

Développement d'un logiciel de simulation et d'optimisation d'un cycle de Brayton dans le cadre de la modélisation d'un turboréacteur double flux

Ir. V. KELNER
Ing. M.-A. PAULUS
GRAMME – Liège

Le présent travail traite de la modélisation 1D d'un turboréacteur (qui fonctionne selon le cycle thermodynamique de Brayton) ainsi que des différentes optimisations réalisées sur le cycle sur base des nouvelles méthodes par algorithme génétique. Cette optimisation multi critères a pour objectifs antagonistes de maximiser la poussée et de minimiser la consommation.

Mots-clés : turboréacteur, cycle de Brayton, modélisation, optimisation, algorithme génétique, simulation.

This paper is discussing the 1D modeling on a turbo fan (which works on the basis of the Brayton thermodynamic cycle) and the different optimizations computed on the cycle with genetic algorithm new methods. The multi criteria optimization has two antagonist objectives : minimizing consumption and maximizing thrust.

Keywords : turbofan, Brayton cycle, modeling, optimization, genetic algorithm, simulation.

1. Introduction

Le turboréacteur double flux est l'un des moteurs les plus utilisés dans l'aviation civile ; il est basé sur le cycle thermodynamique de Brayton. A titre d'exemple, le récent Airbus A380 est équipé de quatre turboréacteurs double flux dernière génération. Il est donc indispensable pour les concepteurs de ces engins de posséder un outil de simulation permettant de prédire les performances en fonction des données de conception. Ils font alors appel à cet outil qui est basé sur un modèle mathématique.

Le présent travail traite de la modélisation 1-D d'un turboréacteur. Il a été effectué dans les locaux du CRIG, (Centre de Recherches des Instituts Groupés), qui est le centre de recherche et développement du groupe HELMo. Les logiciels utilisés furent Matlab et sa *toolbox* Optintool pour la réalisation et l'optimisation du modèle et GasTurb pour les validations correspondantes.

Le principe de fonctionnement d'un turboréacteur est basé sur le cycle thermodynamique de Brayton pour systèmes ouverts et est composé de 4 phases : admission, compression, combustion et détente avec la compression et la détente s'effectuant sur le même arbre mécanique. L'air est mis en mouvement, aspiré à l'atmosphère, avant de subir une compression. S'ensuit une combustion qui va fortement augmenter l'enthalpie de l'air avant qu'il subisse une détente. Lors de la détente, l'air est accéléré et le moteur est propulsé en avant par réaction, d'où le terme réacteur.

Le réacteur choisi pour la réalisation du modèle est un réacteur double flux (et double corps) non mélangés (les flux froid et chaud ne se rencontrent plus avant l'extérieur) et à haut by-pass (rapport débit flux froid/débit flux chaud de l'ordre de 6). Ce type de réacteur est actuellement le plus répandu dans les avions civils. Ces réacteurs sont constitués d'un fan (qui peut être assimilé à un compresseur basse pression), d'un by-pass, d'un compresseur haute pression, d'une chambre de combustion, d'une turbine haute pression qui entraîne le compresseur haute pression, d'une turbine basse pression qui entraîne le fan, d'une tuyère pour le flux chaud et d'une tuyère pour le flux froid. Tous ces composants de même que les conduits les reliant sont caractérisés principalement par des valeurs de températures, pressions et débits en entrée et en sortie. D'autres caractéristiques comme les pertes de charge, les rendements, etc. peuvent aussi entrer en ligne de

compte. Chaque entrée ou sortie d'élément est caractérisée par un numéro de station (fig. 1).

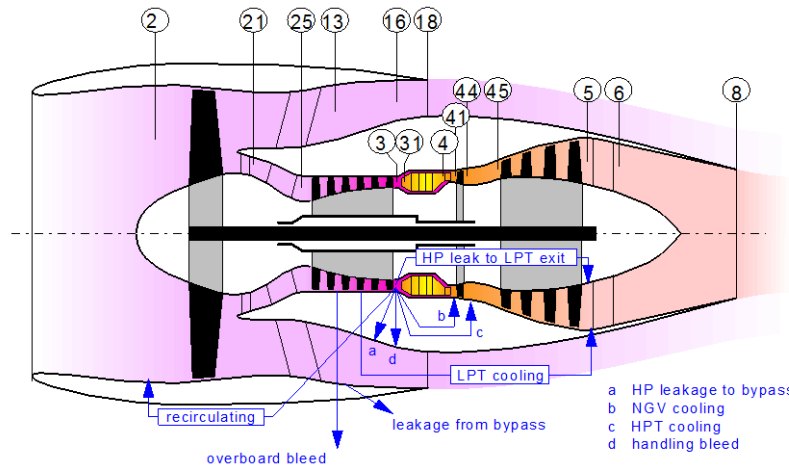


Figure 1 : schéma interne et stations d'un turboréacteur double flux

Les équations mathématiques qui régissent les échanges d'énergie dans le réacteur sont basées sur les bilans thermodynamique et massique de chaque élément.

Le bilan thermodynamique repose sur le premier principe de la thermodynamique pour un système ouvert et sur le principe de la conservation de l'énergie. Ce bilan va vérifier que la quantité d'énergie présente en entrée d'un composant additionnée au travail éventuellement fourni à ce composant est bien égale à la quantité d'énergie présente en sortie moins les pertes éventuelles et/ou le travail reçu par ce composant.

Le bilan massique va, quant à lui, vérifier que la masse totale en entrée du composant est égale à la masse totale en sortie, c'est-à-dire que la masse est bien conservée.

2. Modélisation

2.1 Principes de la modélisation

Le modèle va être constitué de « boîtes noires » (fig. 2) qui vont représenter chaque étape que le fluide parcourt dans le moteur et qui vont être calculées grâce au logiciel choisi. Une boîte noire s'articule comme suit : une série d'entrées va constituer les valeurs à attribuer aux inconnues des équations mathématiques internes de la boîte. Le logiciel va effectuer

les opérations demandées et calculer les valeurs des sorties. Ces valeurs de sorties constitueront soit des valeurs d'entrées de la boîte suivante (si nécessaire), soit des valeurs de fonctionnement du modèle. L'enchaînement des boîtes forme le modèle complet et les entrées du modèle assignées à la première boîte permettront d'obtenir les sorties attendues du modèle à la dernière boîte.

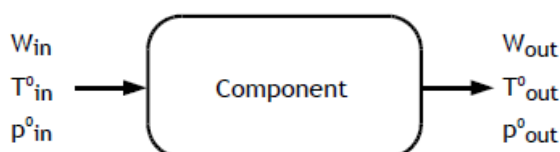


Figure 2 : schéma d'une boîte noire

2.2 Formulation mathématique

Chaque élément du réacteur (composant ou liaison entre deux composants) doit être modélisé mathématiquement et être lié à ceux qui l'entourent. L'enchaînement de tous les éléments liés donnera les résultats du modèle (voir point précédent).

On trouvera ci-après la liste des fonctions implémentées dans Matlab. Ces relations mettent en évidence les principes énoncés précédemment en matière de conservation d'énergie et de masse. Chaque fonction est écrite sous la forme $[sorties]=nom_fonction(entrées)$ et les indices chiffrés correspondent aux stations.

Fonction [P_{alt}, T_{alt}]=isa_info(alt)

Cette fonction permet de calculer les données ISA (*International Standard Atmosphere*) à l'altitude demandée.

Fonction [C_p, γ]=constantes(Temp)

Cette fonction permet de calculer le C_p et le γ pour l'air sec non brûlé en fonction de la température.

Fonction [T_{amb}, P_{amb}, T₁, P₁]=flux_libre(alt, Mn)

Cette fonction calcule les conditions du fluide à l'entrée du réacteur en fonction de l'altitude et du nombre de Mach (image de la vitesse). La température et la pression sont donc des valeurs dites « totales » en raison de la prise en compte de la vitesse.

Fonction [T₂, P₂, W₂, W₁]=conduit 1 2(T₁, P₁, PL₁₂, W_{12std})

L'entrée du réacteur est caractérisée par un espace de quelques dizaines de centimètres situé entre le bord d'attaque du réacteur et la face frontale du fan. Il y a donc des pertes de charge possibles. C'est à cet endroit que l'on définit le débit standard d'entrée qui va permettre de calculer le débit « réel » d'entrée.

Fonction

[W₁₃, W₂₁, W_{13std}, W_{21std}, T₂₁, T₁₃, P₂₁, P₁₃, PW_f, η_{isin}, η_{isout}]=fan(P₂, T₂, W₂, BPR : π₂₁, π₁₃, η_{polin}, η_{polout})

Le fan (ou compresseur basse pression) permet d'envoyer l'air dans les deux conduits qui constituent le réacteur. Le flux 21 se déplace dans le conduit central du réacteur et sera désigné plus tard flux chaud. Le flux 13 est le flux « by-passé » et sera appelé flux froid. Les deux flux ont des caractéristiques propres.

Fonction [T₁₈, P₁₈, W₁₈, W_{18std}]=BPR(T₁₃, P₁₃, PL₁₃, W₁₃)

Cette fonction calcule les conditions du flux froid entre la face arrière du fan et l'entrée de la tuyère correspondante où peuvent apparaître des pertes de charges.

Fonction [T₂₅, P₂₅, W₂₅, W_{25std}]=inter. f. c(T₂₁, P₂₁, PL₂₁₂₅, W₂₁)

Cette fonction calcule les conditions du flux chaud entre la face arrière du fan et l'entrée du compresseur haute pression. Elle est semblable à la précédente et donc prend en charge d'éventuelles pertes de charge.

Fonction

[W₃, T₃, P₃, PW_c, η_{isc}, LT₃, W_{3std}]=compresseur(W₂₅, P₂₅, T₂₅, η_{polc}, π_c, froid1, froid2)

La fonction du compresseur va utiliser les valeurs de rapport de compression et de rendement polytropique pour calculer la température et la pression en sortie. Le débit de sortie sera le débit d'entrée auquel on soustrait les débits de refroidissement. Enfin, la puissance requise par le compresseur est la somme des puissances nécessaires pour l'augmentation de pression d'une part et pour soutenir les flux de refroidissement d'autre part.

Fonction [T₃₁, P₃₁, W₃₁, W_{31std}]=sortie_comp(T₃, P₃, PL₃, W₂₅, W₃, L3A, L3B)

Cette fonction calcule les conditions du flux dans la partie qui suit le compresseur où des fuites (L3A et L3B) et des pertes de charge peuvent apparaître.

Fonction

[P₄,WF₄,T₄,W₄,W_{4Std},FAR]=chambre_comb(T₃₁,P₃₁,LHV_carb,PL₃₁₄,W₃₁,W₂₅,W₂,η_{comb},SOT,L3B,CW₄)

La fonction qui gère les conditions de la chambre de combustion est l'une des plus imposantes. D'une part, étant donné que les gaz brûlés ont un Cp différent, un nouveau calcul est nécessaire. D'autre part, la consommation de carburant est calculée sur base de la température demandée (SOT) et du rapport air-carburant dans les conditions stœchiométriques (pas d'excès ou de défaut dans les réactifs de la réaction de combustion) de la combustion. Enfin, la température et la pression de sortie sont calculées. Le débit total correspond au débit d'air en entrée auquel on additionne le débit de carburant calculé juste avant.

[T₄₄,P₄₄,W₄₄,W_{44Std}]=inter_thp_tlp(T₄₄,P₄₄,PL₄₄,W₄₄)

Cette fonction calcule les conditions du fluide dans le conduit qui sépare les deux turbines. Elle est similaire aux fonctions des conduits qui se situent avant la combustion.

Fonction

[T₄₄,P₄₄,PW_{HP},η_{isHP},expr_{HP},W₄₄,W_{44Std}]=turb_HP(T₄,P₄,PW_c,η_{meca},W₄,η_p,o_{HP},FAR)

La turbine HP étant l'organe qui va fournir son énergie au compresseur HP, la puissance qu'elle va extraire du fluide est directement calculée en fonction de celle demandée par le compresseur, au rendement mécanique près. Cette puissance extraite va dès lors définir la perte en température et en pression dans le fluide, au rendement de la turbine près.

Fonction

[T₅,P₅,PW_{LP},η_{isLP},expr_{LP},W₅,W_{5Std},R₅]=turb_LP(T₄₅,P₄₅,PW_f,W₄₅,η_{meca},η_{polLP},W₂₅,CW₄,froid2,L,T₃,T₃,FAR)

Le calcul des conditions de la turbine BP est pratiquement identique à celui de la turbine HP, en tenant compte des flux de refroidissement qui sont présents.

Fonction

[P₉,π₈₉,T₉,V₉,AN_{eff9},AN_{geom9},FG₉]=tuyere_chaud(P₅,P_{amb},T₅,W₅,CD₈,R₅)

Les tuyères étant considérées « étranglées » (*choked*), la valeur du nombre de Mach en leur centre vaut 1 (d'où M=1). Les valeurs val_Q_x et Q_x (où x vaut f ou c selon que l'on considère le flux froid ou le flux chaud) sont obtenues grâce à une fonction extérieure qui donne les valeurs de la courbe

Q^1 associée. Le rapport de détente de la tuyère est fonction de la pression d'entrée et de la pression de sortie, c'est-à-dire la pression atmosphérique. La section nette géométrique de la tuyère dépend du coefficient de décharge, valeur qui caractérise les pertes par frottement à la sortie de la tuyère. Enfin, la poussée brute est calculée en fonction de la vitesse d'éjection des gaz d'une part et de la différence de pression entre l'étranglement de la tuyère et l'atmosphère d'autre part.

Fonction

$[P_{19}, \pi_{1819}, T_{19}, V_{19}, AN_{eff19}, AN_{geom19}, FG_{19}] = \text{tuyere_froid}(P_{18}, P_{amb}, T_{18}, W_{18}, C_{D18})$

Le calcul est identique à celui de la tuyère du flux chaud.

$[FN, SFN, SFC] = \text{sortie_react}(T_{amb}, W_2, FG_9, FG_{19}, CX, WF_4, Mn)$

Cette dernière fonction calcule les sorties finales du réacteur qui sont la poussée nette, la poussée spécifique et la consommation spécifique. Après avoir calculé la poussée brute totale d'une part (au coefficient de poussée près) et la traînée d'autre part, la poussée nette est calculée par différence des deux premières. Les valeurs spécifiques sont basées sur le débit de carburant.

2.3 Validation

Les résultats donnés par le modèle ont été validés grâce à GasTurb. Les valeurs d'entrée étaient identiques afin de permettre une meilleure validation. Les sorties données par le modèle ont été comparées aux sorties données par GasTurb. Les erreurs relatives sur chaque valeur ont été calculées et ont donné les résultats suivants. Erreur moyenne sur

- Le débit : 0,01 %
- La température : 0,17 %
- La pression : 2,13 %
- Le débit standard : 1,84 %
- Les rendements : 0,04 %
- Les rapports de pression : 4,62 %
- Les sorties principales : 2,13 %

L'analyse de ces erreurs révèle que les divergences les plus importantes se situent au niveau des rapports de pression, et plus

¹ La courbe Q est une courbe caractéristique d'un paramètre quelconque.

particulièrement au niveau des détentes dans les turbines. Après vérification des résultats intermédiaires, il apparaît que les puissances du fan et du compresseur sont sensiblement différentes entre le modèle et GasTurb, ce qui induit une différence semblable au niveau des puissances des turbines. Le modèle donne les valeurs de puissances suivantes : $PW_{fan} = 1127,5 \text{ kW}$ et $PW_{compresseur} = 831 \text{ kW}$. GasTurb donne les puissances suivantes : $PW_{fan} = 1141,85 \text{ kW}$ et $PW_{compresseur} = 876 \text{ kW}$. Ce qui donne comme erreurs relatives : $\Delta PW_{fan} = 1,26\%$ et $\Delta PW_{compresseur} = 5,14\%$. Comme le calcul de la température de sortie des turbines dépend de la puissance qu'elles doivent fournir et que le calcul de la pression de sortie dépend du rapport des températures d'entrée et de sortie, l'erreur du calcul de puissance au niveau compression se répercute sur la détente. De plus, les turbines se suivant, l'erreur ne fait qu'augmenter au fur et à mesure du trajet du fluide.

Une différence sur la valeur du C_p doit probablement provoquer cette erreur sur la puissance au niveau compression. En effet, la puissance est donnée par $PW = C_p \cdot W \cdot \Delta T$. Les valeurs de températures et de débit étant très proches entre le modèle et GasTurb (de l'ordre de 0,3%), le seul élément qui peut perturber le calcul est le C_p . Cette différence s'explique par le fait que le calcul du C_p dans le modèle se fait via une formule analytique², tandis que le calcul dans GasTurb se fait via une formule plus complète et se base sur la chimie du fluide. Cette approche a été envisagée mais était trop conséquente pour être assimilée et incluse dans le modèle.

Pour le reste, on constate que les erreurs moyennes sont bien inférieures à 5%³, ce qui montre que les résultats sont acceptables. Le modèle peut donc être considéré comme validé.

3. Optimisation

3.1 Généralités

Dans tout problème d'optimisation, il faut trouver un maximum ou un minimum. Dans les applications simples, où un seul objectif est recherché, il « suffit » de formuler mathématiquement le problème et d'en chercher la dérivée. Celle-ci trouvée, on peut trouver les maxima ou minima de la

² Cette formule influence aussi l'exposant polytropique et la formule analogue pour les gaz chauds aussi.

³ La valeur seuil pour les erreurs relatives est de 5% en moyenne.

fonction première. Néanmoins, certaines fonctions comportent un (ou plusieurs) optimum (optima) local (locaux). La méthode de la dérivée montre alors ses limites car elle va se cantonner au premier optimum qu'elle va trouver sans savoir s'il est global ou seulement local. C'est dans ces cas, bien plus nombreux en réalité, que les algorithmes génétiques ou les algorithmes qui se basent sur des fonctionnements naturels⁴ amènent des solutions concrètes. Contrairement aux fonctions purement mathématiques, les algorithmes génétiques sont conçus de manière à explorer l'ensemble des solutions possibles et ainsi garder uniquement les solutions adéquates.

Dans le cadre du présent travail, la volonté de trouver la consommation spécifique (SFC) la plus faible tout en gardant une poussée nette (FN) suffisante présente deux objectifs contradictoires lors d'une optimisation. De fait, si on diminue la consommation, la poussée diminue et vice-versa. Nous sommes donc en présence d'un problème multi objectifs pour lequel il existe deux manières de procéder. La première consiste en l'utilisation d'un algorithme conçu pour le multi objectifs et qui va pouvoir optimiser en tenant compte des différents objectifs. La deuxième manière consiste à ramener le problème multi objectifs à un problème mono objectif en attribuant des « poids » aux différents objectifs et en optimisant une fonction objectif à plusieurs reprises lors des variations des poids. Les deux méthodes précitées ont été utilisées dans ce travail.

3.2 Algorithmes génétiques

Les algorithmes génétiques (A.G.) pour l'optimisation numérique font partie des algorithmes métaheuristiques^{5,6} les plus appréciés en raison de la qualité des résultats qu'ils fournissent. Ils sont basés sur la théorie de l'évolution qui avance que les individus d'une population s'adaptent à leur cadre de vie et évoluent. Certains se reproduisent, d'autres subissent des modifications génétiques, d'autres encore disparaissent. C'est en prenant

⁴ De la famille des métaheuristiques.

⁵ Métaheuristiques : *famille d'algorithmes d'optimisation visant à résoudre des classes générales de problèmes mathématiques en combinant des procédures de recherche pour trouver rapidement une bonne approximation de la meilleure solution.* (Source : <http://fr.wiktionary.org/wiki/m%C3%A9taheuristique>)

⁶ D'autres algorithmes métaheuristiques connus sont le recuit simulé, l'essaim de particules ou les colonies de fourmis.

exemple sur la nature que ces algorithmes ont vu le jour à partir des années 60. La figure 3 représente le fonctionnement de ces algorithmes.

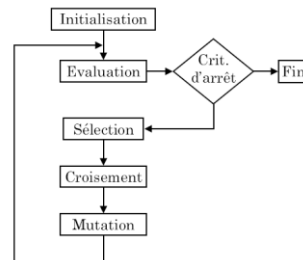


Figure 3 : schéma de fonctionnement d'un algorithme génétique

3.3 Résultats

Trois séries d'optimisations ont été effectuées grâce à l'outil dédié de Matlab (Optimtool) et on trouvera les résultats qui concernent l'optimisation au moyen de six variables (les deux autres étant à quatre et cinq variables). Ces six variables sont les plus facilement ajustables, que ce soit à la conception du réacteur (pour les cinq premières) ou lors du vol lui-même (pour la dernière).

Variables : BPR, Pifo, Pifi, Pic, SOT, Mn

Contraintes⁷ : [2 1.5 1.5 4 1200 0.5] à [8 1.8 2.5 14 1800 0.9]

Objectifs : minimiser SFC, maximiser FN (ou minimiser -FN)

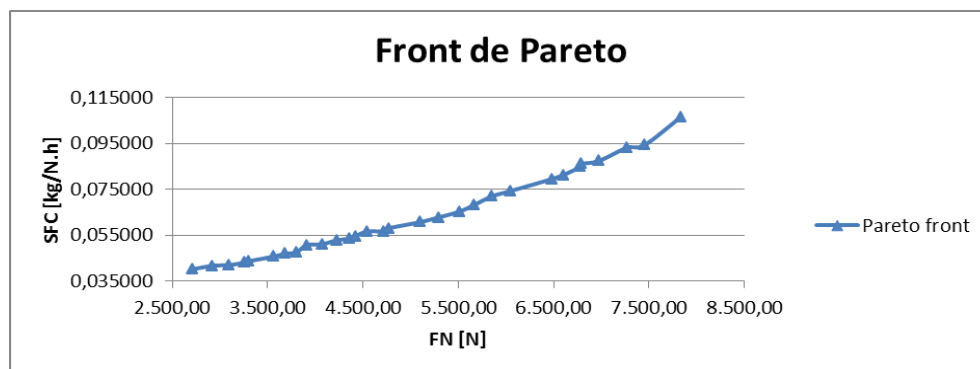


Figure 4 : front de Pareto obtenu par Optimtool pour 6 variables

⁷ Sous la forme [BPR_{min} Pifo_{min} Pifi_{min} Pic_{min} SOT_{min} Mn_{min}] à [BPR_{max} Pifo_{max} Pifi_{max} Pic_{max} SOT_{max} Mn_{max}]

L'étendue des résultats permet de constater que l'A.G. parcourt toutes les solutions possibles et propose un ensemble de solutions sous la forme du front de Pareto. Ce front représente donc les meilleures solutions que l'algorithme a trouvées.

3.4 Validation de l'optimisation

Tout comme pour le modèle, l'optimisation doit être validée. Dans ce cas-ci, deux phases composent la validation. Une première phase a d'abord été de vérifier l'existence des points trouvés par Optimtool. Cette phase a été motivée par le fait que GasTurb ne possède pas d'outil d'optimisation par algorithme génétique et que dans une première approche, l'outil d'optimisation inclus dans GasTurb ne donnait pas de résultats concluants en raison de la méconnaissance de l'algorithme. La deuxième phase a consisté en une véritable validation car l'algorithme de GasTurb était mieux maîtrisé.

3.4.1 Vérification

Afin de vérifier les points trouvés par Optimtool, les paramètres modifiés lors de chaque itération (à savoir BPR, Pifo, Pifi, Pic, SOT et Mn) ont été entrés dans GasTurb. Chaque point a donc été calculé et GasTurb en a donné l'existence ou non.

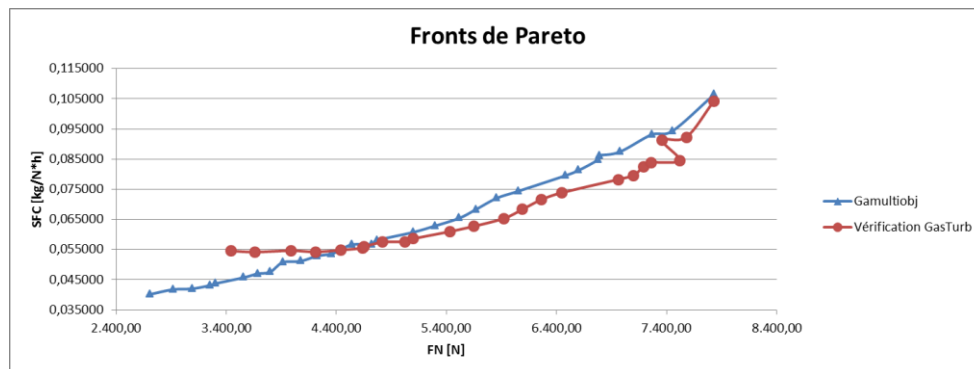


Figure 5 : fronts de Pareto vérification 6 variables

Dans ce cas d'optimisation, huit points sont exclus par GasTurb. Néanmoins, excepté un point « farfelu » à l'itération 30, les fronts se rapprochent et les erreurs entre le modèle et GasTurb sont faibles (fig. 5). A noter que l'erreur moyenne sur FN est de 5,25% et celle sur SFC est de 3,05%.

La vérification des points permet de mettre l'accent sur deux aspects en particulier. Premièrement, l'algorithme génétique explore tous les points de l'espace de conception, ce qui est positif mais peut aussi amener à des points non existants. C'est alors à l'utilisateur qu'il revient de faire le tri, que ce soit avec un autre logiciel ou à la main. Deuxièmement, le modèle montre ses limites lorsqu'il s'agit du calcul de points à basses poussée et consommation. En effet, tous les points situés à gauche sur les graphes sont les plus erronés par rapport à GasTurb. Cela est probablement dû à la présence de conditions particulières sur les paramètres qui ne sont pas incluses dans le modèle car elles ne sont pas connues.

3.4.2 Validation

Une vérification n'étant pas une validation, une connaissance approfondie de GasTurb s'est avérée nécessaire afin de pouvoir effectuer une vraie validation. En effet, l'outil d'optimisation de GasTurb n'est pas multi objectifs, ce qui posait problème lors des premiers essais. Néanmoins, le moyen de valider les résultats obtenus par Optimtool fut trouvé.

L'outil de GasTurb permet de minimiser ou maximiser un seul et unique objectif (appelé *Figure of Merit*, FoM) en fonction de sept variables au maximum et soumis à sept contraintes au maximum. L'objectif peut être soit une sortie prédéfinie soit une sortie personnalisée. C'est la deuxième solution qui a été retenue. En effet, afin de réaliser une optimisation bi objectifs avec un algorithme mono objectif, il faut combiner les deux objectifs en une seule équation (qui sera la FoM) et attribuer un poids à l'un et l'autre. Il ne restera plus qu'à minimiser (ou maximiser) la FoM pour les différentes valeurs des poids. La FoM à minimiser s'écrira donc sous la forme :

$$FoM = a * (-FN) + b * SFC$$

avec les poids

$$a \leq 1, b \leq 1, a + b = 1$$

car pour maximiser FN, il faut minimiser -FN.

Les résultats sont exposés ci-après.

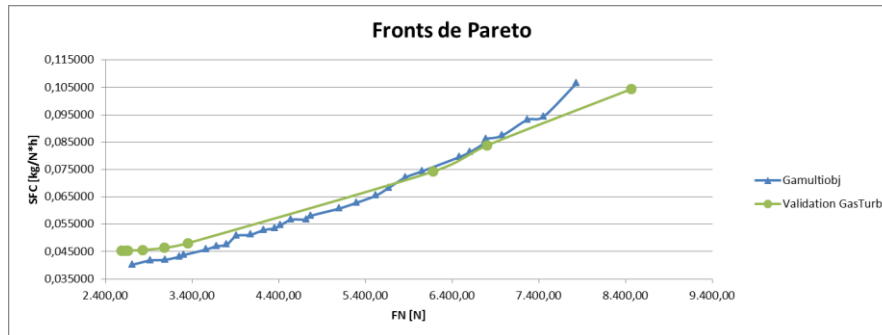


Figure 6 : comparaison des fronts de Pareto pour la validation à 6 variables

Les résultats (fig. 6) montrent que les tendances des fronts donnés par les deux algorithmes sont similaires et certains points sont même pratiquement identiques. Pour une poussée $\approx 3080\text{N}$, le gain en consommation de l'A.G. est de l'ordre de 10% par rapport à l'algorithme de GasTurb.

3.5 Comparaison des algorithmes d'optimisation

L'utilisation de deux algorithmes d'optimisation différents dans ce travail est très intéressante. En effet, cela permet de constater l'efficacité d'un algorithme génétique par rapport à un algorithme mathématique lors d'une optimisation multi objectifs, mais aussi ses faiblesses.

D'une part, on remarque que l'A.G. donne son front de Pareto en une seule fois. En effet, là où le front est calculé point par point avec GasTurb, le front donné par Matlab est calculé d'une traite. Un point donné par GasTurb a été calculé par un *run* et on a donc effectué 11 *runs* afin d'obtenir le front des solutions. Par contre, l'A.G. de Matlab, lui, a calculé l'ensemble des points du front et les a donnés après un seul *run*. Il est donc aisé de constater que, bien qu'un *run* de GasTurb soit plus rapide qu'un *run* de l'A.G., le temps total pour esquisser le front de Pareto est bien moindre pour l'A.G. A plus forte raison, lorsqu'on doit effectuer plus de *runs* dans GasTurb afin d'affiner le front, l'A.G. n'en demandera toujours qu'un seul.

Ensuite, en observant le graphe de comparaison des algorithmes (fig. 9, pour 4 variables), on constate que le front donné par l'A.G. est constitué de solutions réparties plus ou moins uniformément sur la courbe. L'algorithme de GasTurb, en revanche, ne couvre pas entièrement le front et seuls quelques points aux extrémités sont visibles. Le front doit alors être interpolé, ce qui peut amener des solutions non conformes. Une autre façon

de procéder afin d'améliorer le front donné par GasTurb est de raffiner les valeurs des poids en fonction du dernier point obtenu, ce qui permettra d'obtenir plus de solutions, mais nécessitera plus de temps de calcul.

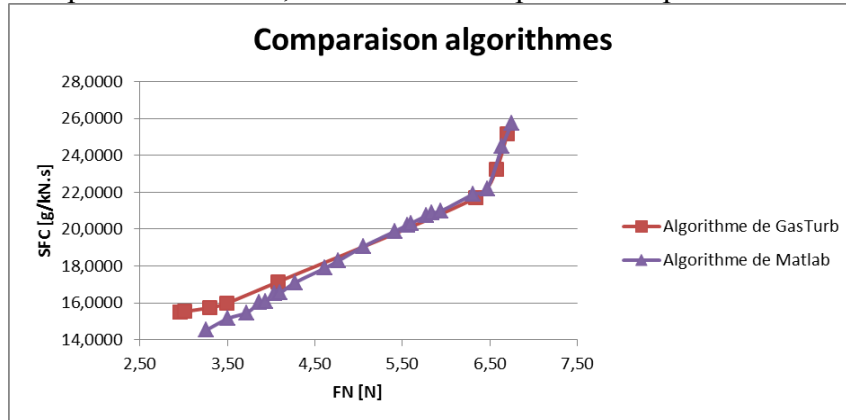


Figure 7 : comparaison des algorithmes

Enfin, d'après le graphe (fig. 7), les points situés sur la gauche du front sont de meilleure qualité lorsqu'il s'agit de ceux donnés par l'A.G. Ceci peut s'expliquer par le fait que l'A.G. est programmé pour explorer tout l'espace de conception et que, dès lors, certaines solutions extrêmes peuvent être trouvées, au contraire d'un algorithme mathématique standard qui pourrait être bloqué par certaines situations (optimum local par exemple).

Ces considérations étant apportées, il semble logique d'adopter l'A.G. comme algorithme d'optimisation par défaut dans tout problème d'optimisation. Néanmoins, certains inconvénients restent rédhibitoires. Un A.G. est assez lourd à mettre en place et comme il fait appel à un calculateur indépendant, il n'est pas toujours facile de gérer l'interfaçage. De plus, les temps de calculs peuvent se révéler assez long (de l'ordre de plusieurs jours pour des calculs poussés de mécanique des fluides). Sans oublier que la paramétrisation de l'A.G. est souvent fastidieuse tant le nombre de paramètres est important (taille de population, critères de mutation, de tournoi,...). Il faut donc analyser les données d'un problème avant de choisir un A.G. pour l'optimisation.

4. Conclusions

4.1 Modèle – Résultats – Erreurs

Le but du travail décrit ci-avant résidait dans la conception d'un modèle 1-D de turboréacteur et sa validation. Au vu des pages qui précèdent, nous pouvons confirmer que ce but est atteint, voire dépassé. Le modèle conçu durant ce travail est un modèle validé qui fournit des résultats très semblables à ceux fournis par GasTurb, le logiciel référence dans ce domaine. En effet, des erreurs inférieures à 10% dans tous les cas et proches de 1% pour beaucoup de points sont considérées comme tout à fait acceptables pour une première approche de modélisation et mènent à la conclusion que le modèle est fiable.

Néanmoins, plusieurs améliorations sont encore possibles. D'une part, une meilleure connaissance du comportement du fluide dans ses conditions de froid ou de chaleur permettrait de mieux définir la valeur du C_p et ainsi diminuer les erreurs qui y sont dues. D'autre part, certains points de fonctionnement du modèle ne peuvent exister en réalité et connaître les conditions limites tant du fluide que des composants permettrait de mieux maîtriser les résultats. Une autre amélioration pourrait également être d'inclure la prise en charge de composantes géométrique et/ou conceptuelle dans le modèle afin de se rapprocher encore de la réalité. Enfin, des options comme le choix du carburant, des flux mixés ou non, etc. pourraient étoffer les possibilités du logiciel.

4.2 Optimisation

Un des aspects les plus intéressants de ce travail concerne l'optimisation multi objectifs par algorithme génétique (A.G.). En effet, les outils actuellement utilisés pour l'optimisation sont pour la plupart basés sur des algorithmes mathématiques. Pourtant, les résultats obtenus dans ce travail montrent que l'efficacité des A.G. les rend bien plus intéressants. Il suffit de compter le temps pour effectuer tous les calculs de création d'un front de Pareto pour se rendre compte que les A.G. sont plus rapides et plus précis que leurs homologues mathématiques. Là où un A.G. bien réglé donne le front en un seul *run*, le mathématique nécessite souvent de jouer avec des coefficients (appelés aussi poids) et d'effectuer plusieurs *runs* afin de constituer une partie plus ou moins complète du front.

Une amélioration réalisable de ce travail serait le développement d'un algorithme génétique propre au modèle afin de procéder à l'optimisation du turboréacteur indépendamment de Matlab.

5. Sources

- métaheuristique*. (2012, 04 21). Consulté le 06 07, 2012, sur Wiktionnaire:
<http://fr.wiktionary.org/wiki/m%C3%A9taheuristique>
- Algorithme génétique*. (s.d.). Consulté le 06 05, 2012, sur Wikipedia:
http://fr.wikipedia.org/wiki/Algorithme_g%C3%A9n%C3%A9tique
- Borguet, S., Kelner, V., & Léonard, O. (2006). *Cycle Optimization of a Turbine Engine : an Approach Based on Genetic Algorithms*. Liège: University of Liège.
- Cavcar, M. (s.d.). *The International Standard Atmosphere (ISA)*. Eskisehir, Turkey: Anadolu University.
- Cycle de Brayton*. (s.d.). Consulté le 06 04, 2012, sur Wikipedia:
http://fr.wikipedia.org/wiki/Cycle_de_Brayton
- Kelner, V. (2011). *Cours de turbomachines*. Liège.
- Kelner, V. (2011). *Introduction à l'optimisation*. Liège.
- Léonard, O. (2006). *Notes de propulsion aérospatiale*. Liège: Université de Liège.
- Les algorithmes génétiques*. (s.d.). Consulté le 06 05, 2012, sur Developpez.com:
<http://khayyam.developpez.com/articles/algo/genetic/>
- Martin, S., Wallace, I., & Bates, D. G. (2008). *Development and Validation of an Aero-engine Simulation Model for Advanced Controller Design*. Seattle: American Control Conference.
- Soares, C. (2007). *Gas Turbines*. Dallas: Butterworth-Heinemann.
- Turboréacteur*. (s.d.). Consulté le 06 01, 2012, sur Wikipédia:
<http://fr.wikipedia.org/wiki/Turbor%C3%A9acteur>
- Turboréacteurs (Généralités)*. (s.d.). Consulté le 06 04, 2012, sur Aéronautique:
http://www.aeronautique.ma/Turboreacteurs-Generalites_a426.html
- Walsh, P. P., & Fletcher, P. (2004). *Gas Turbine Performance (Second Edition)*. Oxford: Blackwell Science Ltd.