_e / \partial \bar{\varepsilon}$. Le cas étudié (maillage, rhéologie et définition des paramètres) est présenté §5.2. En comparant les valeurs analytiques avec un calcul par différences finies, on voit que l'omission de ce terme peut entraîner de très fortes imprécisions dans le calcul du gradient de la fonctionnelle (20% contre 3% dans le cas où ce terme est calculé).

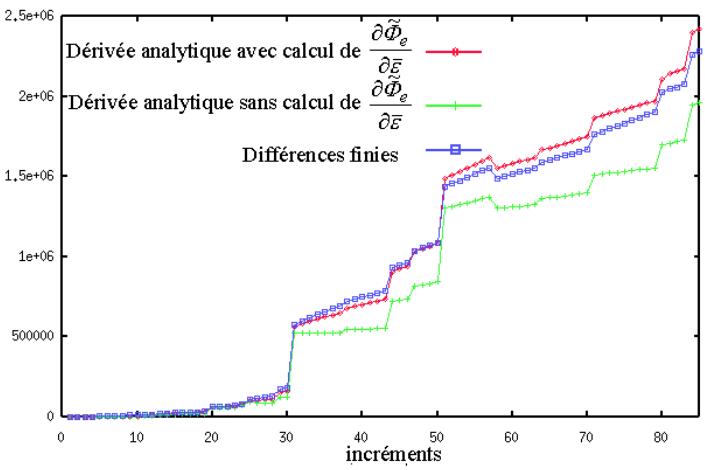


Figure 4.1 : *Evolution de $\mathring{\tilde{\Phi}}_e$ dans le temps. Comparaison entre les courbes calculées analytiquement avec ou sans contribution de ć$\tilde{\Phi}_e$/ć$\bar{\varepsilon}$ et la courbe calculée par différences finies. (n=0.2 et $\bar{\varepsilon}_0$=0)*

4.1.4 Fonction coût repli

A chaque instant *i*, la contribution à la fonction coût repli est égale à l'intégrale de $\dot{\bar{\varepsilon}}$ sur la frontière libre du domaine (3.19). Cette fonction ne dépend que des variables instantanées nodales X_i et V_i au même instant (les indices de temps seront omis par la suite). Les dérivées s'écrivent directement :

$$(4.21)\ \frac{\partial \tilde{\Phi}^i_{rep}}{\partial X} = \int_{\partial_i \tilde{\Omega}_{ref}} \left(\frac{\partial \tilde{\dot{\bar{\varepsilon}}}}{\partial X} \tilde{r} \bar{J}^s + \tilde{\dot{\bar{\varepsilon}}} \frac{d\left(\tilde{r}\bar{J}^s\right)}{dX} \right) 2\pi ds_{ref}\ ,\ \frac{\partial \tilde{\Phi}^i_{rep}}{\partial V} = \int_{\partial_i \tilde{\Omega}_{ref}} \frac{\partial \tilde{\dot{\bar{\varepsilon}}}}{\partial V} 2\pi \tilde{r} \bar{J}^s ds_{ref} \text{ et } \frac{\partial \tilde{\Phi}^i_{rep}}{\partial \bar{\varepsilon}} = 0$$

En remarquant :

$$(4.22)\qquad \frac{\partial \tilde{\dot{\bar{\varepsilon}}}}{\partial X} = \frac{2}{3\tilde{\dot{\bar{\varepsilon}}}} \tilde{\dot{\varepsilon}} : \frac{\partial \tilde{\dot{\varepsilon}}}{\partial X} \quad \text{avec } \frac{\partial \tilde{\dot{\varepsilon}}}{\partial X} = \sum_k \frac{\partial B^k}{\partial X} V^k$$

$$(4.23)\qquad \frac{\partial \tilde{\dot{\bar{\varepsilon}}}}{\partial V} = \frac{2}{3} \tilde{\dot{\varepsilon}} : \sum_k B^k$$

Le calcul de la dérivée du tenseur *B* est renvoyé en annexe. Comme exposé au §3.1.3, cette intégration ne peut être menée classiquement, comme une intégrale de bord. La valeur de $\tilde{\dot{\bar{\varepsilon}}}$ au bord est obtenue par restriction de l'interpolation sur l'élément entier. Néanmoins, la méthode générale reste inchangée et les équations ne sont pas formellement différentes.

4.2 Calcul incrémental des sensibilités

4.2.1 Sensibilités des variables principales – intégration temporelle

Nous venons de voir que les dérivées des fonctions coût sont assujetties au calcul des sensibilités des variables principales du problème direct *X*, *V* et $\bar{\varepsilon}$. De part le schéma d'intégration temporelle explicite et la formulation en vitesse, les variables mécaniques du problème direct sont obtenues en fonction du champ de vitesse *V*. Nous allons voir qu'il en est de même pour le problème de sensibilité : la connaissance de $\mathring{V}$ permet d'avoir accès aux sensibilités de l'ensemble des variables mécaniques. Observons tout d'abord qu'au début du procédé, le maillage du lopin initial étant indépendant de la forme des outils :

$$(4.24)\qquad \mathring{X}^0 = 0$$

Toutefois, il est également possible d'optimiser la forme du lopin initial [Fourment 96b], [Badrinarayanan 95]. L'idée est de déterminer directement la forme optimale de la préforme, et non plus de l'outil qui sert à l'obtenir. Dans ce cas, on paramètre le contour du lopin initial. Les positions des nœuds frontière sont définies par l'équation de la courbe (spline, Bspline ou polynomiale) (cf. §3.2). Les sensibilités initiales sont alors calculées en différentiant l'équation de la courbe. Ainsi, s'il est non nul, $\mathring{X}^0$ est connu en début de procédé.

De la même manière, le champ initial des déformations est généralement indépendant des paramètres de forme :

(4.25) $$\overset{\circ}{\bar{\varepsilon}}^{0} = 0$$

Les conditions initiales étant fixées, les sensibilités $\overset{\circ}{X}$ et $\overset{\circ}{\bar{\varepsilon}}$ sont calculées par une méthode incrémentale similaire à celle utilisée pour le problème direct. Plaçons nous à un instant t quelconque, et supposons connu le champ $\overset{\circ}{V}$ au même instant t, on peut alors exprimer $\overset{\circ}{X}$ à l'incrément suivant, en différentiant le schéma d'intégration temporelle (2.27) :

(4.26) $$\overset{\circ}{X}^{t+\Delta t} = \overset{\circ}{X}^{t} + \overset{\circ}{V}^{t}\Delta t + V\overset{\circ}{\Delta t}$$

Le terme $\overset{\circ}{\Delta t}$ est non nul lorsque le pas de temps est calculé en fonction des évènements de contact. La relation (4.26) s'applique à tous les nœuds internes et de la frontière libre. Le cas des nœuds en contact sera examiné au §4.4. L'actualisation de $\overset{\circ}{\bar{\varepsilon}}$ est réalisée en deux temps. Tout d'abord en différentiant (2.27) :

(4.27) $$\overset{\circ}{\bar{\varepsilon}}^{t+\Delta t} = \overset{\circ}{\bar{\varepsilon}}^{t} + \overset{\circ}{\dot{\bar{\varepsilon}}}^{t}\Delta t + \dot{\bar{\varepsilon}}\overset{\circ}{\Delta t}$$

Puis on remarque que le champ nodal des vitesses de déformation généralisées est calculé par la méthode de lissage d'Orkisz-Liszka (cf. §2.3.1), à partir des champs nodaux X et V. Par abus de notation, nous omettrons les termes de dérivées secondes dans l'équation (2.54). Le gradient de la vitesse est donc obtenu, en tout nœud k du maillage et à tout instant t, par inversion du système :

(4.28) $$A(X^{t})\,grad(V_k^{t}) = b(X^{t}, V^{t})$$

Ce qui donne, après différentiation :

(4.29) $$A\ \overset{\circ}{grad}(V_k^{t}) = \frac{\partial b}{\partial X^{t}}\overset{\circ}{X}^{t} + \frac{\partial b}{\partial V^{t}}\overset{\circ}{V}^{t} - \frac{\partial A}{\partial X^{t}}\overset{\circ}{X}^{t} grad(V_k^{t})$$

Le système linéaire (4.29) est résolu par une méthode directe. $\overset{\circ}{\dot{\bar{\varepsilon}}}$ est finalement obtenu en considérant (4.29) et (4.30) :

(4.30) $$\overset{\circ}{\dot{\bar{\varepsilon}}} = \frac{2}{3\dot{\bar{\varepsilon}}}\dot{\varepsilon} : \overset{\circ}{\dot{\varepsilon}} \quad \text{et} \quad \overset{\circ}{\dot{\varepsilon}} = \frac{1}{2}\left(\overset{\circ}{grad}V + \overset{\circ}{grad}V^{T}\right)$$

Il est ainsi possible de calculer $\overset{\circ}{X}$ et $\overset{\circ}{\bar{\varepsilon}}$, et les sensibilités des variables en découlant, jusqu'à la fin du procédé. Les calculs de $\overset{\circ}{\Delta t}$ et $\overset{\circ}{V}$ sont présentés aux paragraphes §4.2.3 et §4.3.

4.2.2 Calage des outils

En début de procédé, les matrices sont en position ouverte. Le forgeage proprement dit commence au temps t_c, lorsque l'outil entre en contact avec la matière (Figure 4.2). Or, la distance de calage d_c, et par conséquent t_c, dépendent de la géométrie des outils et du maillage. Pour simplifier, on suppose que le lopin initial est indépendant de p. Durant le calage, si le point de l'outil où a lieu le contact appartient à une portion paramétrée de l'outillage, d_c est alors fonction des paramètres de forme décrivant l'outil : $d_c = d_c(p)$. De plus, lors du calage de la passe de finition, d_c dépend aussi du maillage déformé issu de la première opération (Figure 4.3), qui est lui-même dépendant de p.

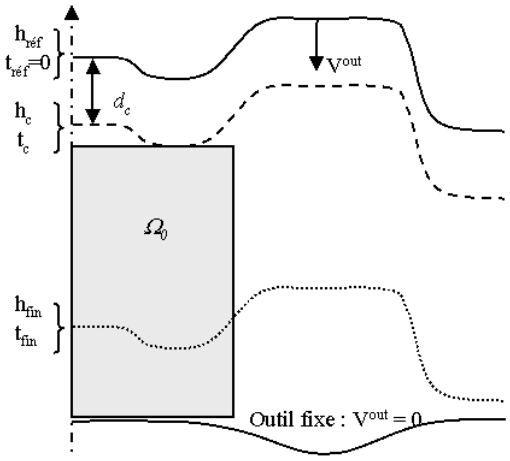


Figure 4.2 : *Calage des outils pour la première opération.*

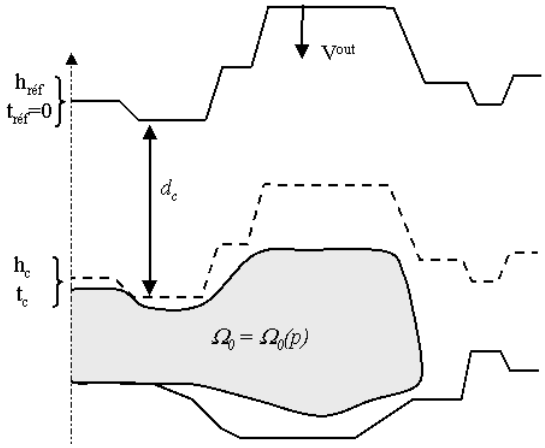


Figure 4.3 : *Calage des outils pour la deuxième opération.*

D'une manière plus générale, on considérera un temps de référence $t_{ref}=0$, où le lopin est positionné entre les matrices, à une hauteur de référence H_{ref} (Figure 4.4). Au temps t_{c1}, le maillage est calé sur l'outil inférieur, à une vitesse « fictive » V_c. Le forgeage commence à l'instant t_{c2}, lorsque l'outil supérieur entre en contact avec le lopin.

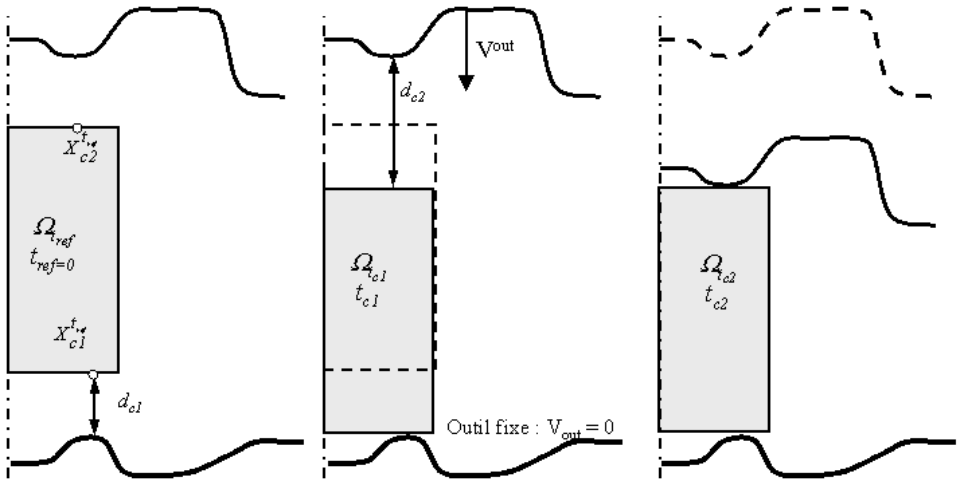


Figure 4.4 : *Définition des phases successives de calage.*

A un instant *t* quelconque du procédé, la position des nœuds est repérée par :

$$X^t(p) = X^{t_{ref}} + \int_{t_{ref}}^{t_{c1}(p)} V(t,p)dt + \int_{t_{c1}(p)}^{t_{c2}(p)} V(t,p)dt + \int_{t_{c2}(p)}^{t} V(t,p)dt \tag{4.31}$$

En remarquant que le champ de vitesse est nul entre les instants t_{c1} et t_{c2}, et constant (égal à V_c) entre les instants t_{ref} et t_{c1}, (4.31) se résume à :

$$X^t(p) = X^{t_{ref}} + (t_{c1} - t_{ref})V_c + \int_{t_{c2}(p)}^{t} V(t,p)dt \tag{4.32}$$

L'instant *t* étant indépendant des paramètres, la sensibilité est donnée par :

$$\overset{\circ}{X}^t(p) = \overset{\circ}{X}^{t_{ref}} + (\overset{\circ}{t}_{c1} - \underbrace{\overset{\circ}{t}_{ref}}_{=0})V_c + \int_{t_{c2}(p)}^{t} \overset{\circ}{V}(t,p)dt - V^{t_{c2}}\,\overset{\circ}{t}_{c2} \tag{4.33}$$

Pour la première opération, $\overset{\circ}{X}_{tref} = 0$ (sauf si on optimise la forme du lopin). A la deuxième opération, ces valeurs sont données par les sensibilités calculées durant la première passe. Les dérivées des instants de calage sont obtenues par :

$$\overset{\circ}{t}_{c1}(p) = \frac{1}{\|V_c\|}\overset{\circ}{d}_{c1} \quad \text{et} \quad \overset{\circ}{t}_{c2}(p) = \frac{1}{\|V_{out}\|}\overset{\circ}{d}_{c2} \tag{4.34}$$

V_c étant orientée négativement suivant l'axe des ordonnées, (4.33) s'écrit finalement :

$$\overset{\circ}{X}^t(p) = \overset{\circ}{d}_{c1}\begin{bmatrix} 0 \\ -1 \end{bmatrix} + \int_{t_{c2}(p)}^{t} \overset{\circ}{V}(t,p)dt - \frac{V^{t_{c2}}}{\|V_{out}\|}\overset{\circ}{d}_{c2} \tag{4.35}$$

Si on repère par X_{out}^{c1} et X_{out}^{c2} les points de l'outillage où a lieu le contact durant les calages (inférieurs et supérieurs respectivement), et $X_{c1}^{t_{ref}}$ et $X_{c2}^{t_{ref}}$ les points correspondant sur la configuration de référence (Figure 4.4), ces distances sont données par :

$$d_{c1}(p) = \left(Z_{c1}^{t_{ref}}(p) - Z_{out}^{c1}(p)\right) \quad \text{et} \quad d_{c2}(p) = \left(Z_{out}^{c2}(p) - Z_{c2}^{t_{ref}}(p) - d_{c1}(p)\right) \tag{4.36}$$

avec, pour la première opération.

$$\overset{\circ}{Z}_{c1}^{tref} = 0 \quad \text{et} \quad \overset{\circ}{Z}_{c2}^{tref} = 0 \tag{4.37}$$

Les termes $\overset{\circ}{Z}_{out}^{c1}$ et $\overset{\circ}{Z}_{out}^{c2}$, intervenant lors de la dérivation de (4.36) sont obtenus en différentiant les équations des courbes paramétrées définissant les outils inférieur et supérieur.

4.2.3 Dérivation du pas de temps

Le pas de temps Δt, intervenant dans les équations d'actualisation des sensibilités (4.26) et (4.27), peut être calculé suivant différentes options : constant pendant toute la simulation et égal à une valeur fixée par l'utilisateur, ou bien choisi de telle manière que la déformation moyenne par incrément ne dépasse pas une valeur prescrite. Une troisième stratégie consiste à ajuster ce pas à la valeur du plus court temps d'arrivée d'un nœud en contact. Cette dernière s'avère en général plus précise, car elle permet d'éliminer les projections dues aux pénétrations des nœuds de la surface libre. Si à l'instant t, on note X_i^t les coordonnées du nœud frontière i qui entre le premier en contact après calcul du champ de vitesse, Δt est alors déterminé grâce au point d'intersection X_B avec l'outil (Figure 4.5).

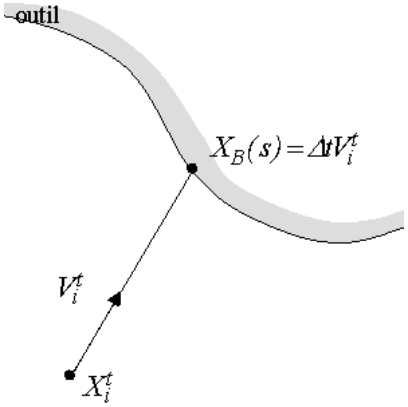


Figure 4.5 : *Détermination du pas de temps*

$$X_i^{t+\Delta t} = X_B = X_i^t + \Delta t V_i^t \tag{4.38}$$

En multipliant (4.38) par le vecteur vitesse, en différentiant et en regroupant les termes :

$$\overset{\circ}{\Delta t}\left\|V_i^t\right\|^2 = \left(\overset{\circ}{X}_B - \overset{\circ}{X}_i^t - 2\Delta t\overset{\circ}{V}_i^t\right).V_i^t + \left(X_B - X_i^t\right).\overset{\circ}{V}_i^t \tag{4.39}$$

L'équation (4.39) se simplifie en remarquant que $\left(X_B - X_i^t\right) = \Delta t V_i^t$ pour donner finalement :

$$\overset{\circ}{\Delta t} = \frac{1}{\left\|V_i^t\right\|^2}\left(\overset{\circ}{X}_B - \overset{\circ}{X}_i^t - \Delta t\overset{\circ}{V}_i^t\right).V_i^t \tag{4. 40}$$

Lorsque le point X_B appartient à une partie paramétrée de l'outil, la valeur de $\overset{\circ}{X}_B$ est obtenue en différentiant l'équation de paramétrisation de la courbe. Ce calcul est présenté au §4.4 dans le cas des courbes Bsplines. A l'inverse, lorsque X_B appartient à une partie fixe de l'outil, ce terme est nul.

Dans la suite de ce travail, nous avons négligé $\overset{\circ}{\Delta t}$. En effet, dans le logiciel Forge2®, le calcul du pas de temps est sensiblement plus complexe que ce qui vient d'être présenté et sa

différentiation plus délicate. Toutefois, nos expériences numériques ont montré qu'il était justifié d'omettre $\mathring{\Delta t}$. Les calculs précédents montrent comment le prendre en compte, en première approximation, si nécessaire.

4.3 Dérivation de l'équation d'équilibre

Le calcul des sensibilités des variables nodales du procédé repose sur le calcul de $\mathring{V}$. Cette variable est naturellement obtenue en différentiant l'équation d'équilibre (2.30) définissant la variable directe V dont elle dérive. Nous avons vu (cf. §2.2.3) que l'imposition des conditions aux limites en vitesse est réalisée par changement de repère, en utilisant des matrices locales de rotation Q. Pour tous les nœuds en contact, ce changement de base revient à imposer par élimination, la vitesse de l'outil dans la direction normale, et à résoudre l'équation résiduelle d'équilibre dans la direction tangente. La différentiation de l'équation (2.30) donne :

$$(\mathring{QR}) = \mathring{Q}R + Q\mathring{R} = 0 \tag{4.41}$$

On remarque que la matrice Q dépend effectivement des paramètres. Les matrices locales Q_k la constituant sont composées des normales et tangentes aux segments de l'outillage (2.32). Leurs sensibilités sont donc non nulles pour les segments appartenant à une portion paramétrée de l'outil. Cette matrice est par contre indépendante des variables nodales X, V et $\bar{\varepsilon}$. Ce qui permet d'écrire :

$$\frac{\partial(QR)}{\partial X} = Q\frac{\partial R}{\partial X} \quad \text{et} \quad \frac{\partial(QR)}{\partial \bar{\varepsilon}} = Q\frac{\partial R}{\partial \bar{\varepsilon}} \tag{4.42}$$

En tenant compte de :

$$\mathring{R} = \frac{\partial R}{\partial V}\mathring{V} + \frac{\partial R}{\partial X}\mathring{X} + \frac{\partial R}{\partial \bar{\varepsilon}}\mathring{\bar{\varepsilon}} \tag{4.43}$$

On obtient :

$$Q\frac{\partial R}{\partial V}\mathring{V} = -\left(Q\frac{\partial R}{\partial X}\mathring{X} + Q\frac{\partial R}{\partial \bar{\varepsilon}}\mathring{\bar{\varepsilon}} + \mathring{Q}R\right) \tag{4.44}$$

Que l'on écrit sous la forme :

$$\underbrace{\left(Q\frac{\partial R}{\partial V}Q^T\right)}_{\frac{\partial \bar{R}}{\partial V}}Q\mathring{V} = -\left(Q\frac{\partial R}{\partial X}\mathring{X} + Q\frac{\partial R}{\partial \bar{\varepsilon}}\mathring{\bar{\varepsilon}} + \mathring{Q}R\right) \tag{4.45}$$

L'équation de la composante normale de la vitesse (2.31) fait apparaître la normale n,

intérieure à l'outillage. Pour tous les nœuds k en contact avec une portion paramétrée de l'outillage, cette direction dépend des paramètres. La composante normale de $\mathring{V}_k$ est alors donnée par :

$$\mathring{V}_k .n_k = \left(V^{out} - V_k\right).\mathring{n}_k \tag{4.46}$$

Pour ces degrés de liberté, les lignes correspondantes de (4.45) sont éliminées et remplacées par (4.46).

Il est ainsi possible de calculer $\mathring{V}$, et par conséquent $\mathring{X}$ et $\mathring{\bar{\varepsilon}}$ jusqu'à la fin du procédé. Lors de la résolution de (4.45), le nombre de systèmes linéaires est égal au nombre de paramètres d'optimisation, tous ces systèmes ayant la même matrice hessienne $\partial\breve{R}/\partial V$. Cette matrice est aussi la matrice du système linéaire tangent calculée dans l'algorithme de Newton-Raphson (cf. §2.2.4). A la fin de cet algorithme, on connaît $\partial\breve{R}/\partial V$ pour le champ de vitesse de l'avant dernière itération En revanche, la matrice $\partial\breve{R}/\partial V$ qui apparaît dans (4.45) correspond au champ solution V. Des tests numériques montrent que les résultats obtenus avec la matrice approchée issue de l'algorithme de Newton-Raphson sont quasiment identiques [Chenot 95]. La matrice approchée étant déjà triangularisée, utiliser cette approximation permet de réduire encore davantage les temps de calcul de l'analyse de sensibilité.

L'équation (4.45) nécessite le calcul des dérivées de l'équation résiduelle d'équilibre par rapport aux variables principales X, V et $\bar{\varepsilon}$. Le terme $\partial R/\partial V$ est déjà calculé pour résolution du champ de vitesse. Le calcul de la matrice $\partial R/\partial X$ constitue la partie la plus délicate de ces différentiations de part le grand nombre de variables intervenant dans (2.26) et dépendant implicitement de la variable X. Ce calcul ayant déjà été présenté par Fourment *et al*. [Fourment 96a], nous ne reprendrons de manière simplifiée, que le calcul de la dérivée du terme de comportement. Par ailleurs, ces calculs ne présentent aucune difficulté particulière et sont communs à la plupart des problèmes d'optimisation, comme par exemple en calcul des structures [Hsu 94] [Haftka 92].

Tout comme pour le calcul des dérivées des fonctions coût, l'idée de base consiste à effectuer les dérivations sur la configuration de référence, indépendante des paramètres. Le terme de comportement de (2.26) s'exprime alors, sur l'espace des éléments de référence, sous la forme :

$$R_c^k = \int_{\tilde{\Omega}_{ref}} 2\tilde{K}\left(\sqrt{3}\tilde{\dot{\bar{\varepsilon}}}\right)^{m-1} \tilde{\dot{\varepsilon}} : B^k \, 2\pi\tilde{r}\bar{J} dw_{ref} \tag{4.47}$$

Par dérivation directe de (4.47), on obtient :

(4.48)

$$\frac{\partial R_c^k}{\partial X} = \int_{\tilde{\Omega}_{ref}} 2\tilde{K}\left(\sqrt{3}\tilde{\dot{\bar{\varepsilon}}}\right)^{m-1} \left(\left((m-1)\frac{1}{\tilde{\dot{\bar{\varepsilon}}}} \frac{\partial \tilde{\dot{\bar{\varepsilon}}}}{\partial X} \tilde{\dot{\varepsilon}} : B^k + \frac{\partial \tilde{\dot{\varepsilon}}}{\partial X} : B^k + \tilde{\dot{\varepsilon}} : \frac{\partial B^k}{\partial X} \right) \tilde{r}\bar{J} + \tilde{\dot{\varepsilon}} : B^k \frac{\partial \tilde{r}\bar{J}}{\partial X} \right) 2\pi dw_{ref}$$

Les termes apparaissant dans (4.48) ont déjà été explicités lors du calcul des dérivées des fonctions coût (cf. §4.1), certains d'entre eux sont détaillés en annexe.
De même :

(4.49) $$\frac{\partial R_c^k}{\partial \bar{\varepsilon}} = \int_{\tilde{\Omega}_{ref}} 2\tilde{K}\left(\sqrt{3}\tilde{\dot{\varepsilon}}\right)^{m-1} \frac{\partial \tilde{K}}{\partial \bar{\varepsilon}} \tilde{\dot{\varepsilon}} : B^k \, 2\pi dw_{ref}$$

Les autres termes sont obtenus de manière similaire.

4.4 Traitement du contact

La formule d'actualisation de $\mathring{X}$ (4.26) ne tient pas compte des éventuelles projections des nœuds frontières sur la surface de l'outillage, et ne s'applique donc que pour les nœuds internes ou de frontière libre. Considérons un nœud du maillage *k* arrivant en contact avec la surface de l'outillage (Figure 4.6), ou déjà en contact mais pénétrant l'outillage (Figure 4.7). On suppose que la surface de l'outil est décrite par une courbe Bspline. Si cette portion correspond à une partie non paramétrée de l'outillage, les formules qui suivent se simplifient en remarquant que $\mathring{t} = \mathring{n} = 0$. Ces mêmes formules s'appliquent dans le cas d'un nœud quittant le contact et devant être reprojeté sur l'outil.

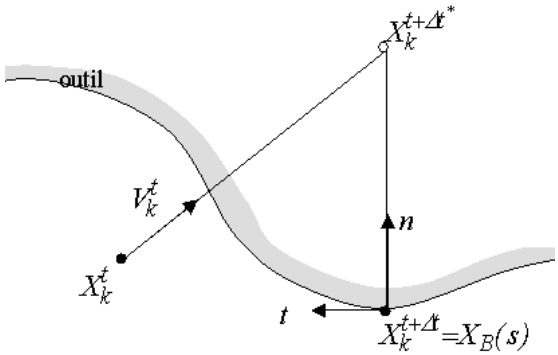


Figure 4.6 : *Projection d'un nœud arrivant en contact*

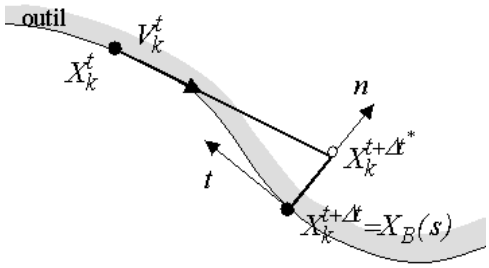


Figure 4.7 : *Projection d'un nœud déjà en contact.*

Les coordonnées du nœud en fin d'incrément sont obtenues en calculant, tout d'abord, une position fictive $X_k^{t+\Delta t^*}$, en l'absence de contact :

(4.50) $$X_k^{t+\Delta t^*} = X_k^t + V_k^t \Delta t$$

On projète ce point sur la surface de l'outil. La projection $X_k^{t+\Delta t}$ coïncide avec le point X_B de la Bspline (Figure 4.6), de tangente *t* et de normale *n* :

(4.51) $$X_k^{t+\Delta t} = X_B \quad \text{et} \quad \left(X_k^{t+\Delta t} - X_k^{t+\Delta t^*}\right).t = 0$$

En différentiant (4.51) :

$$\left(\overset{\circ}{X}_k^{t+\Delta t} - \overset{\circ}{X}_k^{t+\Delta t^*} \right).t + \left(X_k^{t+\Delta t^*} - X_k^{t+\Delta t} \right).\overset{\circ}{t} = 0 \tag{4.52}$$

La composante tangentielle de $\overset{\circ}{X}_k^{t+\Delta t}$ s'exprime donc en fonction de $\overset{\circ}{X}_k^{t+\Delta t^*}$, qui est directement donné par (4.26) et $\overset{\circ}{t}$:

$$\overset{\circ}{X}_k^{t+\Delta t}.t = \overset{\circ}{X}_k^{t+\Delta t^*}.t + \left(X_k^{t+\Delta t^*} - X_k^{t+\Delta t} \right).\overset{\circ}{t} \tag{4.53}$$

Le point X_B appartient à la courbe Bspline. Ses coordonnées sont données par l'équation (3.22) et s'expriment en fonction de l'abscisse s. s est définie par la projection sur la Bspline, et dépend des paramètres de forme p :

$$X_B = X_B\left(s(p), p\right) \tag{4.54}$$

$$\overset{\circ}{X}_B(s) = \begin{bmatrix} \frac{dr_B}{dp}(s) \\ \frac{dz_B}{dp}(s) \end{bmatrix} = \sum_i M_i(s) \begin{bmatrix} \frac{dR_i}{dp} \\ \frac{dZ_i}{dp} \end{bmatrix} + \left(\sum_i \frac{dM_i}{ds}(s) \begin{bmatrix} R_i \\ Z_i \end{bmatrix} \right) \frac{ds}{dp} \tag{4.55}$$

De part la définition des courbes Bsplines, le deuxième terme de (4.55) est le vecteur tangent à la courbe au point X_B. L'équation (4.55) s'écrit alors sous la forme :

$$\overset{\circ}{X}_B(s) = \begin{bmatrix} \frac{dr_B}{dp}(s) \\ \frac{dz_B}{dp}(s) \end{bmatrix} = \sum_i M_i(s) \begin{bmatrix} \frac{dR_i}{dp} \\ \frac{dZ_i}{dp} \end{bmatrix} + \left\| \frac{\partial X_B}{\partial s}(s) \right\| t \frac{ds}{dp} \tag{4.56}$$

En multipliant (4.56) par la normale n, on obtient la composante normale de $\overset{\circ}{X}_k^{t+\Delta t}$:

$$\overset{\circ}{X}_k^{t+\Delta t}.n = \overset{\circ}{X}_B(s).n = \sum_i M_i(s) \begin{bmatrix} \frac{dR_i}{dp}(s) \\ \frac{dZ_i}{dp}(s) \end{bmatrix}.n \tag{4.57}$$

Les termes dR_i/dp et dZ_i/dp sont obtenus à partir de la définition des paramètres d'optimisation (cf. §3.2). La sensibilité de la tangente t, intervenant dans (4.53), ainsi que dans le calcul des dérivées des matrices de rotation Q, est obtenue à partir de l'équation de la direction tangente (3.23), que l'on écrira pour alléger les notations :

$$t = \sum_i t_i(C_1(p), .., C_{Nbsom}(p), s(p)) \tag{4.58}$$

La sensibilité de la tangente à la courbe Bspline au point X_B s'exprime par :

$$\overset{\circ}{t} = \sum_i \sum_k \frac{\partial t_i}{\partial C_k} \frac{dC_k}{dp} + \sum_i \frac{\partial t_i}{\partial s} \frac{ds}{dp} \cong \sum_i \sum_k \frac{\partial t_i}{\partial C_k} \frac{dC_k}{dp} \tag{4.59}$$

En pratique, on négligera le deuxième terme de (4.59), en considérant que le terme $\partial t/\partial s \; ds/dp$ du second ordre par rapport à $\partial t/\partial C \; dC/dp$.

Chapitre 5

Applications

Dans les chapitres précédents, nous avons présenté la méthode incrémentale utilisée pour calculer les sensibilités des variables d'état du problème direct. Ces dernières permettent de calculer les gradients des différentes fonctions coût envisagées lors d'un problème d'optimisation. La précision de ces calculs est importante. Le succès des algorithmes de minimisation de type gradient que nous utilisons, est en effet conditionné par la précision des dérivées qui leurs sont fournies. Une première partie du travail de validation des dérivées analytiques a été menée par Bălan [Bălan 96]. Cette étude porte plus particulièrement sur l'influence du contact, du pas de temps et de la finesse du maillage, sur la précision des dérivées. Elle a été réalisée en comparant les sensibilités analytiques aux valeurs obtenues par un schéma aux différences finies.

Les conclusions de ce travail nous ont guidé dans nos choix et se sont avérées précieuses afin d'obtenir une précision suffisante dans les calculs de sensibilité, pour résoudre des cas d'optimisation complexes. Nous avons ainsi utilisé des maillages fins, des pas de temps adaptés au contact. Une technique similaire a été utilisée pour valider certains développements, comme par exemple la dérivation des équations discrètes par rapport à la déformation $\overline{\varepsilon}$, ou la dérivation de la méthode de lissage d'Orkisz-Liszka (cf. §2.3.1).

Dans le paragraphe 5.1, on s'intéresse au coût, en terme de temps de calcul, de l'analyse de sensibilité, ainsi qu'à la validation des calculs analytiques des dérivées. Un cas particulier d'optimisation de matrice de finition est présenté au §5.2. Les problèmes suivants porteront sur l'optimisation d'outils de préforme, en commençant (§5.3), par deux problèmes d'optimisation sous contrainte. Nous regarderons ensuite l'influence du nombre de paramètres, et les informations que peut apporter l'analyse de sensibilité. Les deux derniers paragraphes portent sur des problèmes industriels présentant des défauts de repli.

5.1 Analyse du coût de calcul et de la précision des dérivées

5.1.1 Surcoût de calcul dû à l'analyse de sensibilité

Nous allons estimer le surcoût, en temps de calcul, que représente l'analyse de sensibilité. Il est en effet important que le coût de calcul de $\mathring{\tilde{\Phi}}$ reste inférieur à celui de $\tilde{\Phi}$. Dans le cas contraire, l'utilisation d'un algorithme de type gradient peut devenir discutable. Cette conclusion est évidemment valable du strict point de vue du coût et ne tient pas compte des informations auxquelles l'analyse de sensibilité permet d'accéder.

L'analyse de sensibilité est réalisée grâce à une méthode de différentiation directe. La partie susceptible de représenter le plus grand nombre de calculs concerne le système (4.44) :

$$\left(Q\frac{\partial R}{\partial V}Q^T\right)Q\mathring{V} = \frac{\partial \breve{R}}{\partial V}Q\mathring{V} = -\left(Q\frac{\partial R}{\partial X}\mathring{X} + Q\frac{\partial R}{\partial \overline{\varepsilon}}\mathring{\overline{\varepsilon}} + \mathring{Q}R\right) \tag{5.1}$$

On se propose donc de comparer les temps de calcul de $\mathring{V}$ et V. On considère un lopin cylindrique, écrasé par un outil décrit par une Bspline comportant 7 points de contrôle. Le maillage comporte environ 1900 nœuds . La rhéologie du matériau est donnée par K=6.5 10^3 Pa, $m=p_f$=0.25, n=0. Le contact est glissant avec α_f=0.25.

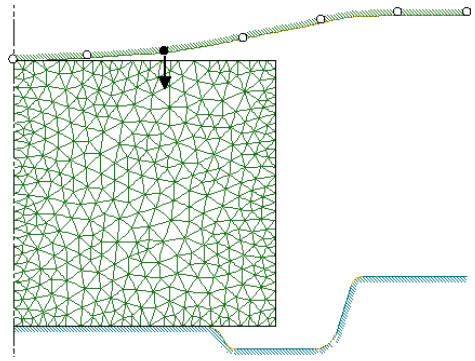


Figure 5.1 : *Maillage et outils utilisés pour les tests de temps – positions des points de contrôle et paramètre utilisé pour la comparaison sur* $\mathring{\Phi}_{ene}$.

Le calcul de $\mathring{V}$ consiste à résoudre autant de systèmes qu'il y a de paramètres. En contrepartie, tous ces systèmes ont la même matrice $\partial \breve{R}/\partial V$. Nous savons qu'il était envisageable de réaliser une économie de temps en ne la recalculant pas pour le champ de vitesse solution, mais en utilisant l'approximation de $\acute{\partial}\breve{R}/\partial V$ issue de la résolution de Newton-Raphson, qui présente l'avantage d'être déjà factorisée. On se propose donc de vérifier le gain et la validité de cette approximation.

La simulation est réalisée sur un *SUN interprise-4000*. Les mesures sont effectuées sur 50 incréments, soit un écrasement du lopin de 20%. A chaque incrément, on considère le temps total requis pour les différentes étapes :

- calcul des termes $\acute{\partial}\breve{R}/\partial X$, $\acute{\partial}\breve{R}/\acute{\partial}\bar{\varepsilon}$, Q , R et $\mathring{Q}$ - assemblage du second membre
- calcul et factorisation de $\acute{\partial}\breve{R}/\partial V$ (lorsque cette matrice est recalculée)
- résolution du système (produit vecteur-matrice)

De même, pour la résolution du problème direct, on considère le temps total nécessaire au calcul des matrices, des seconds membres, des factorisations et des résolutions, pour chaque itération de Newton-Raphson. Lorsque l'on recalcule la matrice hessienne, les temps cumulés des calculs de V et $\mathring{V}$ sont représentés Figure 5.2, pour des nombres croissants de paramètres (1,3,5,7). Les temps correspondants à l'utilisation de l'approximation de $\acute{\partial}\breve{R}/\partial V$ sont reportés sur la Figure 5.3.

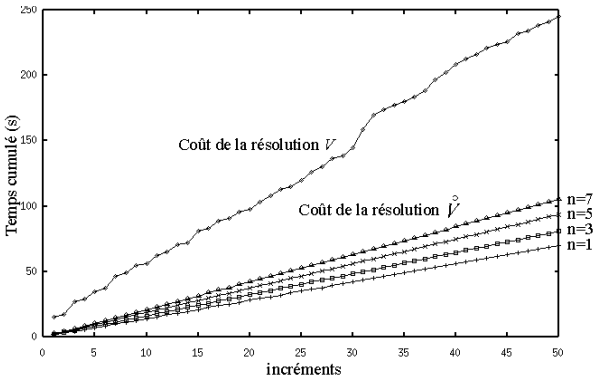


Figure 5.2 : *Temps cumulés des calculs de V et* $\mathring{V}$ *avec un recalcul de* $\acute{\partial}\breve{R}/\partial V$.

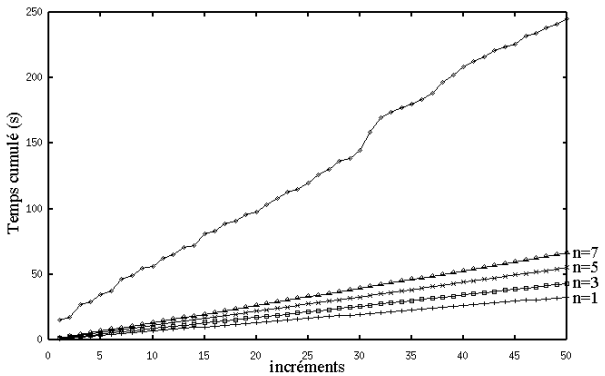


Figure 5.3 : *Temps cumulés des calculs de V et* $\mathring{V}$ *sans recalcul de* $\acute{\partial}\breve{R}/\partial V$.

Comme prévu, le temps de résolution du problème de sensibilité augmente avec le nombre de paramètres. Mais alors que le temps de calcul de $\mathring{V}$ représente environ 40% de celui de V lorsque l'on recalcule $\partial \tilde{R}/\partial V$ (pour 7 paramètres), il est inférieur à 30% dans le cas contraire. L'économie ainsi réalisée correspond au coût de factorisation de la matrice hessienne, qui est approximativement en $2N^2$ opérations (où N désigne le nombre de degrés de liberté). Le gain de temps est donc d'autant plus important que les maillages utilisés sont fins.

Si on suit les tendances données par la Figure 5.3, le coût de l'analyse de sensibilité devient équivalent à la résolution du problème direct pour un nombre de paramètres supérieur à 35. Cette valeur limite permet d'avoir une certaine marge de sécurité, d'autant que la description des contours à optimiser est généralement réalisée pour un petit nombre de paramètres (inférieur à 10 dans la pratique). L'analyse de sensibilité par une méthode de différentiation directe apparaît donc justifiée. Le surcoût de 30% (pour 7 paramètres) dans le cas présent, doit en effet être comparé au 700% qu'aurait entraîner un calcul des dérivées par différences finies.

Afin d'estimer la perte de précision due à cette approximation, on regarde l'évolution de la sensibilité de $\tilde{\Phi}_e$ obtenue lorsque l'on utilise la matrice hessienne approchée. La Figure 5.4 représente, dans les deux cas, les dérivées de $\tilde{\Phi}_e$ par rapport au paramètre défini sur la Figure 5.1.

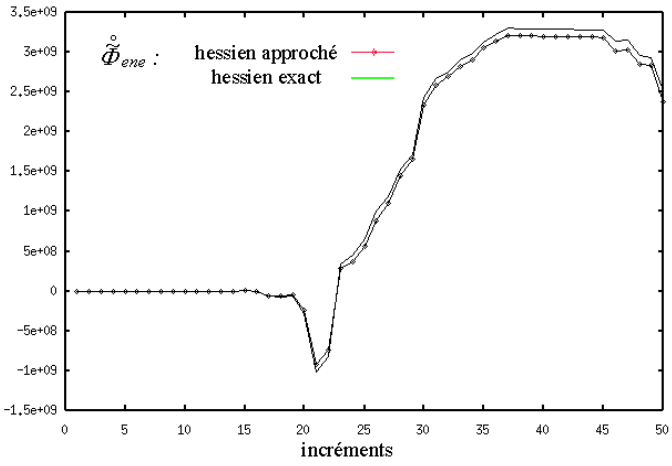


Figure 5.4 : *Evolution de $\mathring{\tilde{\Phi}}_{ene}$ dans le temps. Courbes calculées pour une sensibilité $\mathring{V}$ obtenue soit par recalcul de la matrice hessienne, soit par l'utilisation de l'approximation issue de Newton-Raphson.*

Les courbes présentent un écart inférieur à 3% après 50 incréments de calcul. L'approximation ainsi proposée semble tout à fait licite au regard de la perte de précision.

5.1.2 Validations des dérivées

On considère un matériau écrouissable (n=0.2). Le maillage et l'outillage, ainsi que les autres paramètres rhéologiques, sont identiques à ceux utilisés dans le cas précédent. Pour valider le calcul des dérivées, on utilise un schéma aux différences finies centré. Chaque composante i de la dérivée d'une variable nodale quelconque M est ainsi approchée par :

(5.2)
$$\left(\mathring{M}_{DFC}\right)_i = \frac{M(p+\Delta p\, e_i) - M(p-\Delta p\, e_i)}{2\Delta p}$$

Le vecteur e_i désigne le i[ième] vecteur de base :

(5.3) $$e_i^T = [\underbrace{0\ 0 \cdots 0\ 1}_{i}\ 0 \cdots 0]$$

La précision des dérivées numériques[5] dépend en grande partie de la perturbation Δp. Afin d'éviter les instabilités dues aux erreurs d'arrondi pour les trop petites valeurs de Δp, nous utiliserons une perturbation de 10^{-4} mm, qui se situe dans le domaine des valeurs acceptables pour le cas présent, où le lopin a une hauteur de 100 mm. Les erreurs sont calculées par :

(5.4) $$erreur\,(\overset{\circ}{M}) = \frac{\left\| \overset{\circ}{M}_A - \overset{\circ}{M}_{DFC} \right\|}{\left\| \overset{\circ}{M}_{DFC} \right\|}.\ 100\ [\%]$$

La norme utilisée est la norme euclidienne. On ne calcule qu'une seule composante. Le paramètre perturbé est représenté sur la Figure 5.1.

Les erreurs sur les dérivées des variables intervenant dans la résolution du problème de sensibilité sont données dans le tableau suivant.

Variables	$\overset{\circ}{X}$	$\overset{\circ}{V}$	$\overset{\circ}{X}_{out}$	$\overset{\circ}{t}$	$\overset{\circ}{n}$	$\overset{\circ}{\dot{\bar{\varepsilon}}}$	$\overset{\circ}{\bar{\varepsilon}}$
Erreur [%]	$9{,}0\ 10^{-4}$	$3{,}1\ 10^{-4}$	$3{,}2\ 10^{-6}$	$5{,}2\ 10^{-6}$	$5.1\ 10^{-6}$	$3{,}5\ 10^{-4}$	$9{,}3\ 10^{-4}$

Ces valeurs ont été obtenues au 5[ième] incrément de calcul. La Figure 5.5 montre l'évolution de l'erreur sur $\overset{\circ}{\bar{\bar{\varepsilon}}}$ et $\overset{\circ}{\dot{\bar{\varepsilon}}}$ au cours du temps. L'augmentation est due en particulier aux formules d'actualisation de $\overset{\circ}{X}$ et $\overset{\circ}{\dot{\bar{\varepsilon}}}$ (4.26-4.27), qui conduisent à un cumul de l'erreur dans le temps. L'erreur reste toutefois inférieure à 2% après 50 incréments, soit 20% de déformation.

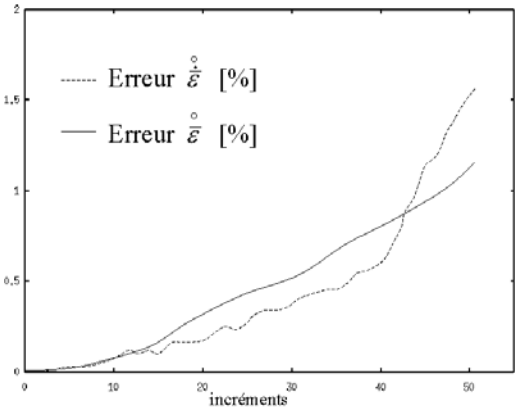


Figure 5.5 : *Evolution de l'erreur sur* $\overset{\circ}{\bar{\bar{\varepsilon}}}$ *et* $\overset{\circ}{\dot{\bar{\varepsilon}}}$ *au cours du temps.*

Les frontières d'outil sont différentes pour $p+\Delta p$, p, $p-\Delta p$. En conséquence, si les pas de temps sont adaptés pour empêcher les pénétrations des nœuds frontières dans les outils, les instants de calcul sont différents suivant les configurations perturbées. Les dérivées numériques perdent alors tout sens. Ces calculs ont donc été effectués avec des pas de temps

[5] On appellera « numériques » les valeurs des dérivées obtenues par différences finies

constants, ce qui implique de projeter les nœuds frontières ayant pénétré dans l'outillage. Il faut de plus considérer les projections des nœuds en contact qui glissent sur les segments de l'outil. Ces phases de reprojection induisent des discontinuités qui sont les principales sources d'imprécision pour l'analyse de sensibilité. Malheureusement, le traitement du contact altère aussi les valeurs des dérivées par différences finies. Après 50 incréments de calcul, l'écart entre les dérivées numériques obtenues par différents schémas (centré, décentré à gauche et à droite) est de l'ordre de 2%, ce qui est comparable à l'écart que l'on vient de déterminer avec les dérivées analytiques.

Il est donc important de posséder un moyen autre que les dérivées numériques pour valider tout nouveau calcul de sensibilité, comme par exemple $\mathring{\dot{\varepsilon}}$ (cf. §4.2.1). Afin de déterminer si l'erreur de 1,5% avec les dérivées numériques est imputable à une erreur de programmation, où aux imprécisions sur $\mathring{X}$ et $\mathring{V}$; nous avons utilisé le logiciel de différentiation automatique ODYSSEE® développé à l'INRIA. A partir du code source, calculant $\dot{\varepsilon}$ en fonction de X et V (cf. §2.3.1), ce logiciel nous a fourni un nouveau code, calculant $\acute{c}\dot{\varepsilon}/\partial V$ et $\acute{c}\dot{\varepsilon}/\partial X$. Il suffit alors d'effectuer l'assemblage :

$$\mathring{\dot{\varepsilon}}_{OD} = \left(\frac{\partial \dot{\varepsilon}}{\partial X}\right)_{OD} \mathring{X} + \left(\frac{\partial \dot{\varepsilon}}{\partial V}\right)_{OD} \mathring{V} \tag{5.5}$$

L'écart entre les dérivées analytiques « manuelles » et celles $\mathring{\dot{\varepsilon}}_{OD}$ ainsi calculées, est inférieur à 0,01% et reste constant au cours du temps. Ceci permet donc de valider la dérivation de la méthode de lissage avec certitude.

Néanmoins, si les outils de différentiation constituent un puissant moyen de validation des calculs analytiques, leur principal objectif réside bien dans le développement de nouveaux calculs de sensibilité : lois de comportement, variables thermiques, etc. . Faute de temps, nous n'avons pas poursuivi dans cette voie. Mais ces outils semblent prometteurs quant à leurs apports indiscutables en terme de précision.

5.2 Optimisation des outils de finition – Forgeage d'une roue

Le procédé étudié nous a été proposé par Mr. M. Payelleville[6]. Il s'agit du forgeage d'une roue en deux opérations. La première passe consiste en un simple écrasement entre tas plats. On s'intéresse uniquement à l'opération de finition. Après forgeage, le plateau supérieur (ou toile) est éliminé. Les dimensions des autres parties des matrices étant imposées par la géométrie de la pièce, la position de ce plateau constitue le seul paramètre géométrique du procédé (Figure 5.6). La hauteur de cette toile a une influence importante sur l'écoulement. Lorsqu'elle est en position haute, elle provoque un écoulement durant lequel la matière se cintre pour donner naissance à un repli (Figure 5.7).

[6] Professeur au Lycée Marie Curie, Nogent/Oise

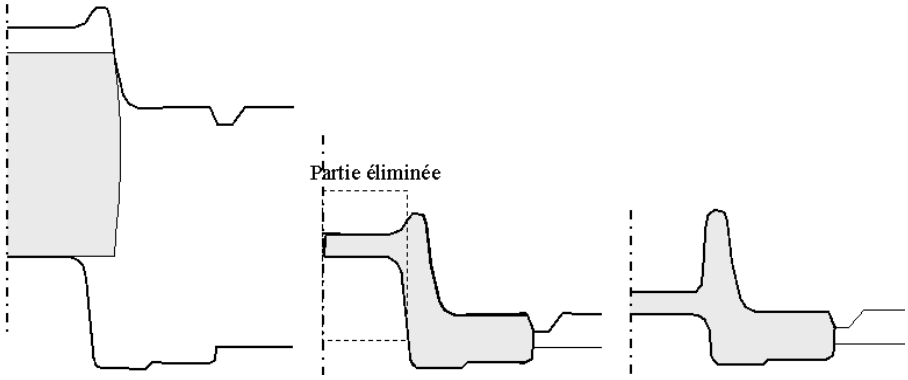


Figure 5.6 : *Dernière opération de forgeage d'une roue. Différentes positions de toile.*

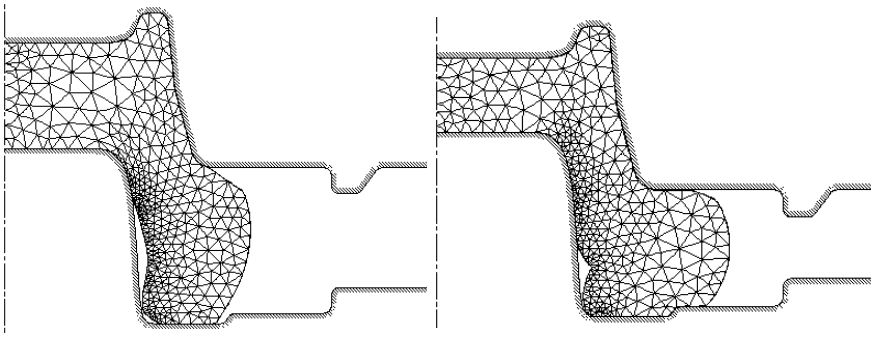


Figure 5.7 : *Formation d'un repli : toile en position haute.*

On se propose de déterminer la hauteur optimale de cette toile, afin de faire disparaître ce défaut. La paramétrisation est représentée sur la Figure 5.8. Elle est réalisée par deux courbes Bsplines comportant chacune 7 points de contrôle, dont 3 fixes. Les points mobiles restants, sont tous liés entre eux. Il n'y a donc qu'un seul paramètre qui règle la hauteur de la toile. Dans le calcul de la fonction coût repli Φ_{rep} (3.18), on ne tient pas compte de la frontière libre située en bavure.

La rhéologie de matériau est donnée par K=1.65 10^3 Pa, m=p_f =0.14, n=0. Le contact est glissant avec α_f=0.2. La hauteur initiale du lopin (après la première opération de forgeage) est de 550mm, l'épaisseur du plateau en fin de procédé est de 40mm.

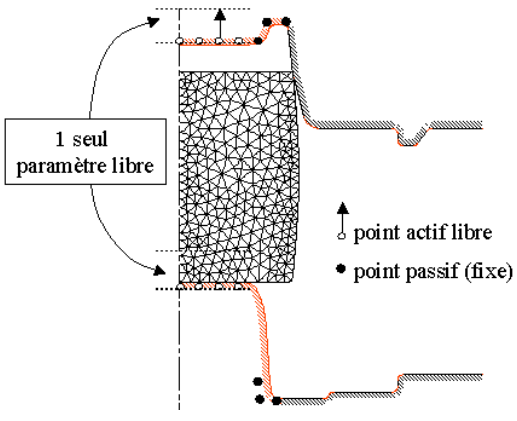


Figure 5.8 : *Paramétrisation de l'outil. La hauteur du plateau est déterminée par un seul paramètre (p_0 = 0).*

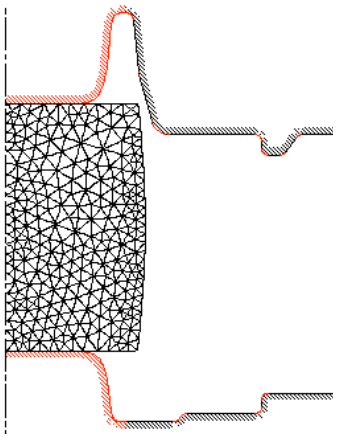


Figure 5.9 : *Hauteur à partir de laquelle le repli n'est plus déclaré ($p_7 \cong$ -8).*

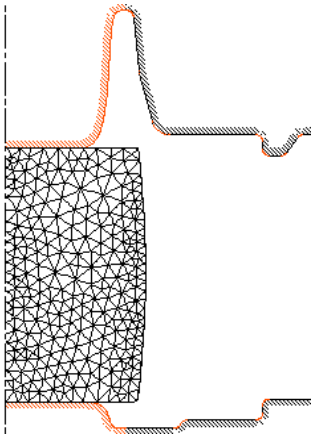


Figure 5.10 : *Hauteur optimale de la toile ($p_{11} \cong$ -15).*

L'algorithme a convergé en 4 itérations (11 simulations). Le repli n'apparaît plus à partir de la 7ième simulation (Figure 5.9). Le procédé optimal (correspondant à la plus faible valeur de $\tilde{\Phi}_{rep}$) est représenté sur la Figure 5.11.

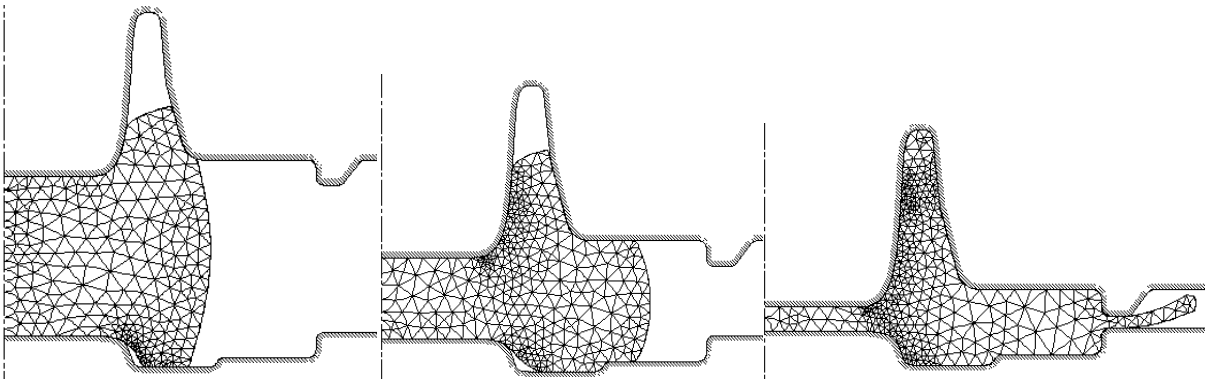


Figure 5.11 : *Procédé optimal - différents instants de forgeage. Absence de repli.*

La fonction ne dépendant que d'une variable, elle peut facilement être approchée. Nous avons ainsi tracé l'évolution de $\tilde{\Phi}_{rep}$, pour une faible plage de variation du paramètre : p ($p \in [-9, -5]$). La « courbe » de la Figure 5.12 a été obtenue au prix de 40 simulations. Cette fonction est globalement décroissante, mais présente des remontées locales, qui peuvent être le reflet de minima locaux, ou encore la conséquence des erreurs de calcul et de discrétisation. Cet exemple traduit la difficulté du problème. La convergence est ici assurée par les valeurs globalement positives du gradient (le paramètre est orienté négativement) (Figure 5.13). On peut même remarquer que certaines valeurs de la dérivée semblent correspondre à des minima locaux, numériques ou non.

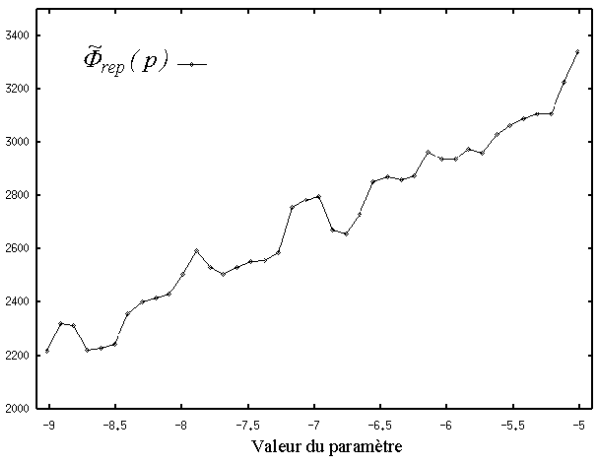


Figure 5.12 : *Représentation graphique de la fonction coût.*

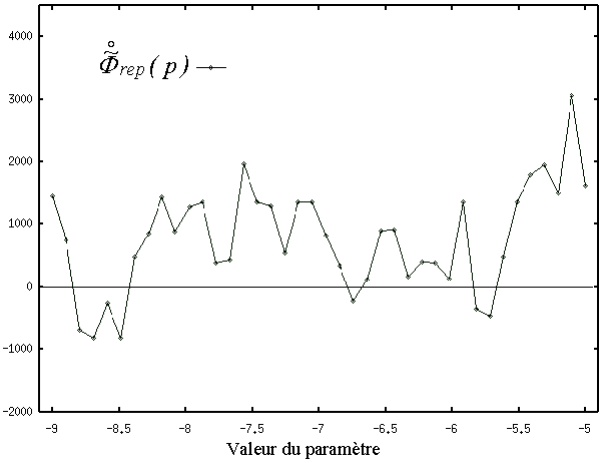


Figure 5.13 : *Valeurs de* $\overset{\circ}{\tilde{\Phi}}_{rep}$.

Ce cas a été résolu en considérant les deux passes de forgeage [Vieilledent 99], et en recherchant l'outil de préforme optimal à partir d'un tas plat.

5.3 Optimisation sous contraintes

Lorsque la fonctionnelle est soumise à des contraintes, le problème de minimisation devient plus difficile. Nous nous limiterons ici aux problèmes soumis à une seule contrainte d'égalité :

(5.6)
$$\begin{cases} \underset{p}{Min}\, \Phi(p) \\ h(p) = 0 \end{cases}$$

Φ désigne le critère d'optimisation principal et h la condition que le vecteur p solution doit vérifier. Nous envisagerons tout d'abord le cas où h est donnée par la fonction remplissage. Dans ce cas, on regardera le comportement de la méthode du Lagrangien augmenté et la méthode de programmation quadratique. Nous verrons ensuite le cas où h représente une condition géométrique. La minimisation est alors réalisée par l'algorithme du gradient réduit généralisé. Les différentes fonctions pouvant être d'ordres très différents, on les adimensionnalise par leur valeur à la première simulation :

(5.7)
$$\Phi^{ad(k)}(p) = \frac{\Phi^{(1)}}{\Phi^{(1)}} = 1 \quad \text{et} \ \forall k > 1 \qquad \Phi^{ad(k)}(p) = \frac{\Phi^{(k)}}{\Phi^{(1)}}$$

5.3.1. Optimisation sous contrainte de remplissage.

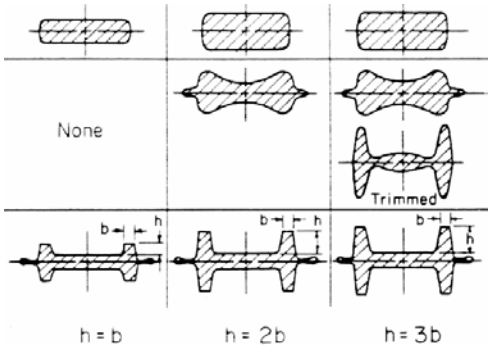


Figure 5.14 : *Gammes de forgeage pour différentes profondeurs de nervure [Altan 73].*

Le procédé étudié dans cette section concerne le forgeage d'une nervure axisymétrique. Généralement, le procédé débute par une phase de mise aux dimensions durant laquelle un lopin initial (cylindrique) de matière est écrasé entre deux tas plats. Cette opération permet d'éliminer les porosités de la matière. A partir de ce nouveau lopin, forger la pièce en une seule opération n'est pas toujours possible. Le nombre de passes intermédiaires nécessaires est fonction du rapport entre la hauteur et la largeur de la nervure. Dans le cas des déformations planes (Figure 5.14), lorsque ce rapport est égal à 2, une seule opération de préformage est généralement requise. Ce procédé, apparemment simple, présente quelques écueils. On peut en effet rencontrer deux types de défauts : des replis de matière lors de la passe de finition et un remplissage incorrect de la matrice finale. Ces deux défauts on été obtenus simultanément avec la préforme de la Figure 5.15.

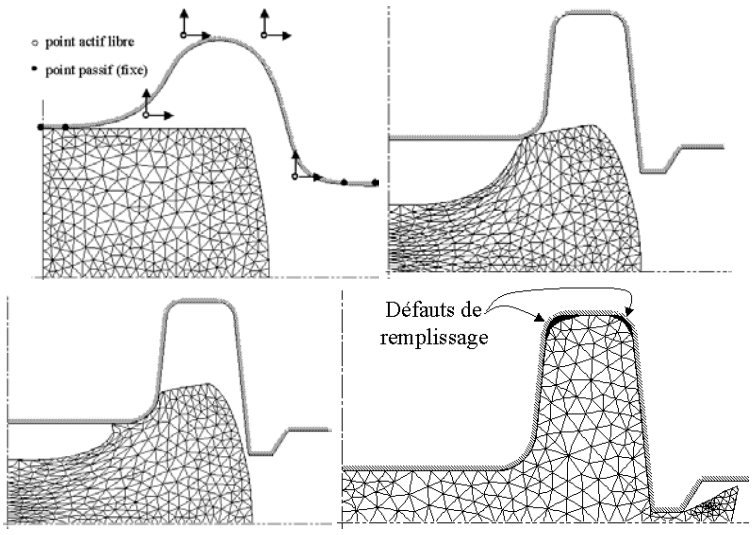


Figure 5.15 :

Haut : description de l'outil de préforme initial et préforme correspondante.

Bas : Défauts lors de la passe de finition : replis et remplissage incomplet de la matrice de finition.

Ce problème d'optimisation multi-objectifs a été abordé comme un problème d'optimisation sous contrainte. On cherche à éliminer les replis tout en imposant le remplissage de la matrice de finition:

$$\begin{cases} \underset{p}{Min}\, \Phi_{rep}(p) \\ \Phi_r(p) = 0 \end{cases} \tag{5.8}$$

où Φ_{rep} et Φ_r désignent respectivement la fonction coût repli (3.18) et la fonction coût remplissage (3.2). La surface libre se trouvant en bavure n'est prise en compte dans le calcul de la fonction repli. La description de la préforme comporte 8 paramètres actifs (Figure 5.15). Le problème (5.8) a été résolu en utilisant deux algorithmes : la méthode du Lagrangien augmenté et une méthode de programmation quadratique.

Dans les deux cas, les valeurs des paramètres ont dû être ajustées pour obtenir une convergence plus rapide. Le multiplicateur λ a pour valeur initiale : $\lambda^{(0)}=0$.

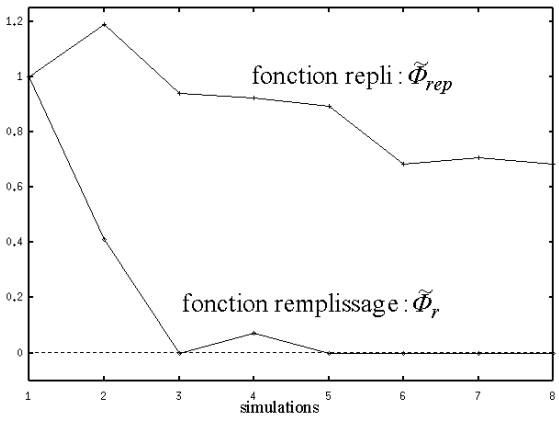


Figure 5.16 : *Algorithme du Lagrangien augmenté. Evolution des fonctions $\tilde{\Phi}_{rep}$ et $\tilde{\Phi}_r$.*

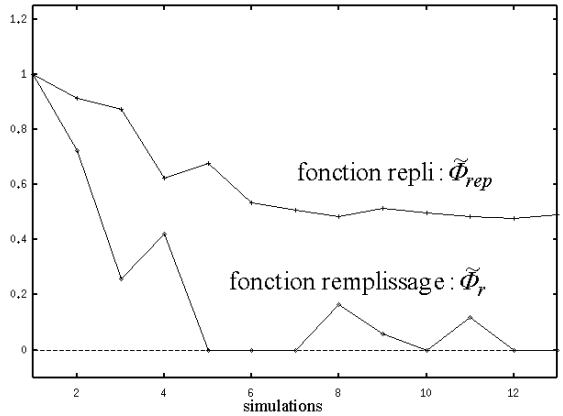


Figure 5.17 : *Méthode quadratique. Evolution de la fonction $\tilde{\Phi}_{rep}$ et de la contrainte $\tilde{\Phi}_r$.*

Pour la méthode du Lagrangien augmenté, les valeurs utilisées sont $\alpha=2$ ($r^{(k+1)}= \alpha\, r^{(k)}$). La vitesse de convergence ne semble satisfaisante que pour une valeur initiale élevée du coefficient de pénalisation ($r^{(0)}=10$). La solution est atteinte en 3 itérations (8 simulations).
Dans le cas de la méthode quadratique, le problème n'a été résolu qu'en imposant de faibles rayons de recherche linéaire. Ce qui explique que le nombre de simulations soit légèrement supérieur : 4 itérations (12 simulations).

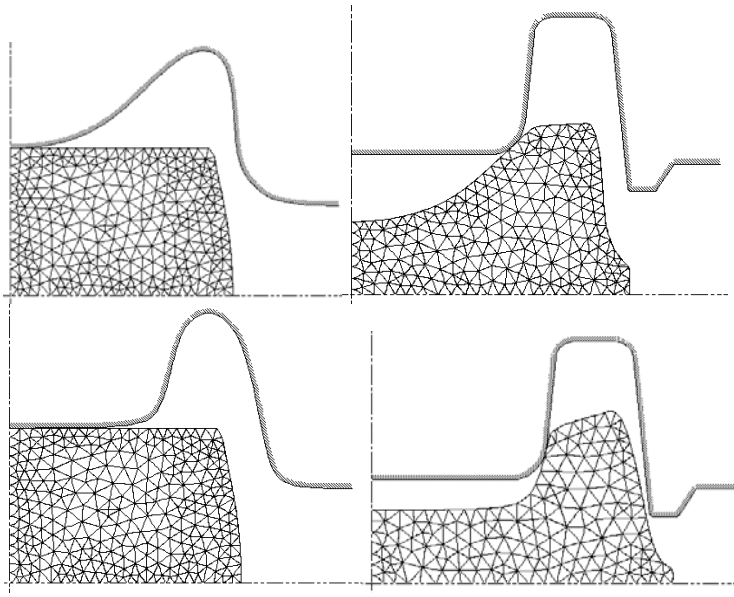


Figure 5.18 : *Gamme optimale déterminée par l'algorithme du Lagrangien augmenté.*

Figure 5.19 : *Gamme optimale déterminée par la méthode de programmation quadratique.*

Il n'y a pas unicité de la solution. Les deux préformes optimales ainsi déterminées sont très différentes. La plus faible valeur de $\widetilde{\Phi}_{rep}$ correspond à la gamme présentée sur la Figure 5.19. Cette solution est toutefois moins bonne du point de vue de l'écoulement. La matière arrive en effet plus vite en bavure. Le remplissage, qui n'est réalisé qu'à la toute fin du procédé, ne semble assuré que par la présence d'un surplus de matière conséquent.

Nous manquons d'expérience et de cas étudiés, pour tirer une conclusion définitive sur l'utilisation d'une méthode en particulier. Les deux algorithmes testés donnent de bons résultats, mais le problème est instable et présente plusieurs optima.

5.3.2. Optimisation sous contrainte de volume.

Le procédé présenté sur la Figure 5.20 comprend trois étapes, dont on ne considèrera que les deux dernières :

- Refoulement d'un lopin cylindrique entre tas plats (non représenté)
- Préformage
- Finition

La pièce forgée est ensuite usinée. La géométrie de la pièce utile est présentée sur la Figure 5.21. Les deux contours (forgé et usiné) sont par endroits très proches. La matrice de finition doit donc être parfaitement remplie en fin de procédé. Il ne s'agit pas stricto sensu d'un procédé en matrices fermées. Il existe en effet des freins de bavures dans lesquels la matière est susceptible de s'écouler. Toutefois, il n'existe pas de surplus de matière. Le volume du lopin est en effet strictement égal au volume utile des matrices de finition.

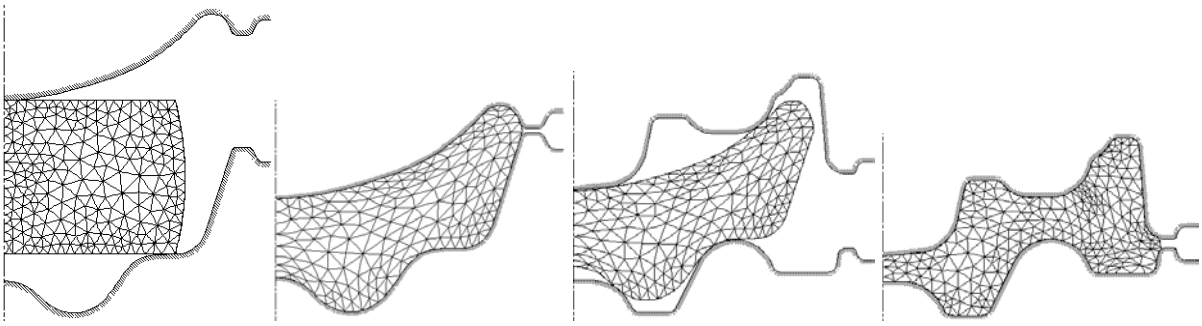


Figure 5.20 : *Gamme initiale de forgeage (préformage et de finition). Pas d'excès de matière.*

Le problème d'optimisation consiste à déterminer une préforme permettant d'obtenir une pièce forgée ayant une taille de grain homogène. Entre les deux dernières passes, la matière est soumise à un certain nombre de traitements thermiques qui induisent une modification importante de sa structure métallurgique. Les modifications de taille de grains dues à la déformation durant la phase de préformage sont ainsi partiellement gommées. En s'appuyant sur les indications des industriels, on considérera que la répartition de taille de grain au début de la passe de finition est homogène.

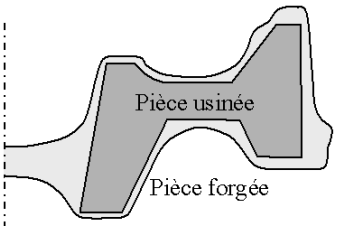


Figure 5.21 : *pièce forgée et forme de la pièce usinée*

Avec les arguments introduits au §3.1.3, ce problème est approché en étudiant la déformation totale. On cherche alors à obtenir une répartition homogène des déformations sur la pièce finale (ou plus exactement sur la partie utile de la pièce forgée). Pour rester cohérent avec ce qui a été mentionné plus haut, on négligera les déformations induites lors de la passe de préformage. Sous ces hypothèses, la répartition des déformations en fin de procédé (Figure 5.22), présente de fortes variations locales dans certaines zones de la partie utile. Ces fortes inhomogénéités deviennent plus flagrantes lorsque que l'on visualise $||grad\bar{\varepsilon}||$ (Figure 5.23).

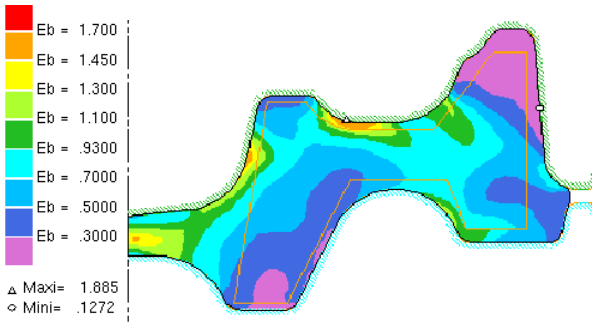


Figure 5.22 : *Isovaleurs de $\bar{\varepsilon}$ en fin de procédé, gamme initiale.*

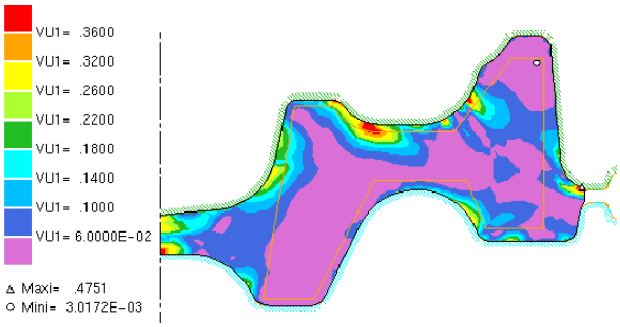


Figure 5.23 : *Isovaleurs de $||grad\bar{\varepsilon}||$, gamme initiale.*

On se propose donc d'optimiser le préforme existante afin de réduire ces variations locales. En conséquence, on utilise la fonction coût Φ_d^1 (3.10). Deux parties de l'outil de préforme sont décrits par des courbes Bsplines (Figure 5.24). Cette description comporte cinq paramètres actifs.

Tout comme pour l'outil de finition, le volume de l'outillage en position fermée doit être constant et égal au volume de matière. En toute rigueur, il suffit qu'il soit supérieur. Mais nous avons préféré traiter ce problème comme un procédé en matrice fermée et donc considérer une égalité.

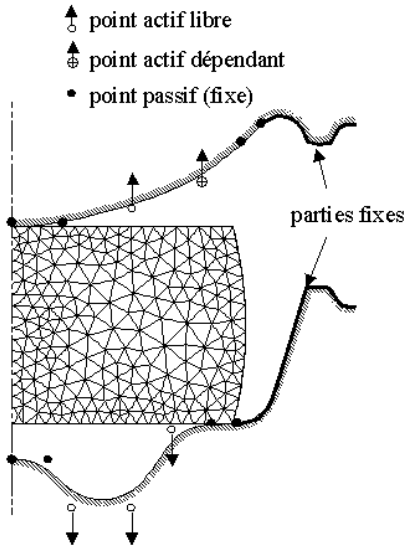


Figure 5.24 : *Description de l'outil de préforme.*

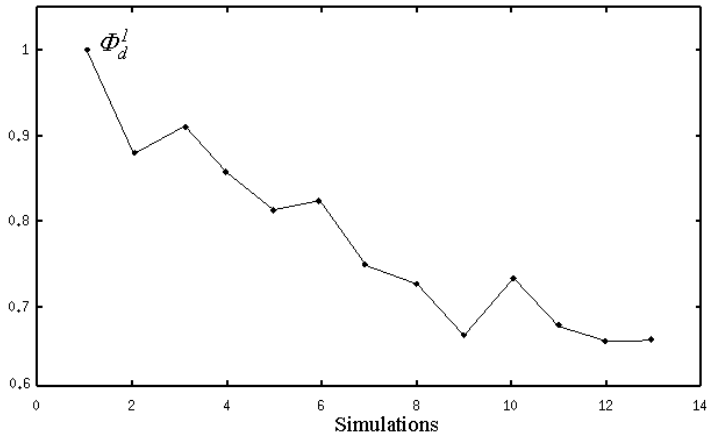


Figure 5.25 : *Evolution de la fonction coût.*

Dans cet exemple, la présence d'un frein de bavure sur l'outillage permet éventuellement à la matière de s'écouler en dehors de la partie utile de la matrice. Imposer un volume d'outillage égal au volume de matière ne permet donc pas d'imposer le remplissage correct des matrices. Toutefois, l'écoulement est simple et les formes d'outil douces, ce qui assure à la matière d'atteindre la zone périphérique à la fin du forgeage, lorsque le contact est déjà établi avec le reste de l'outillage.
Le problème d'optimisation s'écrit finalement :

$$\begin{cases} \underset{p}{Min}\, \Phi_d^l(p) \\ V_{out}(p) = V_0 \end{cases} \tag{5.9}$$

où V_{out} et V_0 désignent le volume de l'outil de préforme en position fermée, et le volume de matière. (5.9) est résolu par un algorithme du gradient réduit généralisé (cf. §3.3.2). La contrainte est ainsi strictement réalisée pour chaque préforme testée, pour un coût de calcul négligeable. Le paramètre choisi pour définir la base des variables dépendantes correspond au point actif le plus haut sur l'outil supérieur (Figure 5.24). Les paramètres rhéologiques du matériau sont donnés par :
K=3.158 10^5 Pa, n=0.11, $\bar{\varepsilon}_0$=0 et m=p_f=0.274. Le contact est glissant avec α_f=0.32.
La solution est atteinte après 4 itérations de l'algorithme, soit 13 simulations. La plus faible valeur de la fonction coût correspond en réalité à la 12^{ième} simulation. La préforme correspondante est présentée sur la Figure 5.26.

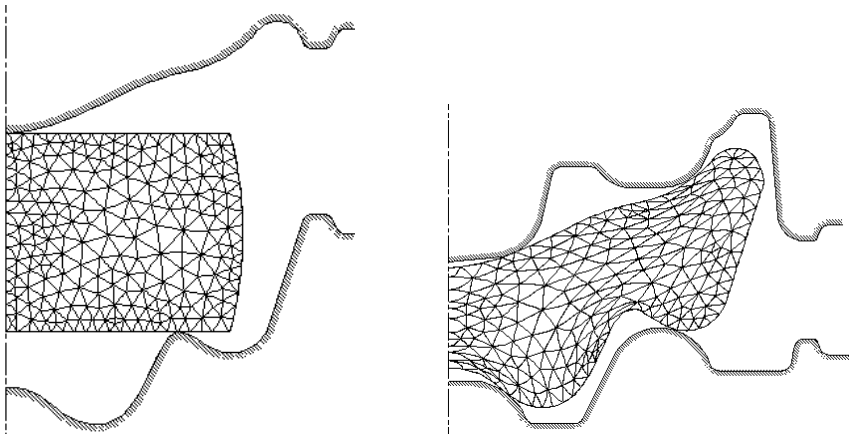


Figure 5.26 : *Outil de préforme optimal et préforme correspondante.*

Les répartitions en fin de forgeage de la déformation totale $\bar{\varepsilon}$, et la norme de son gradient $||grad\bar{\varepsilon}||$, sont représentées Figure 5.27 et Figure 5.28. La différence $\bar{\varepsilon}_{max}$ - $\bar{\varepsilon}_{min}$, initialement égale à 1.36, vaut 1.28 pour la gamme optimisée ; soit une diminution de 10%.

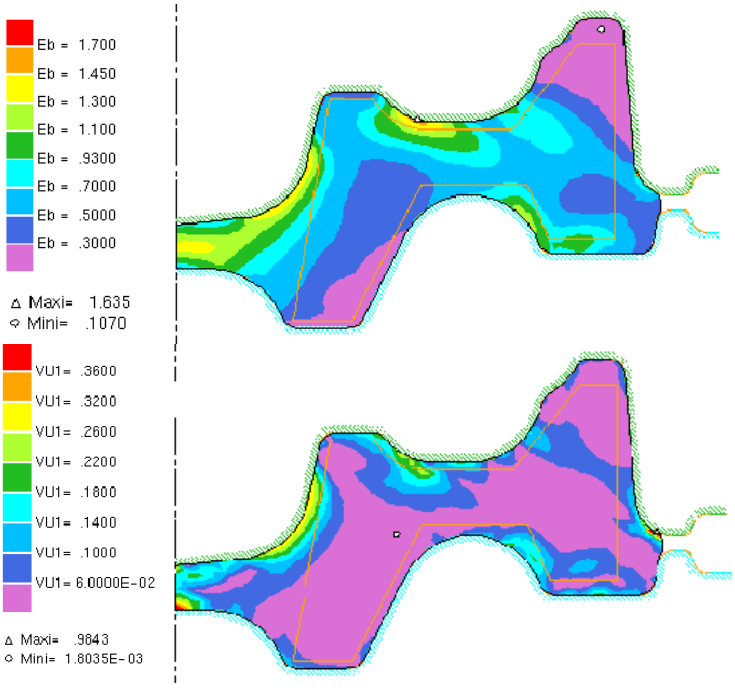


Figure 5.27 : *Répartition de $\bar{\varepsilon}$ pour la gamme optimisée.*

Figure 5.28 : *Valeurs de $||grad\bar{\varepsilon}||$ pour la gamme optimisée.*

Un problème similaire a été résolu dans le cas d'un procédé en matrices closes [Vieilledent 98b]

5.4 Le problème de conception

A partir d'un lopin initial de géométrie donnée, on désire obtenir, en deux passes de forgeage, une pièce finale présentant une taille de grain supérieure à une certaine valeur prescrite. Nous avons abordé ce critère d'optimisation en terme de déformation, en reformulant le problème par : trouver l'outil de préforme optimal permettant d'obtenir une pièce finale où la déformation soit supérieure à une valeur minimale $\bar{\varepsilon}_{\min}$.

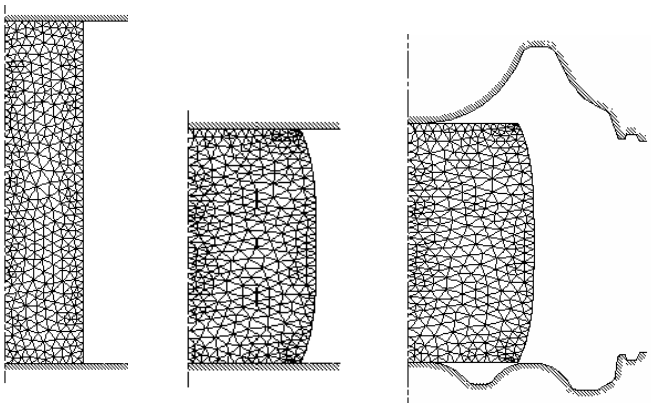
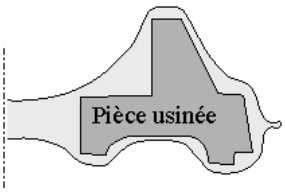


Figure 5.29 : *Gamme initiale, outil de préforme (tas plat) et outil de finition. Forme de la pièce après usinage.*



On se place dans le cadre d'un problème de conception pour lequel on ne possède pas d'idées préconçues sur le type de préforme à utiliser. La solution de départ la plus simple est alors un écrasement entre tas plats. La Figure 5.29 montre la géométrie du lopin, l'outil de préforme envisagé, ainsi que l'outil de finition, dont la course et la géométrie sont fixées. La pièce forgée est ensuite usinée, pour donner finalement le composant voulu. La phase de préformage correspond dans ce cas à 35% de déformation du lopin. La fonction coût utilisée est Φ_d^2 (3.14). Les coefficients définissant f sont : μ=*0.7*, $\bar{\varepsilon}_1$=0.3 et $\bar{\varepsilon}_2$=2.2.

La rhéologie de matériau est donnée par K=9.526 10^5 Pa, n=0.12 et m=p_f =0.27. Le contact est glissant avec α_f=0.3.

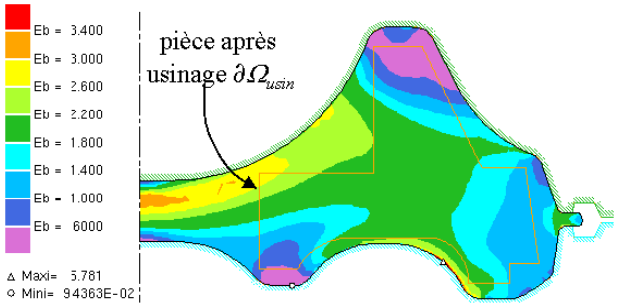


Figure 5.30 : *Répartition initiale des déformations sur la pièce finale ; forme de la pièce après usinage.*

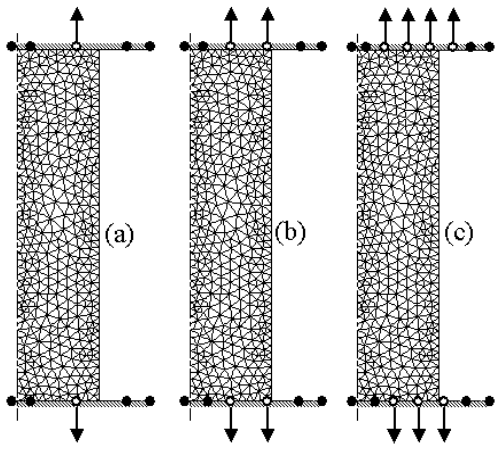


Figure 5.31 : *Différentes paramétrisations envisagées. (● : point passif, ○ : point actif)*

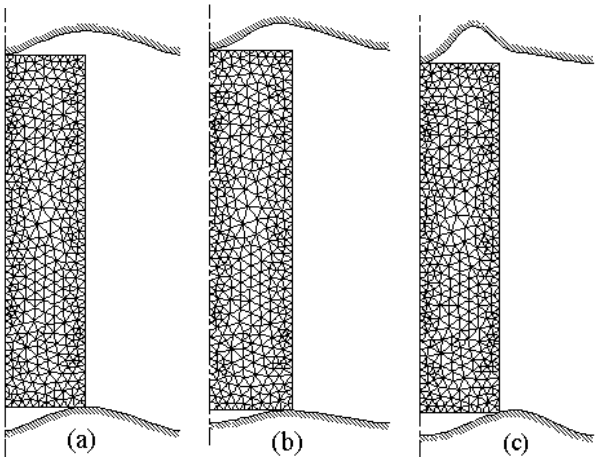


Figure 5.32 : *Préformes optimales correspondantes.*

On utilise plusieurs paramétrisations de l'outil initial : 2, 4 et 7 paramètres (Figure 5.31). Les positions des points fixes (•) sont identiques dans les trois cas. La répartition des

points actifs (∘) est uniforme sur l'intervalle compris entre deux points fixes.

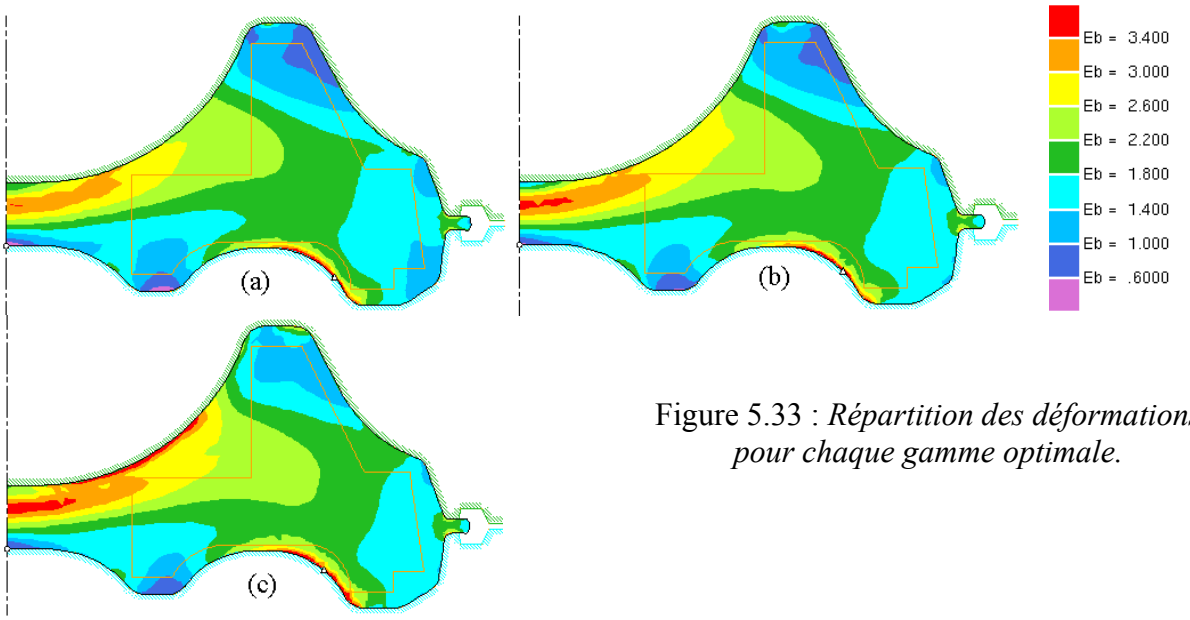


Figure 5.33 : *Répartition des déformations pour chaque gamme optimale.*

La Figure 5.34 montre l'évolution de la fonction, ainsi que de la valeur de $\bar{\varepsilon}$ sur Ω_{usin}, dans les trois cas envisagés. Les fonctions coût sont adimensionnalisées avec (5.7). Pour chaque cas, les valeurs obtenues sont résumées dans le tableau suivant.

	2 paramètres	4 paramètres	7 paramètres
Valeur minimale de Φ_d^2	0.51	0.43	0.38
Valeur de $\bar{\varepsilon}_{min}$ sur Ω_{usin} (valeur initiale : 0.52)	0.77	0.81	0.84
Nombre de simulations nécessaires	9	14	16

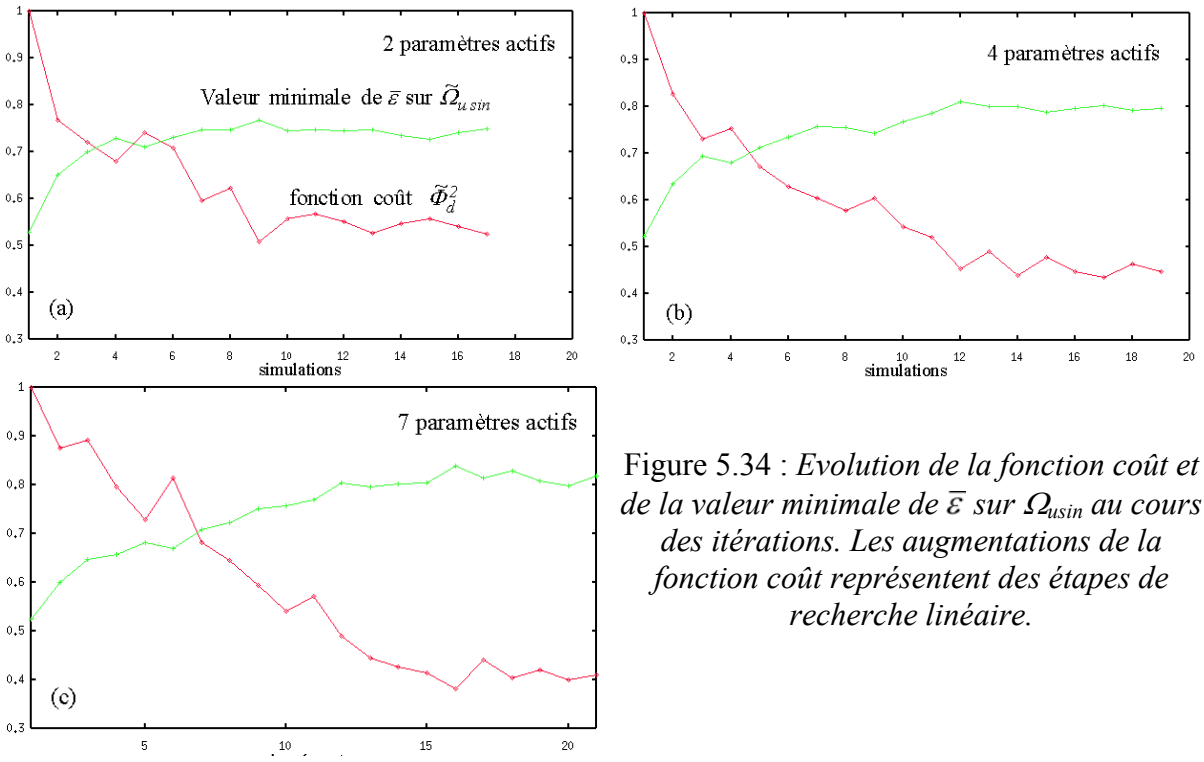


Figure 5.34 : *Evolution de la fonction coût et de la valeur minimale de $\bar{\varepsilon}$ sur Ω_{usin} au cours des itérations. Les augmentations de la fonction coût représentent des étapes de recherche linéaire.*

Dans le cas présent, le nombre de paramètres ne semble pas avoir d'influence sur le « type » de préforme solution (Figure 5.32). Néanmoins, les résultats et les formes obtenus sont différents. La plus forte valeur de $\overline{\varepsilon}_{min}$ correspond à la description la plus fine (7 paramètres). En contrepartie, le nombre de simulation est plus important.

Le logiciel s'applique bien aux problèmes de conception. A partir d'une géométrie initiale simple, il est possible de mener une première optimisation. La solution ainsi proposée peut ensuite servir de base de départ à une optimisation plus locale. De plus, l'utilisation d'un nombre important de paramètres permet de cerner les zones de l'outillage ayant une forte influence sur le résultat. Dans le cas présent, une description à 7 paramètres met en évidence l'importance de la zone supérieure près de l'axe, ce qui n'apparaît pas avec les deux autres descriptions.

Pour un problème d'optimisation, la forme générale des outils est connue. On ne désire alors effectuer que des modifications locales. Dans ce cas, l'analyse de sensibilité donne accès à des informations qui s'avèrent très intéressantes. Dans l'exemple suivant, on cherche à connaître l'influence de la profondeur des gravures sur la déformation en fin de forgeage. Le but est de pouvoir ensuite se livrer à une optimisation locale, à partir de ces informations. L'outillage est décrit par 5 paramètres actifs, définissant les hauteurs des gravures (Figure 5.35). Les dérivées de $\overline{\varepsilon}$ par rapport aux paramètres n°2 et n°3 sont reportées Figure 5.37 et Figure 5.38. On voit, par exemple, que le paramètre n°2 a une influence plus importante sur la partie inférieure de la pièce, où la déformation est la plus forte (Figure 5.36). L'optimisation devra donc préférentiellement porter sur cette portion d'outil, si on souhaite augmenter ou diminuer $\overline{\varepsilon}$ dans cette zone de la pièce. Dans ce cas, le signe de la dérivée (négatif sur cette partie de la pièce) permet de prévoir qu'une gravure moins profonde ($\Delta p > 0$) entraîne une diminution de la déformation.

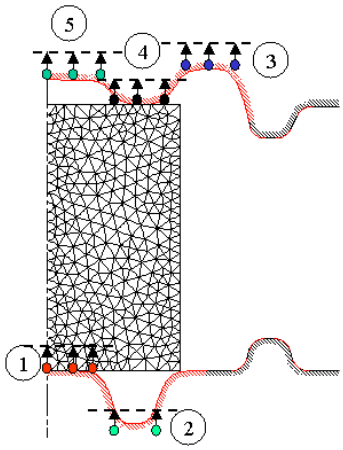


Figure 5.35 : *Description à 5 paramètres.*

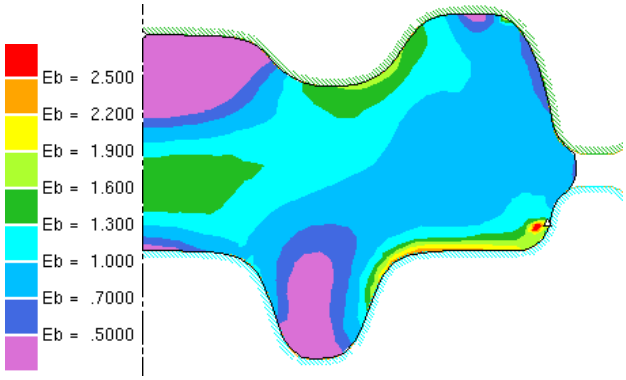


Figure 5.36 : *Répartition de $\overline{\varepsilon}$ en fin de forgeage.*

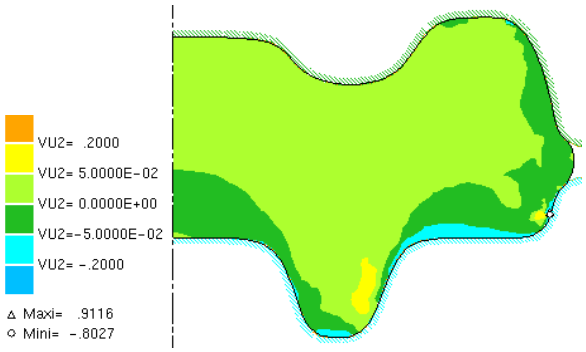


Figure 5.37 : *Sensibilité de $\bar{\varepsilon}$ au paramètre n°2.*

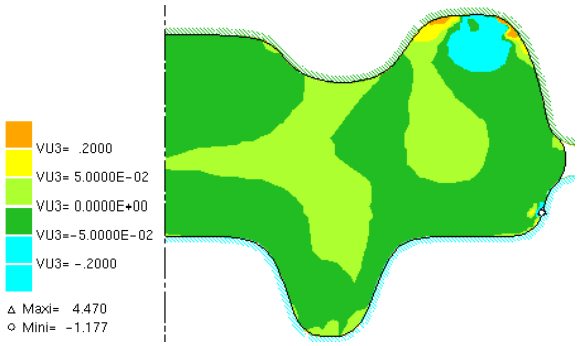


Figure 5.38 : *Sensibilité de $\bar{\varepsilon}$ au paramètre n°3.*

5.5 Forgeage d'un rotor aéronautique : défaut de repli

Le procédé présenté ici constitue un cas test. Il s'agit du forgeage en trois étapes d'un rotor. La première passe est un simple écrasement entre tas plats. On ne considère ici que les deux dernières opérations. La séquence initiale est présentée sur la Figure 5.39.

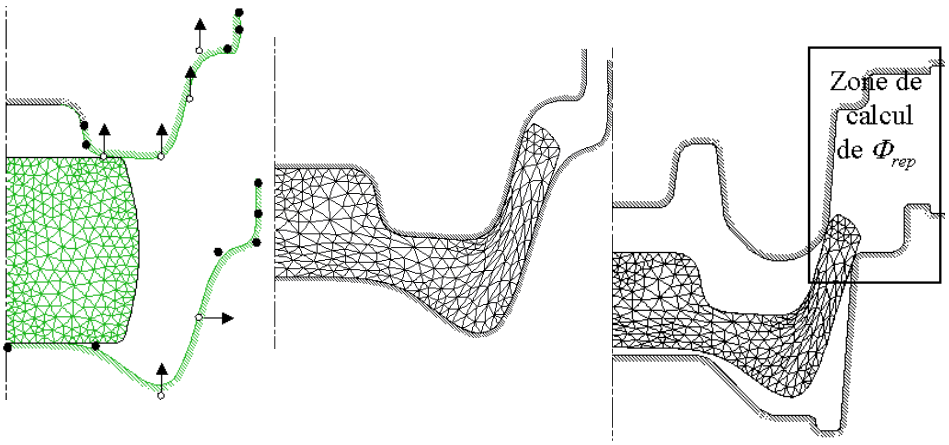


Figure 5.39 : *gauche : paramétrisation de la préforme. centre : préforme initiale. droite : début de la passe de finition.*

La rhéologie du matériau est donnée par K=1.4 10^5 Pa, n=0 et m=p_f =0.11. Le contact est glissant avec α_f=0.35.

Cette gamme de forgeage présente, lors de la dernière étape, plusieurs défauts de repli dans la zone périphérique (Figure 5.40). Ces défauts semblent provenir d'une mauvaise répartition de la matière au début de la dernière opération.

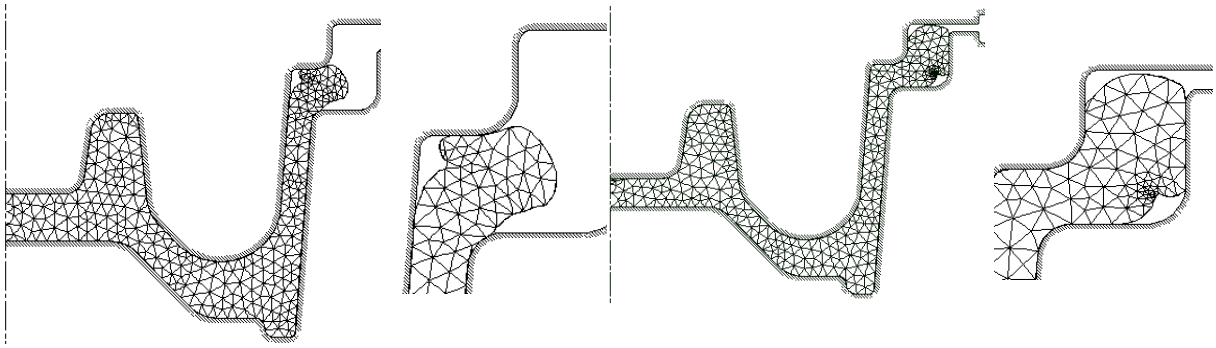


Figure 5.40 : *Différents défauts de repli lors de la passe de finition.*

On utilise la fonction coût repli Φ_{rep} (3.18). Cette dernière n'est calculée que pour la dernière passe de forgeage. De plus, le calcul de Φ_{rep} n'est réalisé que sur les surfaces libres appartenant à la zone indiquée sur la Figure 5.39 (où les défauts apparaissent). En effet, sur une surface ne présentant pas de repli, les valeurs de $\dot{\bar{\varepsilon}}$ sont très inférieures à celles que l'on peut calculer sur une surface de repli. Néanmoins, comme ces valeurs sont intégrées dans le temps, et que les temps d'existence des ces défauts sont courts (pour ce cas), les contributions venant des surfaces sans défauts sont, en fin de procédé, comparables à celles provenant des zones de repli. Les informations issues de la fonction et de son gradient, perdent de leur pertinence, ce qui peut aboutir à l'échec de la procédure d'optimisation.

Afin que l'utilisateur n'ait plus à intervenir pour définir les zones où les calculs doivent être effectués, il est possible d'améliorer la fonctionnelle existante, par exemple par l'adjonction d'un filtre. L'idée est alors de déterminer à partir de quelle valeur de $\dot{\bar{\varepsilon}}$, la surface correspondante est susceptible de donner naissance à un défaut. Cette étape constitue une des améliorations à apporter au logiciel existant, si l'on veut rendre la méthode plus systématique.

Avec les restrictions précédentes, ce problème d'optimisation a été résolu en 5 itérations (13 simulations). La gamme optimisée est présentée sur la Figure 5.41.

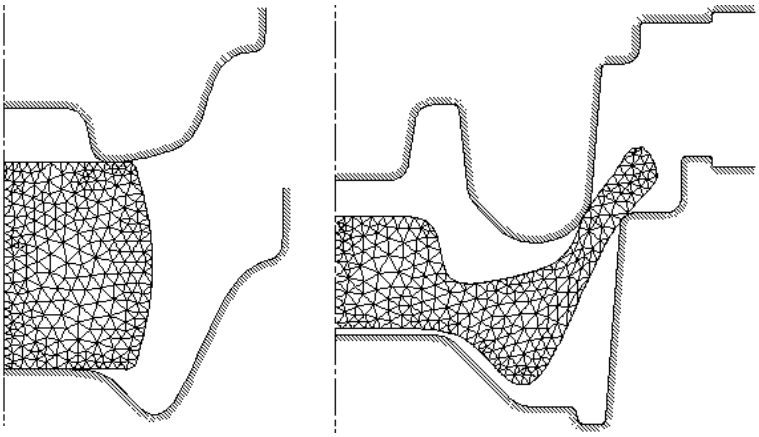


Figure 5.41 : *Outil de préforme optimisé et préforme correspondante.*

La Figure 5.42 montre l'écoulement pour différents instants de forgeage en fin de procédé. Les défauts de repli ont disparu.

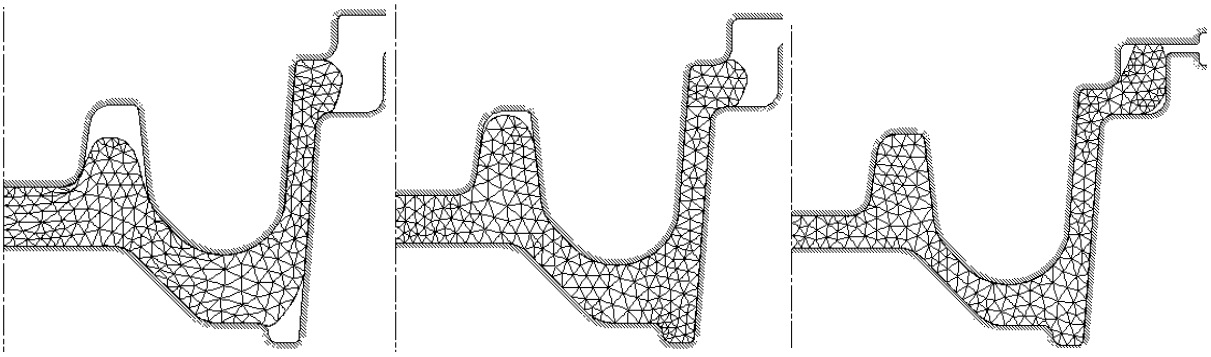


Figure 5.42 : *Instants intermédiaires de forgeage en fin de procédé – absence de repli.*

5.6 Elimination d'un défaut d'aspiration

Nous reprenons le procédé industriel présenté au §3.1.5. Le défaut d'aspiration dont il est question, constitue un cas particulier des défauts de repli. Nous avons vu qu'il est possible de définir un critère géométrique Φ_{rep}^{geo} (3.20), en considérant la distance sur l'axe de symétrie, entre la matière et l'outil.

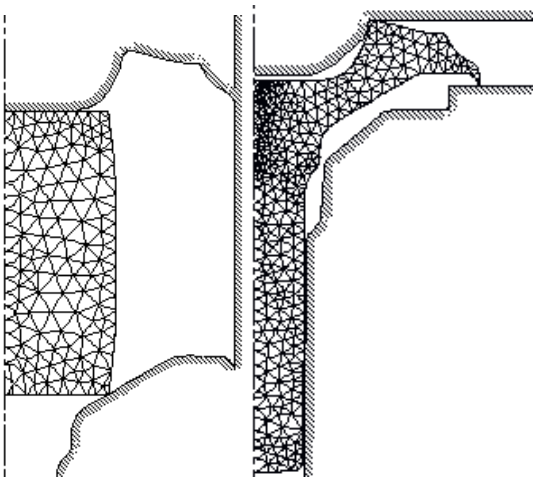


Figure 5.43 : *Gamme initiale – préformage et finition.*

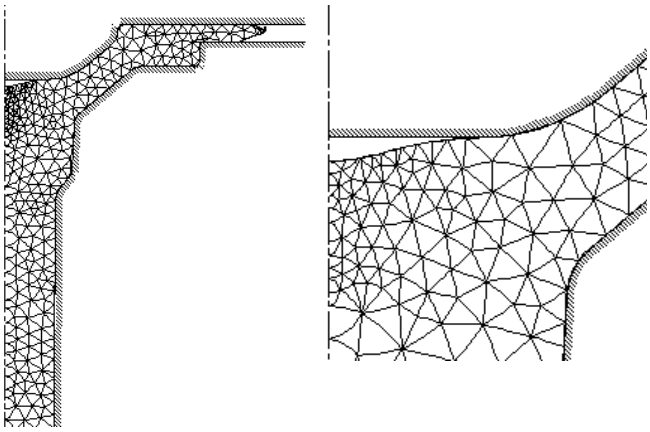


Figure 5.44 : *Décollement de la matière.*

Il s'agit ici d'un problème d'optimisation. Les parties susceptibles d'être modifiées sont fixées. En suivant les consignes données par l'industriel, l'outil de préforme est décrit à l'aide de deux courbes Bsplines, comportant 3 paramètres actifs.

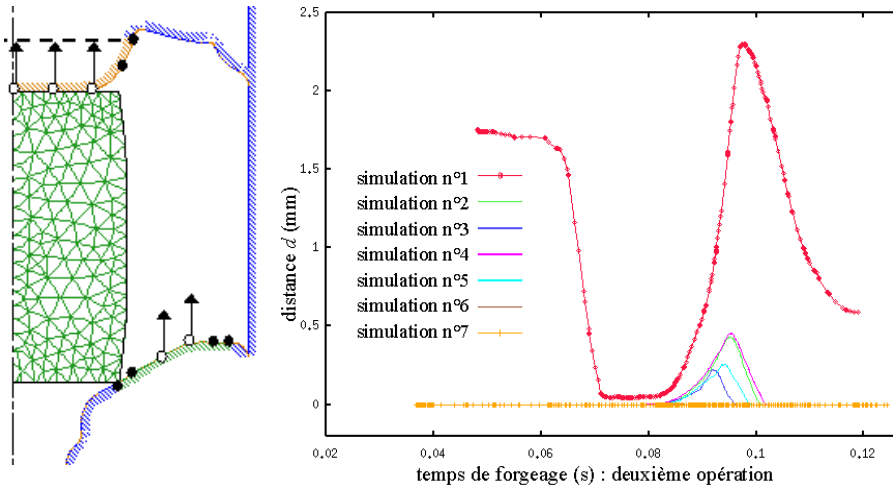


Figure 5.45 :

gauche : paramétrisation de l'outil de préforme.

droite : évolution dans le temps de la distance d sur l'axe, entre la matière et l'outil, en fonction des différentes préformes testées.

Les trois points de contrôle actifs situés sur l'outil supérieur, sont liés à un même paramètre, qui règle ainsi la hauteur de cette partie (Figure 5.45 – gauche).

La Figure 5.45 (droite) montre l'évolution de la distance *d* en fonction du temps de forgeage pour chaque gamme envisagée par la procédure d'optimisation. Après trois itérations (7 simulations), le phénomène de décollement a disparu, la matière reste en contact pendant toute la passe de finition. L'outil de préforme et la préforme correspondante sont reportés sur la Figure 5.46 (a-b).

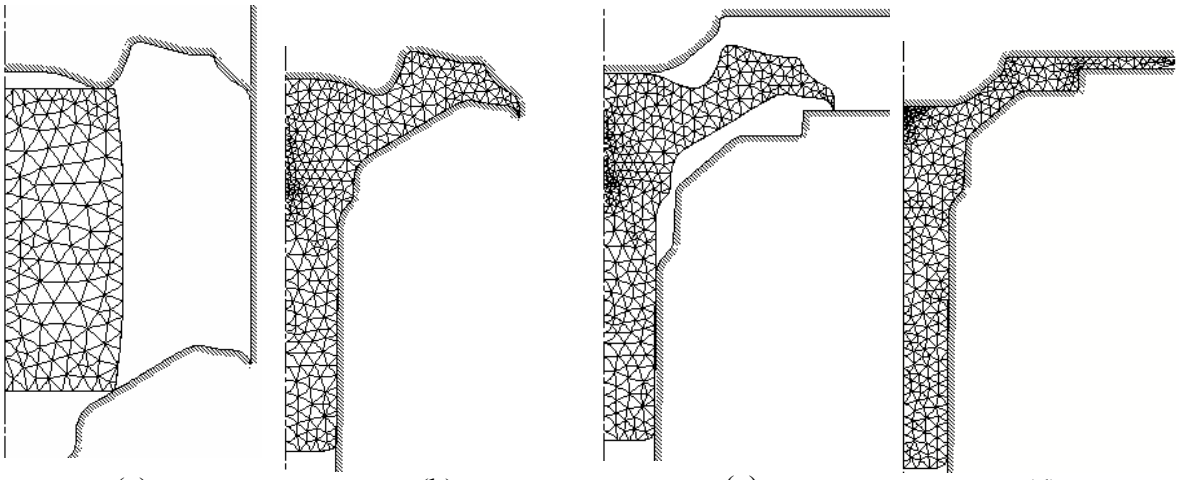


Figure 5.46 : *Gamme optimisée – absence de décollement pendant la dernière opération.*

Ce procédé présente néanmoins un défaut de repli lors de la première opération (Figure 5.47). Pour remédier à ce problème, plusieurs alternatives sont possibles. On peut par exemple repartir de la configuration initiale (Figure 5.43), en considérant un problème de minimisation sous contrainte :

$$\begin{cases} \underset{p}{Min}\, \Phi_{rep}(p) \\ \Phi_{rep}^{geo}(p) = 0 \end{cases} \tag{5.10}$$

où Φ_{rep} désigne la fonction coût repli (3.18).

Le problème d'aspiration étant résolu, une autre possibilité est de ne considérer que le problème de repli lors de la première opération. On part alors de la configuration obtenue précédemment (Figure 5.46 – a), mais en choisissant de nouveaux paramètres d'optimisation. L'idée est de déterminer une forme proche, pour conserver un écoulement sans défaut d'aspiration durant la dernière opération ; mais suffisamment différente pour pouvoir éliminer le nouveau défaut de repli.

Cette approche, qui permet de résoudre un problème de minimisation sans contrainte (à priori plus simple), a été utilisée avec succès sur ce cas. La paramétrisation choisie est présentée sur la Figure 5.48. Le caractère local de cette optimisation est assuré en imposant de faibles variations des paramètres. La gamme optimale, déterminée en seulement 4 simulations par la procédure d'optimisation, ne présentant plus aucun défaut.

Toutefois, la solution reste sujette à caution. La Figure 5.49 représente en effet la répartition de la vitesse de déformation $\dot{\bar{\varepsilon}}$, à un instant intermédiaire de forgeage, avec cette préforme optimisée. Si le repli n'est plus déclaré, on observe tout de même une légère concentration de $\dot{\bar{\varepsilon}}$, qui laisse supposer que le défaut est toujours potentiellement présent.

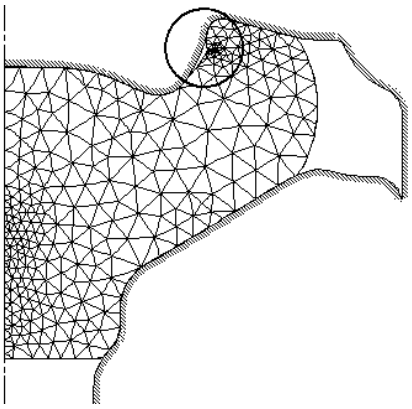


Figure 5.47 : *Apparition d'un repli lors de la première opération avec la gamme optimisée.*

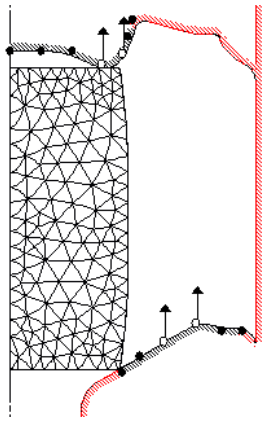


Figure 5.48 : *Paramétrisation pour la méthode d'optimisation locale.*

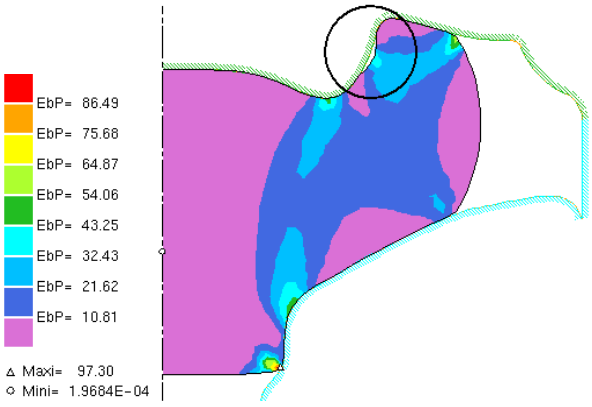


Figure 5.49 : *Préforme solution du problème local d'optimisation - Iso-valeurs de $\dot{\bar{\varepsilon}}$ à un instant intermédiaire de forgeage.*

Nous avons donc envisagé une méthode d'optimisation différente, en utilisant la seule fonction repli Φ_{rep}. Le défaut d'aspiration est en effet caractérisé, comme tous les défauts de repli que nous avons rencontré, par de fortes valeurs de $\dot{\bar{\varepsilon}}$. La paramétrisation utilisée pour l'outil de préforme est représentée sur la Figure 5.51. Le phénomène de décollement disparaît après 9 simulations. La préforme correspondante ne présente aucun défaut de repli.

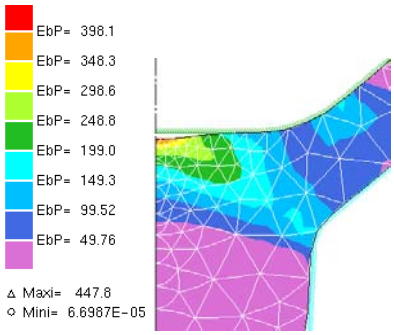


Figure 5.50 : *Défaut d'aspiration caractérisé par de fortes valeurs de $\dot{\bar{\varepsilon}}$.*

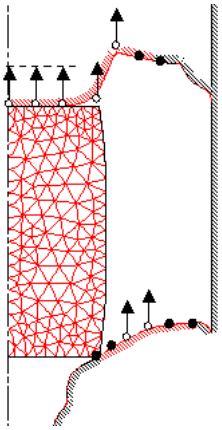


Figure 5.51 : *Description des paramètres d'optimisation*

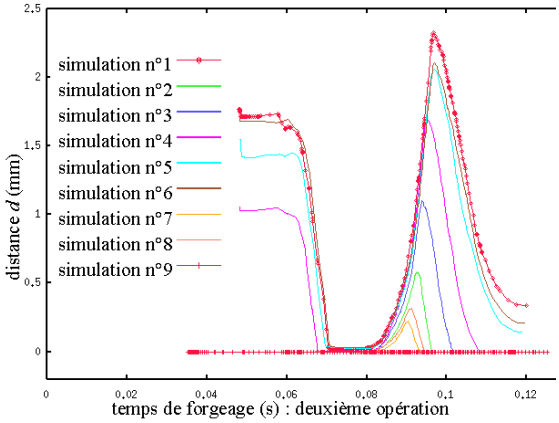
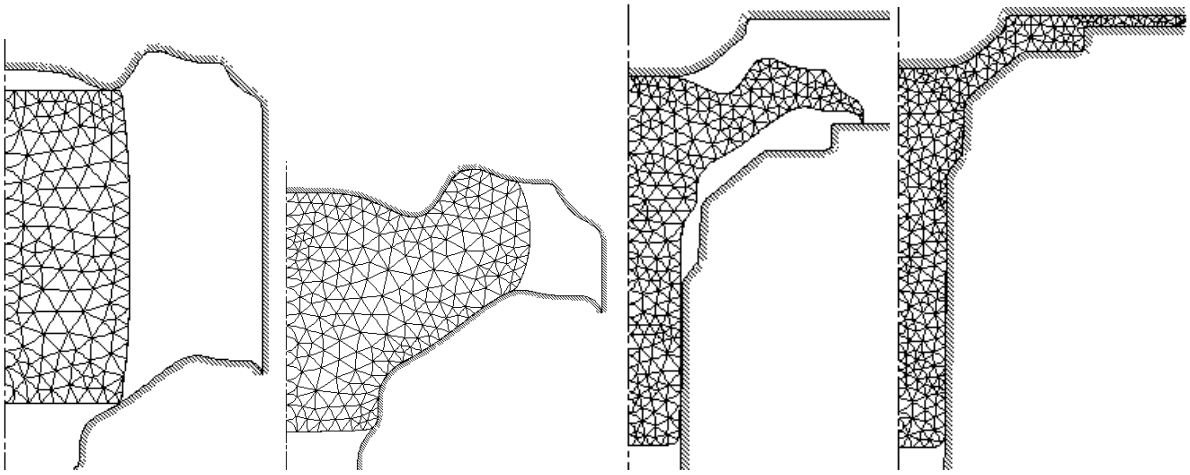


Figure 5.52 : *Evolution du décollement lors de la dernière opération, en fonctions des simulations.*

Le défaut de repli (lors de la première opération) est apparu à deux reprises ($5^{ième}$ et $7^{ième}$ simulation). La gamme optimale ainsi déterminée est représentée sur la Figure 5.53. On remarque que les formes sont très proches de celles obtenues à partir d'un critère géométrique (Figure 5.46). Les formes de l'outil supérieur de préforme semblent néanmoins plus douces, permettant ainsi un écoulement de la matière sans repli durant cette opération.



Première opération : début et instant intermédiaire.

Dernière opération : début et fin.

Figure 5.53 : *Gamme optimale déterminée en utilisant la seule fonction coût Φ_{rep}.*

La gamme solution présentée sur la Figure 5.53 est proche de celle mis en œuvre par les industriels, et déterminée grâce à leur expérience, par une méthode d'essais-erreurs. Les défauts qui ont été éliminés par le logiciel d'optimisation, ont réellement été observés lors d'essais effectués pour cette mise au point. La résolution de ces problèmes a nécessité un investissement important, en temps et en argent. Cet exemple est représentatif du potentiel de ce logiciel d'aide à la conception des procédés de mise en forme.

Conclusion

Ce travail avait pour objectif de développer un logiciel d'aide à la conception optimale en forgeage, qui puisse répondre aux attentes des industriels. La première version dont nous sommes parti, a permis de tester la faisabilité du projet sur quelques cas académiques. Néanmoins, pour être viable, le code d'optimisation devait pouvoir être appliqué avec succès à des procédés de mise en forme complexes et variés.

L'analyse de sensibilité a tout d'abord été fiabilisée par la prise en compte des dérivées de termes supplémentaires, tels les déformations ou les phases de calage des outils. Nous avons ainsi acquis une première expérience dans l'utilisation des logiciels de différentiation automatique.

La méthode de paramétrisation des contours des outils par des courbes Bsplines procure, de manière simple, une grande liberté dans la définition des paramètres géométriques. Elle a été utilisée pour optimiser plusieurs parties des outils de préforme ou de finition.

Le défaut de repli étant un point clé dans la conception des gammes de forgeage, nous avons mis en place une technique de maillage pour la détection et le traitement des replis. Plusieurs critères ont été étudiés. La fonction coût retenue a permis de traiter des défauts de natures apparemment différentes.

La qualité métallurgique des pièces forgées, qui constitue une préoccupation majeure dans de nombreux procédés, a été abordée en première approximation, au travers de deux critères portant sur la déformation.

Des algorithmes sont proposés dans le cas où le procédé est soumis à plusieurs critères, et en particulier pour les procédés en matrices fermées.

Le logiciel a prouvé son efficacité sur plusieurs procédés industriels portant sur des problèmes d'optimisation différents. Toutefois, dans sa forme actuelle, le code peut être amélioré sur différents points, par exemple en supprimant l'intervention de l'utilisateur dans le calcul de la fonction repli, ou en mettant en place des méthodes robustes pour le difficile problème des minima locaux.

Pour le développement futur de ce logiciel d'optimisation, différentes directions semblent intéressantes. La méthode développée pour l'analyse de sensibilité est générale, et peut être appliquée à d'autres critères d'optimisation. Il est ainsi possible d'étendre cette analyse aux variables thermiques, en différentiant le problème thermo-mécanique discret. Ceci permettrait de considérer de nouveaux paramètres, comme la température des outils ou du lopin, qui sont déterminants pour certains procédés.

L'utilisation de lois de comportement différentes, d'une résolution couplée avec les outillages (telle qu'elle existe sur certaines versions de Forge2®), ou la prise en compte de paramètres de procédé, apparaissent comme des perspectives privilégiées.

Du point de vue de l'analyse de sensibilité, ces nouveaux développements peuvent être effectués selon différentes méthodes : différentiation automatique ou méthodes semi-analytiques.

Enfin, nous nous sommes limité à l'optimisation d'un seul outil (préforme ou finition), mais il est possible d'envisager l'optimisation de plusieurs opérations.

Annexes

A.1) Construction des courbes Bsplines : méthode générale

La forme générale d'une approximation d'un polygone de sommets C_i, par une fonction Bspline d'ordre k, est donnée par :

$$X_B(s) = \begin{bmatrix} r_B(s) \\ z_B(s) \end{bmatrix} = \sum_{i=1}^{Nbsom} M_{i,k}(s) C_i = \sum_{i=1}^{Nbsom} M_{i,k}(s) \begin{bmatrix} R_i \\ Z_i \end{bmatrix} \quad s \in [s_{min}, s_{max}[\tag{A.1}$$

Les fonctions de base $M_{i,k}$ sont définies par la formule récursive de Cox-DeBoor [DeBoor 72] :

$$M_{i,1}(s) = \begin{cases} 1 & \text{si } u_i \le s < u_{i+1} \\ 0 & \text{sinon} \end{cases} \tag{A.2}$$

$$M_{i,k}(s) = \frac{s - u_i}{u_{i+k-1} - u_i} M_{i,k-1}(s) + \frac{u_{i+k} - s}{u_{i+k} - u_{i+1}} M_{i+1,k-1}(s) \tag{A.3}$$

avec la convention 0/0=0

Les valeurs u_i forment le vecteur nodal arbitraire de définition des fonctions de base et vérifient : $u_i \le u_{i+1}$. Les $M_{i,k}$ sont donc des fonctions C^{k-2} polynomiales de degré $k-1$ par morceaux et à support compact $[u_i, u_{i+k}]$. La courbe est décrite en faisant varier l'abscisse s de s_{min} à s_{max}, avec :

$$s_{min} = u_k \quad \text{et} \quad s_{max} = u_{Nbsom+1} \tag{A.4}$$

Ces fonctions de base vérifient de plus :

$$\forall s \in [s_{min}, s_{max}[\quad \sum_{i=1}^{Nbsom} M_{i,k}(s) = 1 \tag{A.5}$$

Lorsque les u_i sont choisis en progression arithmétique (le plus souvent de raison 1), la courbe Bspline est dite « uniforme ». Nous utiliserons par la suite des courbes Bsplines d'ordre 4 (degré 3), en omettant l'indice k dans (A.1).

Si on considère un polygone de définition à sept sommets (Figure A.2), et un vecteur nodal uniforme, les fonctions de base correspondantes sont représentées sur la Figure A.1. En tenant compte de (A.4), l'abscisse s possède quatre intervalles de définition $[u_4, u_5]$,…, $[u_7, u_8]$. Les portions correspondantes de la courbe Bspline sont représentées par des traits pleins ou pointillés sur la Figure A.2. Pour $s = u_4$, les seules fonctions de base non nulles sont M_1, M_2 et M_3. Le premier point de la courbe est donc défini par :

(A.6) $$X_B(u_4) = \begin{bmatrix} r_B(u_4) \\ z_B(u_4) \end{bmatrix} = M_1(u_4)\begin{bmatrix} R_1 \\ Z_1 \end{bmatrix} + M_2(u_4)\begin{bmatrix} R_2 \\ Z_2 \end{bmatrix} + M_3(u_4)\begin{bmatrix} R_3 \\ Z_3 \end{bmatrix}$$

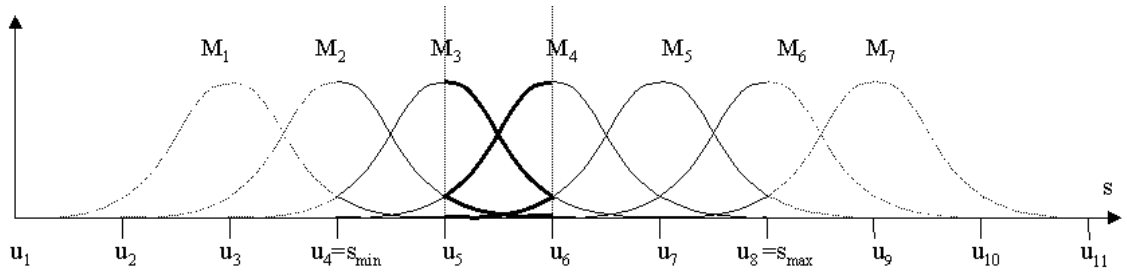


Figure A.1 : *Fonctions de base cubiques des courbes Bspline.*

La formule (A.6) montre clairement que le premier point $X_B(u_4)$ ne coïncide pas avec le premier point de contrôle C_1.

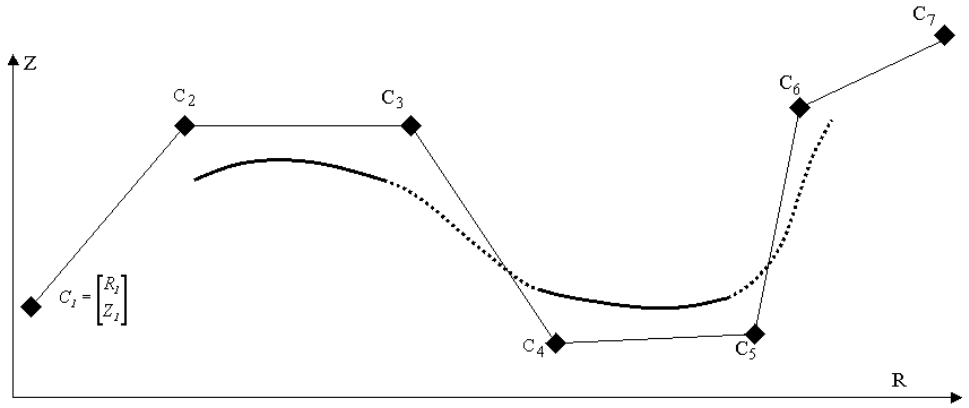


Figure A.2 : *Définition d'un contour polygonal à 7 sommets.*

Ce type de courbe présente la propriété de variation locale : la modification d'un point de contrôle C_i, n'influence que la portion de courbe correspondant à l'intervalle $[u_i, \ldots, u_{i+4}]$ où la fonction de base M_i est non nulle. Il s'agit ici d'un processus d'approximation et non d'interpolation, car la courbe ne passe pas par les points de définition. Dans le cadre de la description des outils de forgeage, une telle représentation n'est pas acceptable. Il est en effet nécessaire que les extrémités de la courbe Bspline coïncident avec les premiers et derniers points de contrôle, ceci afin d'assurer la continuité avec les parties fixes de l'outillage (fixes au sens du problème d'optimisation). De plus, il est préférable d'imposer que la courbe soit tangente aux segments extrêmes du polygone de définition, afin de contrôler les directions pour éviter les angles trop marqués dans la description du contour. L'imposition de telles conditions aux limites peut se faire de plusieurs manières. Nous avons utilisé des vecteurs nodaux uniformes « ouverts », caractérisés par [Rogers 89] :

(A.7) $$\begin{cases} u_1 = .. = u_4 = 0 \\ i = 1,..,Nbsom-3 : u_{i+4} = i \\ u_{Nbsom+1} = .. = u_{Nbsom+4} = Nbsom - 3 \end{cases}$$

Dans le cas d'un polygone à sept points de contrôle (Figure A.2), le graphe des fonctions de base correspondantes est reporté sur la Figure A.3 :

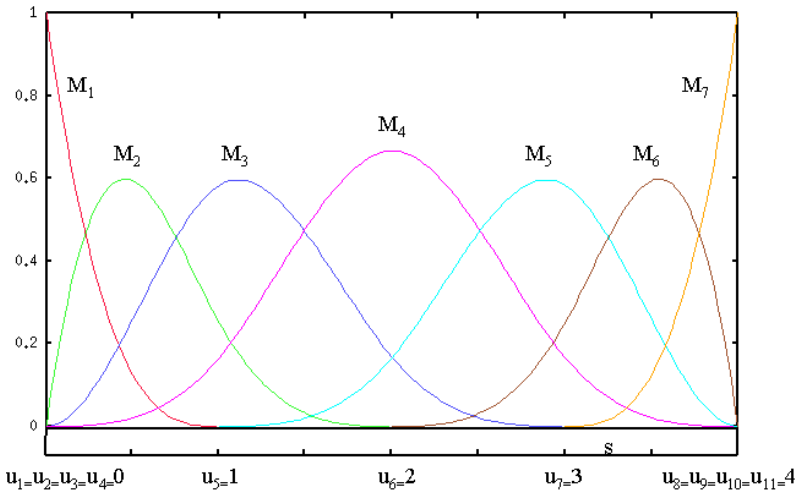


Figure A.3 : *Fonctions de base pour vecteur uniforme ouvert à 7 points de contrôle*

En prenant pour référence les notations de la Figure A.2, on remarque que les conditions aux limites sont alors respectées :

(A.8)
$$\begin{cases} M_1(0)=1,\ M_i(0)=0\ \ \forall i>1 \Rightarrow\ X_B(0)=C_1 \\ \dfrac{dM_2}{ds}(0)=-\dfrac{dM_1}{ds}(0)>0 \Rightarrow\ \dfrac{dX_B}{ds}(0)=\dfrac{dM_2}{ds}(0)(C_2-C_1) \end{cases}$$

et les formules similaires en $s=s_{max}=4$. La courbe finalement obtenue est reportée sur la Figure A.4.

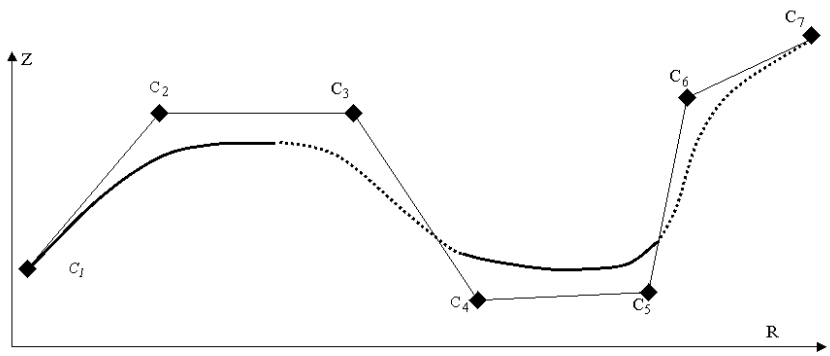


Figure A.4 : *Approximation par une Bspline cubique uniforme ouverte.*

A.2) Paramétrisation des contours : cas particuliers

Nous présentons ici deux cas particuliers de définition des Bsplines, qui peuvent être utiles lors de la description des outils à optimiser.

a) Définition de points caractéristiques liés à un même paramètre d'optimisation

On peut par exemple imposer que plusieurs points actifs soient liés au même paramètre. Cette option est intéressante, en particulier lorsque l'on cherche à optimiser la hauteur de l'outil sur l'axe de symétrie, tout en conservant une tangente horizontale (Figure A.5).

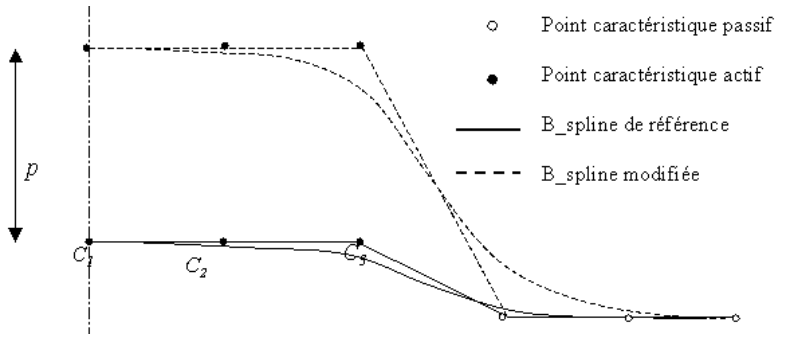


Figure A.5 : *Points de contrôle actifs liés à un seul paramètre.*

b) Définition de points caractéristiques multiples

La description d'une courbe Bspline à partir de (A.1) n'impose aucune condition sur les coordonnées des points de contrôle C_i. Il est possible de définir plusieurs fois le même point. Les différentes courbes obtenues en augmentant l'ordre de multiplicité d'un point de contrôle sont représentées sur la figure 2 :

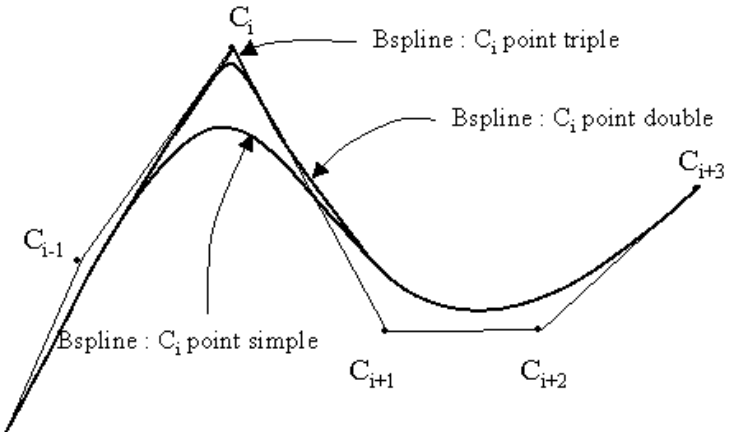


Figure A.6 : *Effet d'un point de contrôle multiple.*

Lorsque la multiplicité d'un point est égale au degré de la Bspline (ordre-1), la courbe passe alors par le point, et décrit exactement une portion des deux segments du polygone en C_i. Cette propriété est une simple conséquence de (A.1)-(A.3)-(A.5). La discontinuité de la normale ainsi générée, rend ce type de méthode inacceptable pour notre problème d'optimisation. Néanmoins, l'utilisation de points doubles s'avère très utile lorsque l'on cherche à approcher de manière C^2 une succession de segments. La figure 3 présente une courbe Bspline approchant au plus prés un polygone, dont tous les points de contrôle sont doubles.

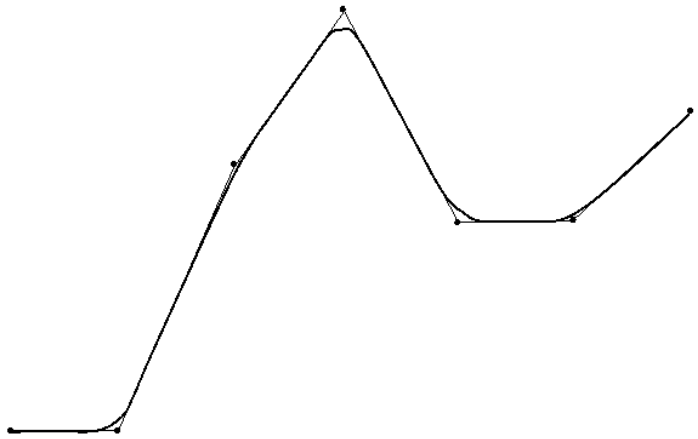


Figure A.7 : *Courbe pour un polygone à points de contrôle doubles.*

Cette manière de procéder constitue une bonne approximation de la paramétrisation à partir de segments et d'arcs.

A.3) Calcul et dérivation des gradients des fonctions de base

Les gradients des fonctions de base sur l'espace réel sont calculés de manière classique, à partir de l'espace de référence :

$$\begin{bmatrix} \frac{\partial N^k}{\partial \tilde{r}} \\ \frac{\partial N^k}{\partial \tilde{z}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial \tilde{r}} & \frac{\partial \eta}{\partial \tilde{r}} \\ \frac{\partial \xi}{\partial \tilde{z}} & \frac{\partial \eta}{\partial \tilde{z}} \end{bmatrix} \begin{bmatrix} \frac{\partial N^k}{\partial \xi} \\ \frac{\partial N^k}{\partial \eta} \end{bmatrix} = \left(J^{-1}\right)^T \begin{bmatrix} \frac{\partial N^k}{\partial \xi} \\ \frac{\partial N^k}{\partial \eta} \end{bmatrix} \tag{A.9}$$

où la matrice de passage J est donnée par :

$$J = \begin{bmatrix} \frac{\partial \tilde{r}}{\partial \xi} & \frac{\partial \tilde{z}}{\partial \eta} \\ \frac{\partial \tilde{r}}{\partial \xi} & \frac{\partial \tilde{z}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2\tilde{r}} \sum_k R_k^2 \frac{\partial N^k}{\partial \xi} & \frac{1}{2\tilde{r}} \sum_k R_k^2 \frac{\partial N^k}{\partial \eta} \\ \sum_k Z_k \frac{\partial N^k}{\partial \xi} & \sum_k Z_k \frac{\partial N^k}{\partial \eta} \end{bmatrix} \tag{A.10}$$

Les dérivées par rapport à la variable nodale X sont obtenues par :

$$\begin{bmatrix} \frac{d\left(\frac{\partial N^k}{\partial \tilde{r}}\right)}{dX} \\ \frac{d\left(\frac{\partial N^k}{\partial \tilde{z}}\right)}{dX} \end{bmatrix} = \frac{d\left(J^{-1}\right)^T}{dX} \begin{bmatrix} \frac{\partial N^k}{\partial \xi} \\ \frac{\partial N^k}{\partial \eta} \end{bmatrix} \tag{A.11}$$

La dérivée de la matrice inverse est calculée en remarquant que :

(A.12) $$J^T J^{-T} = I \;\Rightarrow\; \frac{dJ^T}{dX} J^{-T} + J^T \frac{dJ^{-T}}{dX} = 0$$

Soit encore :

(A.13) $$\frac{dJ^{-T}}{dX} = -\left(J^{-T} \frac{dJ^T}{dX} J^{-T} \right)$$

Avec pour finir :

(A.14) $$\frac{dJ}{dX} = \begin{bmatrix} -\frac{J_{11}}{\tilde{r}} \frac{d\tilde{r}}{dX} + \frac{1}{\tilde{r}} \sum_k R_k \frac{dR^k}{dX} \frac{\partial N^k}{\partial \xi} & -\frac{J_{11}}{\tilde{r}} \frac{d\tilde{r}}{dX} + \frac{1}{\tilde{r}} \sum_k R_k \frac{dR^k}{dX} \frac{\partial N^k}{\partial \eta} \\ \sum_k \frac{dZ^k}{dX} \frac{\partial N^k}{\partial \xi} & \sum_k \frac{dZ^k}{dX} \frac{\partial N^k}{\partial \eta} \end{bmatrix}$$

A.4) Dérivation des termes $\frac{\partial \tilde{r}\bar{J}}{\partial X}$ et $\frac{\partial \tilde{r}\bar{J}^s}{\partial X}$

Ces termes découlent des calculs effectués précédemment :

(A.15) $$\tilde{r}\bar{J} = (\tilde{r}J_{11}J_{22} - \tilde{r}J_{12}J_{21})$$

D'où

(A.16) $$\frac{d\tilde{r}\bar{J}}{dX} = (\frac{d(\tilde{r}J_{11})}{dX} J_{22} + \tilde{r}J_{11} \frac{dJ_{22}}{dX} - \frac{d(\tilde{r}J_{12})}{dX} J_{21} - \tilde{r}J_{12} \frac{dJ_{21}}{dX})$$

avec :

(A.17) $$\begin{cases} \frac{d(\tilde{r}J_{11})}{dX} = \sum_k R_k \frac{dR^k}{dX} \frac{\partial N^k}{\partial \xi} \\ \frac{dJ_{22}}{dX} = \sum_k \frac{dZ^k}{dX} \frac{\partial N^k}{\partial \eta} \\ \frac{d(\tilde{r}J_{12})}{dX} = \sum_k R_k \frac{dR^k}{dX} \frac{\partial N^k}{\partial \eta} \\ \frac{dJ_{21}}{dX} = \sum_k \frac{dZ^k}{dX} \frac{\partial N^k}{\partial \xi} \end{cases}$$

Le terme de bord $d\left(\tilde{r}\bar{J}^s\right)/dX$ est obtenu de manière similaire à partir de (4.6).

A.5) Dérivation du gradient des vitesses

A partir de l'interpolation du champ des vitesses (2.21), on écrit :

(A.18) $$\frac{\partial \tilde{v}_r}{\partial \tilde{r}} = -\frac{1}{\tilde{r}^2}\sum_k R_k V_k^r N^k + \frac{1}{\tilde{r}}\sum_k R_k V_k^r \frac{\partial N^k}{\partial \tilde{r}}$$

(A.19) $$\frac{\partial \tilde{v}_r}{\partial \tilde{z}} = \frac{1}{\tilde{r}}\sum_k R_k V_k^r \frac{\partial N^k}{\partial \tilde{z}}$$

(A.20) $$\frac{\partial \tilde{v}_z}{\partial \tilde{r}} = \sum_k V_k^z \frac{\partial N^k}{\partial \tilde{r}}$$

(A.21) $$\frac{\partial \tilde{v}_z}{\partial \tilde{z}} = \sum_k V_k^r \frac{\partial N^k}{\partial \tilde{z}}$$

Dans le cadre d'une formulation axisymétrique, le tenseur $\dot{\varepsilon}$ ne possède que quatre composantes non nulles. Il est alors facile de l'écrire sous forme vectorielle :

(A.22) $$\begin{bmatrix} \dot{\varepsilon}_{rr} \\ \dot{\varepsilon}_{zz} \\ \dot{\varepsilon}_{rz} \\ \dot{\varepsilon}_{\theta\theta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial v_r}{\partial r} \\ \dfrac{\partial v_z}{\partial z} \\ \dfrac{1}{2}\left(\dfrac{\partial v_r}{\partial z} + \dfrac{\partial v_z}{\partial r}\right) \\ \dfrac{v_r}{r} \end{bmatrix}$$

L'introduction au niveau discret de l'opérateur B, permet l'écriture en fonction du champ nodal des vitesses :

(A.23) $$\begin{bmatrix} \tilde{\dot{\varepsilon}}_{rr} \\ \tilde{\dot{\varepsilon}}_{zz} \\ \tilde{\dot{\varepsilon}}_{rz} \\ \tilde{\dot{\varepsilon}}_{\theta\theta} \end{bmatrix} = \sum_k B^k \begin{bmatrix} V_k^r \\ V_k^r \end{bmatrix}$$

(A.24) $$B^k = \begin{bmatrix} \dfrac{R_k}{\tilde{r}}\left(\dfrac{\partial N^k}{\partial \tilde{r}} - \dfrac{N^k}{\tilde{r}}\right) & 0 \\ 0 & \dfrac{\partial N^k}{\partial \tilde{z}} \\ \dfrac{R_k}{2\tilde{r}}\dfrac{\partial N^k}{\partial \tilde{z}} & \dfrac{1}{2}\dfrac{\partial N^k}{\partial \tilde{r}} \\ \dfrac{R_k}{\tilde{r}}\dfrac{N^k}{\tilde{r}} & 0 \end{bmatrix}$$

La dérivation des termes des éléments de B par rapport à X, implique les différents termes :

(A.25) $$\frac{dR^k}{dX^i_j} = \delta^m_k \delta^1_i \quad i = 1,2 \quad \forall k = 1, Nbnoe \quad \forall j = 1, Nbnoe$$

(A.26) $$\frac{\partial \widetilde{r}}{\partial X} = \sum\nolimits_k \frac{R^k}{\widetilde{r}} \frac{dR^k}{dX} N^k$$

ainsi que les dérivées des gradients des fonctions de base présentées au paragraphe précédent.

Bibliographie

[Aarts 89] E. H. L. Aarts, J. Korst. *Simmulated Annealing and Boltzman Mechanics*, Wiley& Sons, 1989.

[Abadie 79] J. Abadie, A. Haggag. Performance du gradient réduit généralisé avec une méthode quasi-Newtonienne pour la programmation non linéaire. *R.A.I.R.O. Recherche Opérationnelle*, 13(2), pp. 209-216, 1979.

[Abouaf 87] M. Abouaf. *Modélisation de la compaction des poudres métalliques frittées – approche par la mécanique des mileux continus.* Thèse de Doctorat, ENSMP, CEMEF, 1987.

[Altan 73] T. Altan, F. W. Boulger, J. R. Becker, N. Akgerman, H. J. Henning. *Forging Equipment, Materials and Practices.* National Technical Information Service, U. S. Departement of Commerce, Springfield, Virginia, 1973.

[Antùnez 96] H. J. Antùnez, M. Kleiber. Sensitivity of Forming Processes to Shape Parameters. *Comp. Meth. Appl. Mech. Engng*. 137, pp. 189-206, 1996

[Arentof 95] K. Arentoft, T. Wanheim. The basis for a design support system to prevent defects in forging. *Journal of Material Processing Tech*., 67, pp. 227-232, 1995.

[Armijo 66] L. Armijo. Minimization of Functions Having Continuous Partial Derivatives. *Pacific. J. Math.*, 16, pp. 1-3, 1966.

[Arora 91] J. S. Arora, A. I. Chahandre, J. K. Paeng. Mutliplier Method for Engineering Optimization. *Int. J. Num. Meth. Engng,* 32, pp. 1485-1525, 1991.

[Badrinarayanan 95] S. Badrinarayanan, A. Contentinescu, N. Zabaras. Preform Design in Metal Forming. Dans Dawson (éditeur), *Numiform95*, pp. 533-538. Balkema: Rotterdam, 1995.

[Badrinarayanan 96] S. Badrinarayanan, N. Zabaras. Sensitivity Analysis for the Optimal Design of Forming Processes. *Comp. Meth. Appl. Mech. Engng.*, 129(4) pp. 319-348. Balkema: Rotterdam, 1995.

[Bakshi 94] M. Bakshi-Jooyabari, I. Pillinger, T. A. Dean. Development of an Intelligent Knowledge-Based System (IKBS) for Forging Die Design. *J. of Mater. Proc. Technol.*, 45, pp. 689-694, 1994.

[Barlat 94] F. Barlat, K. Chung, O. Richmond. Anisotropic Plastic Portentials for Polycrystal and Application to the Design of Optimum Blank Shapes in Sheet Forming. *Metal. Mater. Transactions A*, 25, pp. 1209-1216, 1994.

[Bălan 95] T. Bălan, L. Fourment, J.-L. Chenot. Optimisation des gammes de forgeage – bilan des travaux. Rapport Technique, Action concertée de recherche pour la modélisation des opérations de forgeage, GT4, CETIM, juin 1995.

[Bălan 96] T. Bălan. *Optimisation de forme des outils de forgeage par méthode inverse*. Thèse de Doctorat, ENSMP, CEMEF, 1991.

[Becker 89] M. Becker, R. Kopp. A New Approach to Optimzation of Metal Forming Processes. Dans : E.G. Thompson et al. (éditeurs), *Numiform'89*, pp. 107-113. Balkema: Rotterdam, 1989.

[Bellet 90] M. Bellet, J.-L. Chenot, L. Fourment, E. Massoni, P. Montmitonnet et al. *Eléments finis et mise en forme des métaux. Séminaire de plasticité.* CEMEF, Sophia-Antipolis, 1994.

[Berg 95] J. M. Berg, J. Malas. Open-Loop Control of a Hot Forming Process. Dans Shen et al. (éditeurs), *5th Int. Conf. On Numer. Meth. In Ind. Forming Proc.*, Ithaca, New York, pp. 539-544, 1995.

[Bhavikatti 77] S.S. Bhavikatti, C.V. Ramakrishnan. Shape Optimization of Structural Systems using Finite Elements and Sequential Linear Programming. *Large Engineering Systems*, Avin Wexler Pergamon Press, pp. 224-235, 1977.

[Biglari 98] F. R. Biglari, N. P. O'Dowd, R. T. Fenner. Optimum Design of Forging Dies using Fuzzy Logic in Conjunction with the Backward Deformation Method. *Int. J. of Mach. Tools Manufacture*, 38, pp. 981-1000, 1998.

[Boër 85] C. R. Boër, H. Rydstad, G. Schröder. Choosing Optimal Forging Conditions in Iso-thermal and Hot-die Forging. *J. Appl. Metalworking*, 3(4), pp. 421-431, 1985.

[Bonnans 85] F. Bonnans, C. Saguez. *Méthodes numériques.* Cours de l'E.N.S.T.A., 1985.

[Bourdin 98] J. P. Bourdin, J. P. Bonnafé, J. Delmotte, E. Grosjean, J. M. Roelandt. Age Creep Forming Process Optimization for Aeronautical Structures. Computational Mechanics. Dans : S. Idelsohn et al. (éditeurs), *Computational Mechanics. New Trend and Applications*. pp. 1-16 Barcelone, Espagne, 1998.

[Box 87] G. E. P. Box, N. R. *Drapper. Empirical Model Building and response surfaces*. John Wisley & Sons, 1987.

[Broyden 70] C. G. Broyden.The convergence of a class of double-rank minimization algorithms. *J. Inst. Of Math. And its Appl.*, 6, pp. 222-231, 1970.

[Cardoso 88] J. B. Cardoso, J. S. Arora. Design Sensivity Analysis for Non Linear Structural Response. *AIAA J.*, 26(5), pp. 595-603, 1988.

[Cescutti 89] J. P. Cescutti. *Contribution à la simulation numérique du forgeage à chaud*. Thèse de Doctorat, ENSMP, CEMEF, 1989.

[Chamouard 66] A. Chamouard. *Static et dynamique appliquée au déformations à chaud, estampage & forge*. 1-2, Dunod, 1966.

[Chen 95] M. F. Chen, A. M. Maniatty. An Inverse Technique for the Optimization of Some Forging Processes. Dans : Shen et al. (éditeurs), *Simulation of Material Processing: Theory, Methods and Applications*. pp. 545-550, Balkema: Rotterdam, 1995.

[Chenot 93a] J.-L. Chenot, L. Fourment. Finite Element Modeling of Metal Forming: from Process Simulation to Physical Identification. Dans : Cross et al. (éditeurs), *Mathematical Modelling for Materials Processing*. pp. 43-65, Clarendon Press, Oxford, 1993.

[Chenot 93b] J.-L. Chenot. Finite Element Modeling and Optimization of Metal Forming Processes. Dans : *Second National Conference on Boundary and Finite Element*, Sibiu, Roumanie, 1993.

[Chenot 95] J.-L. Chenot, L. Fourment, M. Glowacki, T. Bălan. Rapport d'avancement des travaux. Optimisation automatique des gammes de forgeage. Rapport technique, CEMEF, Action concertée de recherche pour la modélisation des opérations de forgeage, GT4, CETIM, Juin 95.

[Choi 96] J. C. Choi, B. M. Kim, S. W. Kim, C. H. Kim. Computer-Aided Forging-Die Design for Two-Dimensional Cross-Sectional Components. *J. of Mater. Proc. Technol.*, 62, pp. 124-132, 1996.

[Chung 92a] K. Chung, O. Richmond. Ideal Forming - I. Homogeneous Deformation with Minimum Plastic Work. *Int. J. Mech. Sci*., 34(7), pp. 575-591, 1992.

[Chung 92b] K. Chung, O. Richmond. Ideal Forming - II. Sheet Forming with Optimum Deformation. *Int. J. Mech. Sci*., 34(8), pp. 617-633, 1992.

[Chung 97] J.S. Chung, S.M. Hwang. Application of a Genetic Algorithm to the Optimal Design of the Die Shape in Extrusion. *J. Mater. Process. Technol*., 72, pp. 69-77, 1997.

[Chung 98] S. H. Chung, S. M. Hwang. Optimal process design in non-isothermal, non-steady metal forming by the finite element method. *Int. J. for Num. Meth. In Engrg.*, 42, pp1343-1390, 1998.

[Coupez 91] T. Coupez. *Grandes transformations et remaillage automatique.* Thèse de Doctorat, ENSMP, CEMEF, 1991.

[Coupez 99] T. Coupez, A. I. Nouatin. Optimisation of Forming by using the Simplex Method and Preliminary Results on an Explicit 3D Viscoelastic Solution. Dans : J. A. Covas (éditeur), *2nd ESAFORM Conference,* pp. 477-480, Guimarães, 1999.

[DeBoor 72] C. DeBoor. On calculation with B-splines. *J. Approx. Theory,* 6, pp. 50-62, 1972.

[Di Lorenzo 98] R. Di Lorenzo, F. Micari. An Inverse Approach for the Design of the Optimal preform Shapes in Cold Forging. Dans *: J. L. Chenot et al. (éditeur), 1st ESAFORM Conference,* pp. 263-266, Sophia-Antipolis, 1998.

[Ebrahimi 97] S. A. Ebrahimi, D. A. Tortorelli, J. A. Dantzig. Sensitivity Analysis and Nonlinear Programming Applied to Investment Casting Design. *Appl. Math. Modelling*, 21, pp. 113-123, 1997.

[Fletcher 63] R. Fletcher, M. J. D. Powell. A Rapidly Convergent Descent Method for Minimization. *Computer Journal*, 6, pp. 163-168, 1963.

[Fletcher 64] R. Fletcher, C. M. Reeves. Function Minimization By Conjugate Gradient. *Computer Journal*, 7, pp. 149-154, 1964.

[Fletcher 75] R. Fletcher. An Ideal Penalty Function for Constrained Optimization. *J. Inst. Maths Applics.*, 15, pp. 319-342, 1975.

[Fortin 82] M. Fortin, R. Glowinski. *Méthodes de lagrangien augmenté.* Dunod, 1982.

[Fourment 92] L. Fourment. *Estimateurs d'erreurs et maillage adaptatif pour la simulation des procédés de mise en forme.* Thèse de Doctorat, ENSMP, CEMEF, 1992.

[Fourment 96a] L. Fourment, J. L. Chenot. Optimal Design for Non-steady-state Metal Forming Processes. – I. Shape Optimization Method. *Int. J. Num. Meth. Engng.,* 39(1), pp.33-51, 1996.

[Fourment 96b] L. Fourment, T. Bălan, J.L. Chenot. Optimal design for non steady-state metal forming processes - II Application of shape optimization in forging, *Int. J. for Num. Methods in Engineering*, 39(1), 1996, 51-66.

[Fourment 97] L. Fourment, D. Vieilledent, J.–L. Chenot. Automatic Design of Tools in Forging. Dans : *NAFEMS World Congress'*97, pp. 909-917, Stuttgart, 1997.

[Fourment 98a] L. Fourment, D. Vieilledent, J.–L. Chenot. Shape Optimization of the Axisymmetric Preform Tools in Forging using Direct Differantiation Method. *Int. J. of Forming Proc*., 1(4), pp. 399-423, 1998.

[Fourment 98b] L. Fourment, D. Vieilledent, J.-L. Chenot. The inverse Problem of Design in Forging. Dans : J. L. Chenot et al. (éditeur), *1st ESAFORM Conference,* pp. 271-274, Sophia-Antipolis, 1998.

[Fourment 99] L. Fourment, D. Vieilledent. Computer Aided Preforming tool shape Design in Hot Forging. Dans : *6th I.C.T.P Conference*, (à paraittre), Nuremberg, 1999.

[Francavilla 75] A. Francavilla, C.V. Ramakrishnan, O.C. Zienkiewicz. Optimization of Shape to Minimize Stress Concentration. *Journal of Strain Analysis,* 10(2), pp63-70, 1975.

[Gao 97] Z. Y. Gao, G. P. He, F. Wu. Sequential System of Linear Equations Algorithme for Nonlinear Optimization Problems with General Constraints. *J. of Optim. Theory and Appl*., 2, pp. 371-397, 1997.

[Gavrus 96] A. Gavrus. *Identification automatique des paramètres rhéologiques par analyse inverse.* Thèse de doctorat, ENSMP, 1996.

[Germain 85] Y. Germain. *Modélisation par éléments finis d'écoulement viscoplastique avec frottement – application au forgeage à chaud*. Thèse de Doctorat, Ecole des Mines de Paris, 1985.

[Ghouali 97] M. A. Ghouali. *Nouvel outil d'aide à la conception de forme optimale : analyse de sensibilité analytique locale.* Thèse de doctorat, ENS Cachan, 1997.

[Ghouali 96] A. Ghouali, S. Ortola, A. Oster, G. Duvaut. Local Sensitivity Analisys for the Forging Gamut Optimization. Dans : *Colloque IDMME96*, pp. 985-994, 1995.

[Ghouali 98] A. Ghouali, S. Ortola, A. Oster, G. Duvaut. Local Analitical Design Sensitivity Analisys of the Forging Problem. Dans : *Colloque IDMME96*, pp. 985-994, 1998.

[Ghouati 98a] O. Ghouati, J. C. Gelin. Identification of Material Parameters Directly from Metal Forming Processes. *J. Mater. Process. Technol*., 80-81, pp. 560-564, 1998.

[Ghouati 98b] O. Ghouati, J. C. Gelin. Optimisation of Metal Forming Process Parameters with Damage Minimization. *J. Mater. Process. Technol*., 80-81, pp. 597-601, 1998.

[Ghouati 99] O. Ghouati, J. C. Gelin, M. Baida, H. Lenoir. Simulation and Control of Hydroforming Processes for Tubes or Flanges. Dans : J. A. Covas (éditeur), *2nd ESAFORM Conference,* pp. 473-476, Guimarães, 1999.

[Gill 81] P. E. Gill, W. Murray, H. Wright. *Practical Optimization.* Academic Press, 1981.

[Glynn 95] D. Glynn, G. Lyons, J. Monaghan. Forging Sequence using an Expert System. *J. of Mater. Proc. Technol.*, 55, pp. 95-102, 1995.

[Goldberg 89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[Grandhi 93] R. V. Grandhi, A. Kumar, A. Chaudhary, J. Malas. State-Space Representation of Optimal Control of Non-Linear Materials deformation using the Finite Element Method. *Int. J. Num. Meth. Engng.*, 36, pp. 1967-1986, 1993.

[Guillaume 93] P. Guillaume, M. Masmoudi. Computation of High Order Derivatives in Optimal Shape Design. *Numeriche Mathematik*, 67, pp. 231-250, 1993.

[Haftka 92] R. T. Haftka, Z. *Elements of Structural Optimization*, Série : *Solid Mechanics and its Applications*, Vol. 11. Kluwer Academic Publisher, 3ième édition, 1992.

[Han 93] C. S. Han, R. V. Grandhi, R. Srinavasan. Optimum design of Forging Die Shapes using Nonlinear Finite Element Analysis. *AIAA Journal*, 31(4), 774-781, 1993.

[Hill 86] R. Hill. Extremal Paths of Plastic Work and Deformation. *J. Mech. Phys. Solids*., 34, pp. 511- 517, 1986.

[Hsu 94] Y. L. Hsu. A Review of Structural Shape Optimization. *Computer in Industry*, 26, pp. 3-13, 1994.

[Hsu 97] Q. C. Hsu, R. S. Lee. Cold Forging Design based on the Induction of analytical Knowledge. *J. of Mater. Proc. Technol.*, 69, pp. 264-272, 1997.

[Hwang 86] S. M. Hwang, S. Kobayashi. Preform Design in Disk Forging. *Int. J. Tool Des. Res.*, 26(3), pp. 231-243, 1986.

[Jensen 98] M. R. Jensen, F. F. Damborg, K. B. Nielsen, J. Danckert. Optimization of the Draw-Die Design in Conventionnal Deep-Drawing in Order to Minimise Tool Wear. *J. Mater. Process. Technol*., 83, pp. 106-114, 1998.

[Joun 91] M .S. Joun, S. M. Hwang. Optimal Process Design in Steady-State Metal Forming By Finite Element Method – I. Theorical Considerations. *Int. J. Mach. Tools Manufact.*, 33(1), pp. 55-61, 1993.

[Joun 98] M .S. Joun, M. Hwang. Die shape optimal design in three-dimensional shape metal extrusion by the finite element method. *Int. J. for Num. Meth. In Engrg.,* 41, pp. 311-335, 1998.

[Kim 95] H. S. Kim, Y. T. Im. Expert System for Multi-Stage Cold-Forming Process Design with a Re-Designing Algorithm. *J. of Mater. Proc. Technol.*, 54, pp. 271-285, 1995.

[Khun 51] W. W. Khun, A. W. Tucker. Nonlinear Programming. Dans : University of California Press, *2nd Berkeley Symposium on Mathematical Statistic and Probability*, pp. 481-492, Berkeley, 1951.

[Kopp 95] R. Kopp, R. Thomas, R. Bünten. Optimization of Multi-Stage Metal Forming Processes using Parrallel Computers. Dans : Shen et al. (éditeurs), *Simulation of Material Processing: Theory, Methods and Applications*, pp. ,45-449, Balkema: Rotterdam, 1995.

[Krishnakumar 89] K. Krishnakumar. Micro-Genetic Algotrithms for Stationary and Non-Stationary Functions Optimization. *SPIE-Intelligent control and adaptative system*, 1196, pp. 289-296, 1989.

[Kusiak 89] J. Kusiak, E. G. Thompson. Optimization techniques for Extrusion Die Shape Design. Dans: E. G. Thompson et al. (éditeurs), *Numiform'89*, pp. 569-574. Balkema: Rotterdam, 1989.

[Kusiak 94] J. Kusiak, EM. Pietrzyk, J.-L. Chenot. Die Shape Design and Evaluation of Microstructure Control in the Closed-Die Axisymmetric Forging by Using FORGE2 Program. *ISIJ International*, 34(9), pp. 755-760, 1994.

[Lapokov 97] R. Lapokov, P. F. Thompson. An Approach to the Optimal Design of Rolling Passes. *Int. J. Tools Manuf.*, 37(8), pp.1143-1154, 1997.

[Lapokov 98] R. Lapokov. Improvement of Die Life by Minimisation of damage Accumulaiton and Optimisation of Preform Design. *J. Mater. Process. Technol.*, 80-81, pp. 608-612, 1998.

[Legat 93] V. Legat, J. M. Marchal. Die Design: An Implicit Formulation for the Inverse Problem. *Int. J. Meth. Fluids*, 16, pp. 28-42, 1993.

[Liou 97] J. H. Liou, D. Y. Jang. Forging Parameter optimization Considerong Stress Distributions in Products Through FEM Analysis and Robust Design Methodology. *Int. J. Mach. Tools Manuf.* 37(6), pp. 775-782, 1997.

[Liszka 80] T. Liszka, J. Orkisz. The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics. *Computer and Structures*, 11, pp. 83-85, 1980.

[Mader 95] P. Mader. *Optimisation globale par différentiation automatique*. Rapport de stage n° SRT/95/21. CERFAC, 1995.

[Maniatty 96] A. M. Maniatty, M. F. Chen. Shape Sensitivity Analysis for Steady Metal Forming Processes. *Int. J. Num. Meth. Engng.*, 39(7), pp. 1199-1217, 1996.

[Maniatty 97] A. M. Maniatty, M. F. Chen. Algorithm for Optimization of Steady Forming. Dans : D. R. J. Owen et al. (éditeurs), *Computational Plasticity, Fundamental and Applications*. Barcelone, Espagne, 1997.

[Michaleris 94] P. Michaleris, D. A. Tortorelli, C. Vidal. Tangent Operator and Design Sensitivity Formulations for Transient Non-Linear Coupled Problems with Applications to Elastoplasticity. *Int. J. Num. Mech. Engng.*, 37, pp.2471-2499, 1994.

[Michalewicz 96] Z. Michalewicz. Genetic Algorithms+Data Structures=Evolution Programs. *Springer Verlag, New-York*, 3ième édition, 1996.

[Mihelič 98] A. Mihelič, B. Štok. Tool Design Optimization in Extrusion Processes. *Computers and Structures*, 68, pp. 283-293, 1998.

[Miles 95] M. P. Miles, L. Fourment, J.-L. Chenot. Incremental Mass conservation and Adaptative Remeshing for the Thermo-Mechanical Coupling between Workpiece and Tool in Non Steady-State Metal Forming. Dans :

Shen et al. (éditeurs), *Simulation of Material Processing: Theory, Methods and Applications*, pp. 431-434, Cornell Univ. Ithaca, juin 1995.

[Minkowycz 88] M. Minkowycz, W. J. Sparrow, G. E. Schneider, R. H. Pletcher. *Handbook of Heat Transfert*, Wisley&Sons, 1998.

[Minoux 83] M. Minoux. *Programmation mathématique : théorie at algorithmes*. Collection Technique et Scientifique des Télécomunications. Tome 1, ed. Dunod , 1983.

[Morris 82] A. J. Morris. *Foundations of Structural Optimization: A unified approach*. Wiley Series in Numerical Methods for Engineering. Wiley&Sons, Chichester, 1982.

[Myers 95] H. Myers, C. Montgomery. *Response Surface Methodology: Process and Product. Optimization Using Designed Experiments*. Wiley-Interscience, 1995.

[Nakamashi 98] E. Nakamashi, T. Honda. Optimum Design of Sheet Forming Process by Using Finite Element and Discretized Optimization Method. *Int. J. of Forming Proc.*, 1(2), pp. 163-185, 1998.

[Navarrina 98] F. Navarrina, S. L. Fontan, I. Colominas, E. Bendito, M. Casteleiro. High Order Shape Design Sensitivity an Unified Approach. Dans : S. Idelsohn et al. (éditeurs), *Computational Mechanics. New Trend and Applications*. Barcelone, Espagne, 1998.

[Nelder 65] J.A. Nelder, R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7, pp. 308-313, 1965.

[Noiret 96] C. Noiret, F. Lauro, D. Lochegnies, J. Oudin. Otimisation by Inverse Method : Application to the Forming of Hollow Glass Items. Dans *: J. L. Chenot et al. (éditeur), 1st ESAFORM Conference,* pp. 437-440, Sophia-Antipolis, 1998.

[Oh 94] S. I. Oh, S. M. Yoon. A new Method to Design Blockers. *Annals of the CIRP*, 43(1), pp. 245-248, 1994.

[Ohata 98] T. Ohata, Y. Nakamura, T. Katayama, E. Nakamachi, N. Omori. Improvement of Optimum Process Design System by Numerical Simulation. *J. Mater. Process. Technol.*, 80-81, pp. 635-641, 1998.

[Park 83] J. J. Park, N. Rebelo, S. Kobayashi. A New Approach to Preform Design in Metal Forming with the Finite Element Method. *Int. J. Tool Des. Res*., 23(1), pp. 71-79, 1983

[Patnaik 97] S. N. Patnaik. A Cascade Optimization Strategy for Solution of Difficult Design Problem. *Int. J. Num. Meth. Engng*., 40, pp. 2257-2266, 1997.

[Pironneau 84] O. Pironneau. *Optimal Shapes Design for Elliptic Systems*. Springer-Verlag, 1984.

[Powell 64] M.J.D. Powell. An Efficient Method for Finding the Minimum of a function of Several Variables without Calculating Derivatives. *Computer Journal,* 7, pp.155-162, 1964.

[Powell 75] M.J.D. Powell. Restart Procedure for the Conjugate-Gradient Method. *Math. Prog.*, 2, pp. 241-254, 1975.

[Powell 78] M. J. D. Powell. Algorithms for Nonlinear Constraints that use Lagrangian Functions. *Math. Prog.*, 14, pp. 224-248, 1978.

[Press 92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical recipes in Fortran. The Art of Scientific Computing*. Cambridge University Press, 2ième édition, 1992.

[Rogers 89] D. F. Rogers, J. A. Adams. Mathematical elements for computer graphics. Second Edition, McGraw-Hill Publishing Company, 1989.

[Rostaing 93] N. Rostaing-Schmidt. *Différentiation automatique: Applicatioon à un problème d'optimisation en météorologie.* Thèse de Doctorat, Université Nice-Sophia-Antipolis, 1993.

[Roy 94] R. Roy, K. Chodnikiewicz, R. Balendra. Interpolation of Forging Preform Shapesusing Neural Networks. *J. of Mater. Proc. Technol.*, 45, pp. 695-702, 1994.

[Roy 97] S. Roy, S. Ghosh, R. Shivpuri. A New Approach to Optimal Design of Multi-Stage Metal Forming Processes with Micro-Genetic Algorithms. *Int. J. Mach. Tools Manufact.,* 37(1), pp. 29-44, 1997.

[Santos 98] M. T. Santos, J. Lama. Numerical Optimzation of Forging Processes using the Response Surface Approach. Dans *: J. L. Chenot et al. (éditeur), 1st ESAFORM Conference,* (papier hors édition), Sophia-Antipolis, 1998.

[Shimmerling 98] P. Shimmerling, J. C. Sisson, A. Zaidi. *Pratique des plans d'expérience.* TEC & DOC, 1998.

[Schittkowski 80] K. Schittkowski. *Nonlinear Programming Codes. Informations, Test, Performance*. Lecture Notes in Economic and Mathematical System. Springer-Verlag, Berlin, Heidelberg, New-York, 1980.

[Schittkowski 85] K. Schittkowski. *Computational Mathematical Programming*. NATO ASI Series, Vol. F15, Springer-Verlag, Berlin, 1985.

[Schittkowski 94] K. Schittkowski, C. Zillober, R. Zotemal. Numerical Comparison of Nonlinear Programming Algorithms for Structural Optimization. *Structural Optimization*, 7, pp. 1-19, 1994.

[Smelser 86] R. E. Smelser, O. Richmond. A numerical study of the effect of die profile on extrusion. Dans : Thompson et al. (éditeurs), *Numiform'86*, pp. 305-312, Balkema: Rotterdam, 1986.

[Soyris 90] N . Soyris. *Modélisation tridimensionnelle du couplage thermique en forgeage à chaud*. Thèse de Doctorat, ENSMP, CEMEF, 1990.

[Srinavasan 94] R. Srinavasan, G. H. K. Reddy, S. S. Kumar, R. V. Grandhi.Intermediate Shapes in Closed Die Forging by the Backward Deformation Optimization Method (BDOM). *ASM J. Mater. Engng Performance,*3, pp. 501-513, 1994.

[Srinavasan 90] R. Srinavasan, J. S. Gunasekara, H. L. Gegel, S. M. Doraivelu. Extrusion Trough Controlled Strain Rate Dies. *J. Mater. Shaping Technol.,*8, pp. 133-141, 1990.

[Talbi 96] E. G. Talbi, P. Bessière, J. Ahuactzin, E. Mazer. *Un algorithme génétique paralléle pour l'optimisation.* Technique et science informatiques, 15(8), pp. 1105-1129, 1996.

[Tortorelli 94] D. A. Tortorelli, P. Michaleris. Design Sensitivity Analysis : Overview and Review. *Inverse Problems in Engineering*, 1, pp. 71-105, 1994.

[Tronel 93] Y. Tronel. *Contribution à la modélisation thermo-mécanique et physique du forgeage à chaud*. Thèse de Doctorat, ENSMP, CEMEF, 1993.

[Ulysse 99] P. Ulysse. Optimal Extrusion Die Design to Achieve Flow Balance. *Int. J. Mach. Tools Manuf*, 39, pp. 1047-1064, 1999.

[Vieilledent 98a] D. Vieilledent, L. Fourment, P. Lasne. Toward an Automatic Preform Design Software for Axisymmetrical Forging. Dans : *Colloque IDMME98*, pp. 483-490, Compiègne, 1998.

[Vieilledent 98b] D. Vieilledent, L. Fourment, J.-L. Chenot. Non Linear Sensitivity Analysis in Forging: Application to Preform Tool Shape Optimization. Dans : *4th World Congress on Computational Mechanics*, (CDROM), Buenos Aires, Argentine, 1998.

[Vieilledent 99] D. Vieilledent, L. Fourment, J.-L. Chenot. Optimisation géométrique des outils en mise en forme. Dans : *14ième Congrés français de mécanique*, (à paraître), Toulouse, 1999.

[Weinberger 97] H. F. Weinberger. On the Nonexistence of Certain Ideal Forming Operations for Extrusion and Drawing Dies. *J. Mech. Phys. Solids*, 45(8), pp. 1275-1280, 1997.

[Wolfe 71] P. Wolfe. Convergence Condition for the Ascent Method. *SIAM Review*, 13, pp.185-188, 1971.

[Zabaras 93] N. Zabaras, S. Badrinarayanan. Inverse Problems and Techniques in Metal Forming Processes. *ASME 93*, Inverse Problems in Engineering: Theory and Practice, pp. 65-76, 1993

[Zabaras 95] N. Zabaras, S. Kang. Control of the freezing Interface Motion in Two-Dimensional Solidification Processses using the Adjoint Method. *Int. J. Num. Meth. Engng.,* 38, pp. 63-80, 1995.

[Zhao 95] G. Zhao, E. Wright, R. V. Grandhi. Forging Preform Design with Shape complexity Control in Simulation Backward Deformation. *Int. J. of Mach. Tools Manufacture,* 35(9), pp. 1225-1239, 1995.

[Zhao 97a] G. Zhao, E. Wright, R. V. Grandhi. Preform Die Shape Design in Metal Forming using an Optimization Method. *Int. J. Num. Meth. Engng*., 40, pp. 1213-1230, 1997.

[Zhao 97b] G. Zhao, E. Wright, R. V. Grandhi. Sensitivity analysis based preform shape design for net-shape forging. *Int. J. Mach. Tools Manufact.*, 37(9), pp. 1251-1271, 1997.

[Zolesio 79] J. P. Zolesio. *Identification de domaines par déformations.* Thèse de Doctorat, Université de Nice, 1979.

[Zoutendijk 60] G. Zoutendijk. *Methods of Feasible Directions*. Elsevier Publishing Company, Amsterdam, 1960.