



INSTITUTO TECNOLÓGICO DE AERONÁUTICA (ITA)
PRGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS E
TECNOLOGIAS ESPACIAIS

WRITTEN AND CODED BY ALISSON VINICIUS BRITO LOPES

Report:
PROJECT No. 2

SÃO JOSÉ DOS CAMPOS-SP

21 DE MAIO DE 2015

1 INTRODUÇÃO

As malhas podem ser classificadas em três tipos: 1) Malhas cartesianas (se referindo a malhas que não se conformam a superfície), 2) Malhas estruturas e 3) Malhas não estruturadas. Quanto ao processo de geração de malhas ULLER e AZEVEDO (1991) mostram três classificações, a saber: algébricos, analíticos e diferenciais. O que se percebe na literatura é que essa classificação não é muito bem definida e cada autor apresenta sua forma própria de classificação dos processos de geração.

A geração de malhas estruturadas (malhas que se conformam a configuração é um assunto bastante tratado na literatura) como se pode ver no livro e nas diversas publicações de Thompson (1977). De uma forma geral a grande diferença entre uma malha estruturada e uma malha não estrutura se diz no fato que nas malhas estruturadas a relação é intrínseca, ou seja, tenho/conheço todos os pontos $(i, j, j+1, i+1, i-1, j-1)$, contudo, o grande problema é que não é trivial gerar malhas sobre configurações complexas. Neste contexto a experiência deste trabalho mostra que também é bastante difícil fazer o ajuste de parâmetros/funções para atrações de linhas e pontos.

As malhas onde recai sobre o usuário a responsabilidade de explicitar a vizinhança (por meio, por exemplo, de uma consulta a uma tabela para saber quem são os vizinhos) são conhecidas como malhas não estruturadas.

1.1 Geradores Parabólicos

Os geradores de malhas parabólicas, como aquele desenvolvido por NOACK e ANDERSON (1988) e também apresentado no *paper* de ULLER e AZEVEDO (1991) (os mesmos argumentam também que este tipo de gerador não é muito comum na literatura) é desenvolvido por meio de um processo de marcha no espaço, saindo da fronteira interior, ou seja, do perfil até a fronteira exterior, e permite que se especifiquem os pontos da malha em ambas as fronteiras. A vantagem da utilização deste tipo de gerador é que o mesmo garante ortogonalidade da malha na superfície do aerofólio, além disso, esse tipo de malha poderia ser utilizado também para solução dos escoamentos.

O método para a geração de uma malha parabólica inicia-se basicamente considerando a geração de uma malha de referência local que se trata de uma malha hiperbólica/algébrica, o que se busca com o gerador parabólico é uma suavização da malha de referência local. A equação para geração dessa malha parte do seguinte conjunto de equações (1), onde o parâmetro r indica as coordenadas x ou y :

$$\begin{aligned} A(x_{\xi\xi} + \phi x_{\xi}) - 2Bx_{\xi\eta} + C(x_{\eta\eta} + \psi x_{\eta}) &= 0 \\ A(y_{\xi\xi} + \phi y_{\xi}) - 2By_{\xi\eta} + C(y_{\eta\eta} + \psi y_{\eta}) &= 0 \end{aligned} \tag{1}$$

As funções de controle do grid ϕ e ψ são utilizadas para obter um certo agrupamento de pontos no grid, mas no presente trabalho estas funções serão inicializadas com valor zero. Os coeficientes A, B e C são representados conforme a eq. (2)

$$A = x_\eta^2 + y_\eta^2 \quad B = x_\xi x_\eta + y_\xi y_\eta \quad C = x_\xi^2 + y_\xi^2 \quad (2)$$

Aproximando todas as derivadas da equação (2) por operadores centrados de segunda ordem nos leva a um algoritmo para geração de grid elíptico, que pode ser resolvido em um procedimento semelhante aquele adotado para a solução da equação de Laplace conforme a equação (3)

$$\begin{aligned} 2Ar_{i+1,j} - 4(A + C)r_{i,j} + 2Ar_{i-1,j} \\ = B(r_{i+1,j+1} - r_{i-1,j+1} - r_{i+1,j-1} + r_{i-1,j-1}) - 2C(r_{i,j+1} \\ + r_{i,j-1}) \end{aligned} \quad (3)$$

O índice “i” corresponde à direção ξ e “j” corresponde a direção η . Essa equação leva de fato a um sistema tridiagonal periódico para cada linha $j = \text{constante}$, necessitando assim conhecer os valores de r para a linha $j + 1$. Uma vez que o problema que surgiu na implementação esta no fato do não conhecimento do valor de r para esta posição, isto acaba levando ao conceito de malha de referência local, conforme apresentado por NOACK e ANDERSON (1988).

1.1.1 Geração da malha de referência local

A geração da malha de referência local aplicada neste trabalho segue a metodologia desenvolvida por NOACK e ANDERSON (1988). Primeiramente calcula-se o jacobiano da transformada (o jacobiano é proporcional à área da célula) eq. (4) e impondo que a malha de referência local é ortogonal, ou seja, $B = 0$, calcula-se a eq. (5)

$$x_\xi y_\eta - x_\eta y_\xi = J^{-1} \quad (4)$$

$$x_\xi x_\eta + y_\xi y_\eta = B \quad (5)$$

Resolver para x_η e y_η , utilizando a direção η como sendo a direção de marcha é possível obter as seguintes equações (6) e (7) hiperbólicas explícitas para a geração do grid :

$$x_\eta = \frac{-y_\xi S_\eta}{\sqrt{x_\xi^2 + y_\xi^2}} \quad (6)$$

$$y_\eta = \frac{x_\xi S_\eta}{\sqrt{x_\xi^2 + y_\xi^2}} \quad (7)$$

Onde S é um comprimento de arco ao longo de uma linha $\xi = \text{constante}$, e a derivada do comprimento de arco na direção de marcha é calculada conforme a eq. (8). R é a distância entre o ponto $(i, j - 1)$ e a corresponde fronteira externa $(i, JMAX)$, $\Delta\bar{s}$ e s são calculados respectivamente, pelas seguintes equações (10) e (11) :

$$S_\eta = \Delta\bar{s}R \quad (8)$$

$$R = \sqrt{(x_{i,JMAX} - x_{i,j-1})^2 + (y_{i,JMAX} - y_{i,j-1})^2} \quad (9)$$

$$\Delta\bar{s} = \frac{\bar{s}_j - \bar{s}_{j-1}}{\bar{s}_{jmax} - \bar{s}_{j-1}} \quad (10)$$

Onde s é determinado por meio da seguinte expressão:

$$s_{j+1} = s_{i,j} + YSF(s_{i,j} - s_{i,j-1}) \quad (11)$$

YSF é parâmetro que controla o estiramento da malha na direção η . Um valor adequado para este parâmetro deve estar no seguinte intervalo: $1.0 \leq YSF \leq 1.25$. O espaçamento inicial (dy) entre $s(0)$ e $s(1)$ pode ser determinado por meio de uma progressão geométrica, conforme feito no código.

O próximo passo é gerar a malha ortogonal por meio das equações (12) e (13), lembrando que essa malha nunca chega até a fronteira (posição $j = j_{jmax}$) :

$$x^o_{i,j} = x_{i,j-1} + (x_\eta)_{i,j-1} \quad (12)$$

$$y^o_{i,j} = y_{i,j-1} + (y_\eta)_{i,j-1} \quad (13)$$

As derivadas y_ξ e x_ξ são obtidas pela aplicação de operadores centrados (eqs. 14 e 15), sendo que no domínio computacional $\Delta\xi = \Delta\eta = 1$:

$$x_\xi = \frac{1}{2} (x_{i+1,j-1} - x_{i-1,j-1}) \quad (14)$$

$$y_\xi = \frac{1}{2} (y_{i+1,j-1} - y_{i-1,j-1}) \quad (15)$$

A parcela algébrica da malha de referência local é calculada conforme as equações (16) e (17):

$$x^{int}_{i,j} = x_{i,j-1} + \Delta\bar{s}(x_{i,jmax} - x_{i,j-1}) \quad (16)$$

$$y^{int}_{i,j} = y_{i,j-1} + \Delta\bar{s}(y_{i,jmax} - y_{i,j-1}) \quad (17)$$

Uma vez estabelecido os dois procedimentos para a geração da malha de referência local, os valores das coordenadas podem ser calculados por meio de uma operação de interpolação, sendo que quanto mais próximo da superfície do perfil, procurar-se-á privilegiar a ortogonalidade e quanto mais distante do perfil, maior será a influência da parcela algébrica, essa ponderação é feita por uma função do tipo “switching” (eq. (18)) sendo que as coordenadas finais são definidas pelas eqs. (19) e (20):

$$\epsilon = \frac{j-1}{J_{MAX}-1} \quad (18)$$

$$x_{i,j} = \epsilon x_{i,j}^{int} + (1 - \epsilon) x_{i,j}^o \quad (19)$$

$$y_{i,j} = \epsilon y_{i,j}^{int} + (1 - \epsilon) y_{i,j}^o \quad (20)$$

Deve-se perceber, contudo que no *paper* do NOACK e ANDERSON os mesmo argumentam que os valores de ϵ devem variar de zero na superfície do aerofólio até o valor de um no *shock*, no presente trabalho ϵ assume valor um na fronteira externa.

ULLER e AZEVEDO (1991) observaram que as derivadas dos coeficientes A, B e C na geração da malha parabólica devem ser discretizadas para a direção ξ , conforme as equações (21) e (22) :

$$x_{\xi} = \frac{1}{2} (x_{i+1,j+1} - x_{i-1,j+1}) \quad (21)$$

$$y_{\xi} = \frac{1}{2} (y_{i+1,j+1} - y_{i-1,j+1}) \quad (22)$$

As derivadas na direção η para estes coeficientes devem ser calculadas conforme as seguintes eqs. (23) e (24):

$$x_{\eta} = \frac{1}{2} (x_{i,j+1} - x_{i,j}) \quad (23)$$

$$y_{\eta} = \frac{1}{2} (y_{i,j+1} - y_{i,j}) \quad (24)$$

Desta maneira estes coeficientes não lineares são sempre avaliadas considerando os dados no *grid* de referência.

1.2 Geradores Elípticos

Este projeto visa o desenvolvimento de um programa computacional, a ser desenvolvido em linguagem científica C++ para implementação de geradores de malhas computacionais (estruturadas), especificamente a geração de malha elíptica com topologia “O”

sobre um perfil biconvexo e do perfil aeronáutico NACA 0012. Para isso o programa deve ser capaz de solucionar a equação de Poisson (eq. 25). Geradores Elípticos são construídos admitindo-se que as coordenadas no espaço computacional (ξ, η) devem satisfazer, cada uma delas, uma equação de Poisson no domínio físico, onde ξ e η são as variáveis dependentes e x e y são as variáveis independentes :

$$\begin{aligned} Ax_{\xi\xi} - 2Bx_{\xi\eta} + Cx_{\eta\eta} + D(Px_{\xi} + Qx_{\eta}) &= 0 \\ Ay_{\xi\xi} - 2By_{\xi\eta} + Cy_{\eta\eta} + D(Py_{\xi} + Qy_{\eta}) &= 0 \end{aligned} \quad (25)$$

Para se obter a solução numérica do problema é necessário estabelecer uma aproximação de diferenças finitas para as equações (25). As derivadas presentes são substituídas por operadores de diferenças finitas centrados e de 2ª ordem de precisão. O operador de resíduo para este caso é definido conforme a equação (26)

$$L(\phi)_{i,j} = [A_{i,j}\delta_{\xi\xi} - 2B_{i,j}\delta_{\xi\eta} + C_{i,j}\delta_{\eta\eta}](\phi)_{i,j} \quad (26)$$

Uma vez que as variáveis na eq. 25 não são função do tempo, a mesma será resolvida utilizando técnicas conhecidas como métodos de relaxação, o que significa que: a solução da mesma será obtida por meio de um processo iterativo. A solução desta equação será considerada convergida, quando a solução se tornar invariante com a continuação das iterações. Neste contexto, o *resíduo* é a grandeza sobre a qual se efetuará os testes de convergência.

Utilizando-se operadores centrados, os termos de derivada segunda estão sendo aproximados pelos operadores:

$$\delta_{\xi\xi}(\phi)_{i,j} = (\phi)_{i+1,j} - 2(\phi)_{i,j} + (\phi)_{i-1,j} \quad (27)$$

$$\delta_{\xi\eta}(\phi)_{i,j} = \frac{1}{4}[(\phi)_{i+1,j+1} - (\phi)_{i+1,j-1} - (\phi)_{i-1,j+1} + (\phi)_{i-1,j-1}] \quad (28)$$

$$\delta_{\eta\eta}(\phi)_{i,j} = (\phi)_{i,j+1} - 2(\phi)_{i,j} + (\phi)_{i,j-1} \quad (29)$$

Além disso, todas as derivadas envolvidas nos coeficientes A, B, C e D, também devem ser discretizadas por operadores centrados:

$$\frac{\partial}{\partial \xi}(\phi)_{i,j} \cong \delta_{\xi}(\phi)_{i,j} = \frac{1}{2}[(\phi)_{i+1,j} - (\phi)_{i-1,j}] \quad (30)$$

$$\frac{\partial}{\partial \eta}(\phi)_{i,j} \cong \delta_{\eta}(\phi)_{i,j} = \frac{1}{2}[(\phi)_{i,j+1} - (\phi)_{i,j-1}] \quad (31)$$

Independente do esquema específico de relaxação a ser utilizado (embora todos os métodos propostos fosse utilizados e testados neste presente trabalho) o problema discreto deste projeto estará convenientemente escrito na forma padrão de correção (*forma delta*), equações (32) e (33):

$$\begin{aligned} N\Delta x_{i,j}^n + \omega Lx_{i,j}^n &= 0 \\ N\Delta y_{i,j}^n + \omega Ly_{i,j}^n &= 0 \end{aligned} \quad (32)$$

Onde,

$$\begin{aligned} \Delta x_{i,j}^n &= x_{i,j}^{n+1} - x_{i,j}^n \\ \Delta y_{i,j}^n &= y_{i,j}^{n+1} - y_{i,j}^n \end{aligned} \quad (33)$$

O termo $Lx_{i,j}^n$ ou $Ly_{i,j}^n$ é a discretização da equação diferencial (operador resíduo) e $\Delta x_{i,j}^n$ ou $\Delta y_{i,j}^n$ (eq. 33) é a correção a ser efetuada no calculo das posições x e y da malha no nível de iteração n .

O operador N representa os esquemas de iteração a serem utilizados, já escritos em forma padrão de correção. Os 3 esquemas de iteração a serem utilizados neste trabalho são : SLOR, AF1 e AF2. Todos estes esquemas serão testados no presente trabalho. Inicialmente programar-se-á o método AF1, conforme a equação (34):

$$N_{AF1}(\)_{i,j} = -\frac{1}{\alpha}(\alpha - A_{i,j}^n \delta_{\xi\xi})(\alpha - C_{i,j}^n \delta_{\eta\eta})(\)_{i,j}^n \quad (34)$$

O parâmetro α é chamado de parâmetro de aceleração de convergência usual dos métodos AF. O processo iterativo de solução através do esquema de relaxação proposto vai exigir a existência de uma malha inicial, e conforme solicitado na proposta do projeto, a malha inicial será a malha parabólica, e como já discutido, este gerador garante ortogonalidade da malha na superfície do perfil. Como se deseja obter uma malha com topologia "O" sobre o aerofólio, deve-se admitir condições de contorno periódicas na direção circunferencial.

Tanto o perfil biconvexo quanto o perfil NACA 0012 apresentam equações relativamente simples, conforme equações (35) e (36); A fronteira externa é um circulo de raio 6,5 vezes a corda do perfil.

$$y = 2tx(1 - x) \quad (35)$$

$$y = \pm \frac{t}{0.2} c \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \left(\frac{x}{c} \right) - 0.3516 \left(\frac{x}{c} \right)^2 + 0.2843 \left(\frac{x}{c} \right)^3 - 0.1036 \left(\frac{x}{c} \right)^4 \right] \quad (36)$$

2 DOCUMENTAÇÃO E APRESENTAÇÃO DE RESULTADOS

2.1 Casos a serem estudados

Quatro estudos de caso foram solicitados para esse projeto (os mesmos serão apresentados resumidamente): o caso 1 consiste em implementar o método AF1, fazer uma análise da influência dos parâmetros α e ω na razão de convergência, buscando determinar valores ótimos que maximizem a convergência, além disto sugere-se implementar a sequência de alfas. O caso 2 trata-se de verificar o efeito da implementação das funções P e Q , por meio da modificação do operador de resíduo, consultando para isso os *papers* do KRIEGER, e as diversas publicações do THOMPSON, neste contexto o livro do FLETHCER também poderá ser utilizado.

O caso 3 consiste na implementação do método AF2 e investigação da influência dos parâmetros α e ω na razão de convergência, e na comparação da eficiência deste método com o método AF1, neste caso será programado também um contador de tempo computacional para avaliação do custo computacional de cada método.

O caso 4 consiste na geração da malha elíptica, utilizando todo o procedimento anteriormente desenvolvido para gerar a malha com topologia "O" sobre um perfil NACA 0012, utilizando para isso o gerador parabólico inicialmente desenvolvido.

O objetivo geral deste trabalho consiste, na geração da malha parabólica, posteriormente a elíptica utilizando os métodos AF1 e SLOR. Deve-se, contudo escolher dois estudos de caso dentro os 4 acima apresentados. Entretanto, têm-se como objetivo específico "tentar" realizar os 4 estudos.

A visualização das malhas geradas, tanto do perfil biconvexo, quanto do perfil NACA 0012, serão feitas por meio de uma visão da malha como um todo (visão geral) e em termos de detalhes próximos ao bordo de ataque e ao bordo de fuga.

2.2 AF1 ou ADI (ou do inglês ADI, *Alternating Direction Implicit*)

A utilização do método AF1 é feita em dois passos (como apresentado por HOLST (1979) e implementado no trabalho de MORGENSTERN e AZEVEDO (1990)). No primeiro passo resolve-se uma matriz tridiagonal periódica e novamente no segundo passo se resolve outra matriz tridiagonal.

$$\begin{aligned} \frac{1}{\alpha} (\alpha - A_{i,j}^n \delta_{\xi\xi}) (\alpha - C_{i,j}^n \delta_{\eta\eta}) (\Delta r)_{i,j}^n \\ = \omega \left[A_{i,j}^n (r_{i,j}^n)_{\xi\xi} - 2B_{i,j}^n (r_{i,j}^n)_{\xi\eta} + C_{i,j}^n (r_{i,j}^n)_{\eta\eta} \right] \end{aligned} \quad (37)$$

Passo 1:

$$(\alpha - A_{i,j}^n \delta_{\xi\xi}) \Delta s_{i,j}^n = \alpha \omega L r_{i,j}^n \quad (38)$$

Passo 2:

$$(\alpha - C_{i,j}^n \delta_{\eta\eta}) \Delta r_{i,j}^n = \Delta s_{i,j}^n \quad (39)$$

$\Delta s_{i,j}^n$ é o resultado intermediário obtido por meio da solução de duas tridiagonais periódicas para cada linha $\eta = \text{constante}$ (ou seja, mantém-se “j” fixo e resolve tridiagonais em “i”). Os valores de x e y , para a iteração $n+1$, são obtidos no segundo passo a partir dos valores de $\Delta s_{i,j}^n$, resolvendo-se neste caso duas tridiagonais “simples” para linhas $\xi = \text{constante}$, uma vez que o sistema gerado será apenas tridiagonal dado que as condições de contorno em $j = 1$ e $j = JMAX$ são do tipo *Dirichlet*, neste caso o algoritmo implementado é aquele mesmo do Projeto 1.

2.2.1 Solução da Matrix tridiagonal Periódica

Existem na literatura vários algoritmos construídos para solução de matrizes tridiagonal periódica (e.g. aqueles apresentado nos livros do HIRSH ou do FLETCHER). Entretanto achou-se interessante trabalhar na adaptação do algoritmo descrito por meio de um pseudocódigo disponível no Apêndice B (página 142) da seguinte referência: AZEVEDO (1988). *Transonic Aeroelastic Analysis of Launch Vehicle Configurations*. Sendo que o processo de “inversão” da matriz tridiagonal periódica envolve o processo de decomposição LU, resolvido em dois passos: um *Forward Sweep* e um *Backward Sweep*.

A matriz tridiagonal periódica obtida no passo 1 tem o seguinte formato:

$$\begin{bmatrix} b_1 & c_2 & & & a_{i_{MAX}-1} \\ a_1 & b_2 & c_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{i_{MAX}-3} & b_{i_{MAX}-2} & c_{i_{MAX}-1} \\ c_1 & & & a_{i_{MAX}-2} & b_{i_{MAX}-1} \end{bmatrix} \begin{Bmatrix} \Delta s_{1,j}^{(n)} \\ \vdots \\ \Delta s_{i,j}^{(n)} \\ \vdots \\ \Delta s_{i_{MAX}-1,j}^{(n)} \end{Bmatrix} = \begin{Bmatrix} vt_1 \\ \vdots \\ vt_i \\ \vdots \\ vt_{i_{MAX}-1} \end{Bmatrix} \quad (40)$$

Onde:

$$a_{i-1,j} = -A_{i,j}^n \quad (41)$$

$$b_{i,j} = \alpha + 2A_{i,j}^n \quad (42)$$

$$c_{i+1,j} = -A_{i,j}^n \quad (43)$$

$$vt_{i,j} = \alpha \omega L r_{i,j}^n \quad (44)$$

A matriz tridiagonal simples é resolvida para cada valor de “ i ” fixo dentro de intervalo $1 \leq i \leq i_{MAX} - 1$, resolve-se tridiagonais em “ j ”:

$$\begin{bmatrix} b_1 & c_2 & & & \\ a_1 & b_2 & c_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{j_{MAX}-4} & b_{j_{MAX}-3} & c_{j_{MAX}-2} \\ & & & a_{j_{MAX}-3} & b_{j_{MAX}-2} \end{bmatrix} \begin{Bmatrix} \Delta r_{i,2}^{(n)} \\ \vdots \\ \Delta r_{i,j}^{(n)} \\ \vdots \\ \Delta r_{i,j_{MAX}-1}^{(n)} \end{Bmatrix} = \begin{Bmatrix} vt_1 \\ \vdots \\ vt_j \\ \vdots \\ vt_{j_{MAX}-2} \end{Bmatrix} \quad (45)$$

Onde:

$$a_{i,j-1} = -c_{i,j+1} \quad (46)$$

$$b_{i,j} = \alpha + 2 \cdot c_{i,j+1} \quad (47)$$

$$c_{i,j+1} = -c_{i,j+1} \quad (48)$$

$$vt_{i,j} = \Delta s_{i,j+1}^{(n)} \quad (49)$$

2.2.2 Considerações sobre sequência de alfas (α 's) aplicadas ao método AF1 e AF2

Alguns autores sugerem utilizar uma sequência de α 's em vez de usar um valor de α constante para todo o processo iterativo. Esta ideia é motivada e tem sua raiz no conceito de aniquilação de autovetores e no fato que para um dado valor de α o esquema trabalha bem em altas frequências e mal em baixas frequências, ou vice-versa, isso significa dizer que valores pequenos de α correspondem a valores grandes de Δt e neste caso estou tentando aniquilar λ pequenos, valores altos de α representa valores pequenos de Δt e estou tentando aniquilar as altas frequências.

Segundo BALLHAUS, JAMESON e ALBERT uma sequência adequada de α 's é expressada pela seguinte equação (50):

$$\alpha_k = \alpha_H \left(\frac{\alpha_L}{\alpha_H} \right)^{\frac{k-1}{M-1}} \quad (50)$$

As recomendações discutidas em sala de aula nos diz que uma boa sequência de α 's deve ter entre 5 e 10 α 's elementos (M). Para o esquema AF1 $\alpha_L = O(1)$ e $\alpha_H = O\left(\frac{4}{\Delta x^2}\right)$, enquanto para o método AF2 $\alpha_L = O(1)$ e $\alpha_H = O\left(\frac{1}{\Delta x}\right)$, onde Δx é o menor intervalo na malha considerando ambas as direções.

É de salientar que o contexto que essa sequência de alfas aparece é no contexto da solução da equação do potencial completo. Iremos adaptar a escolha dos parâmetros α para o contexto deste trabalho.

2.3 SLOR (*Sucessive Line over Relaxation*)

O operador N para o método SLOR é conforme a equação (51):

$$N_{SLOR}(\)_{i,j} = \left(E_{\eta}^{-1} - \frac{2}{r}\right) + \frac{1}{r} A_{i,j} \delta_{\xi\xi} \quad (51)$$

$$(A_{i,j} \Delta r_{i+1,j} - 2A_{i,j} \Delta r_{i,j} + A_{i,j} \Delta r_{i-1,j} + r \cdot C_{i,j} \Delta r_{i,j-1} - 2\Delta r_{i,j}) = -r\omega L r_{i,j} \quad (52)$$

$$a_{i-1,j} = 1A_{i,j}^n \quad (53)$$

$$b_{i,j} = -2 - 2A_{i,j}^n \quad (54)$$

$$c_{i+1,j} = 1A_{i,j}^n \quad (55)$$

$$vt_{i,j} = -r\omega L r_{i,j}^n - r \cdot \Delta r_{i,j-1}^n \quad (56)$$

Onde r é o parâmetro de relaxação do método SLOR, e ω é igual a 1. A aplicação deste operador leva a um sistema tridiagonal periódico em " i ", tendo sua solução para o intervalo $2 \leq j \leq j_{MAX} - 1$.

2.4 AF2 (*AF, Approximate Factorization*)

O esquema AF2 consiste em um procedimento de dois passos. As equações (57) e (58) são resolvidas primeiramente por meio do processo de solução de uma tridiagonal periódica e depois resolvendo uma matriz tridiagonal simples.

$$N_{AF2}(\)_{i,j} = -\frac{1}{\alpha} (\alpha \tilde{\delta}_{\eta} - A_{i,j} \delta_{\xi\xi}) (\alpha - C_{i,j} \tilde{\delta}_{\eta}) (\)_{i,j}^n \quad (57)$$

Os operadores *Backward* e *Forward* são representadas pelas equações (58) e (59):

$$\tilde{\delta}_\eta(\)_{i,j} = (\)_{i,j} - (\)_{i,j-1} \quad (58)$$

$$\vec{\delta}_\eta(\)_{i,j} = (\)_{i,j+1} - (\)_{i,j} \quad (59)$$

$$\begin{aligned} & (\alpha \tilde{\delta}_\eta - A_{i,j} \delta_{\xi\xi})(\alpha - C_{i,j} \vec{\delta}_\eta)(\Delta r)_{i,j}^n \\ & = \alpha \omega \left[A_{i,j}^n (r_{i,j}^n)_{\xi\xi} - 2B_{i,j}^n (r_{i,j}^n)_{\xi\eta} + C_{i,j}^n (r_{i,j}^n)_{\eta\eta} \right] \end{aligned} \quad (60)$$

Passo 1:

$$(\alpha \tilde{\delta}_\eta - A_{i,j} \delta_{\xi\xi}) \Delta s_{i,j}^n = \alpha \omega L r_{i,j}^n \quad (61)$$

Passo 2:

$$(\alpha - C_{i,j} \vec{\delta}_\eta) \Delta r_{i,j}^n = \Delta s_{i,j}^n \quad (62)$$

A resolução do primeiro passo leva a um sistema tridiagonal periódico, que tem o seguinte formato:

$$\begin{bmatrix} b_1 & c_2 & & & a_{i_{MAX}-1} \\ a_1 & b_2 & c_3 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{i_{MAX}-3} & b_{i_{MAX}-2} & c_{i_{MAX}-1} \\ c_1 & & & a_{i_{MAX}-2} & b_{i_{MAX}-1} \end{bmatrix} \begin{Bmatrix} \Delta s_{1,j}^{(n)} \\ \vdots \\ \Delta s_{i,j}^{(n)} \\ \vdots \\ \Delta s_{i_{MAX}-1,j}^{(n)} \end{Bmatrix} = \begin{Bmatrix} vt_1 \\ \vdots \\ vt_i \\ \vdots \\ vt_{i_{MAX}-1} \end{Bmatrix} \quad (63)$$

Onde:

$$a_{i-1,j} = -A_{i,j}^n \quad (64)$$

$$b_{i,j} = \alpha + 2A_{i,j}^n \quad (65)$$

$$c_{i+1,j} = -A_{i,j}^n \quad (66)$$

$$vt_{i,j} = \alpha \omega L r_{i,j}^n + \alpha \Delta s_{i,j-1}^n \quad (67)$$

No segundo passo é resolvido um sistema tridiagonal não periódico, em um procedimento semelhante aquele feito no projeto 1.

2.4.1 Resultados

Utilizou-se um computador de uso pessoal do tipo *Laptop* com processador Intel® Core™ i5-3210M CPU @ 2.50GHz com sistema operacional *Windows 7 Home basic* de 64 bits. Os códigos foram desenvolvidos no *Code::Blocks* que trata-se de um *open source* ambiente integrado para desenvolvimento de códigos em linguagens: C, C++ e Fortran.

Aproveitou-se a oportunidade deste projeto para trabalhar em alguns conceitos **básicos** de Orientação a Objetos, alocação dinâmica de memória, de-alocação de memória, passagem de classes e métodos dentro de *functions* e retorno de objetos por meio de *functions* utilizando a linguagem C++. Devido à necessidade constante de avaliação de parâmetros e de experimentos com o código, optou-se por rodar o mesmo no modo *Debug* por meio do compilador GNU *Debugger* ou mais conhecido como GDB.

Todas as simulações numéricas aqui apresentadas foram realizadas utilizando a *double precision* conforme pode ser evidenciado no anexo A onde é apresentado o código fonte deste trabalho. As simulações foram feitas considerando um fator de estiramento $XSF = 1.18$ (estiramento para as linhas na direção ξ) e $YSF = 1.25$ (estiramento para as linhas na direção η). Embora se tenha esperado o resíduo atingir o “zero de máquina”, considerou-se para a geração final da malha que a solução obtida pelos métodos iterativos estaria convergida quando fosse atingida uma tolerância de $10E-12$.

As Figura 1 e 2 mostram a influência do parâmetro α na razão de convergência do método AF1/ADI. Inicialmente, escolheu-se um valor de ω dentro da metade do limite de estabilidade constante igual 1.0 e foi feita a variação do parâmetro α . Conforme discutido nas notas de aula e no *paper* do HOLST, o limite de estabilidade linear para aplicação do AF1 requer um valor de $\alpha \geq 0$ e $0 \leq \omega \leq 2$. Neste sentido será feita a variação dos parâmetros dentro deste intervalo. Apresentar-se-á as curvas de resíduo para a variável x e y considerando o perfil biconvexo.

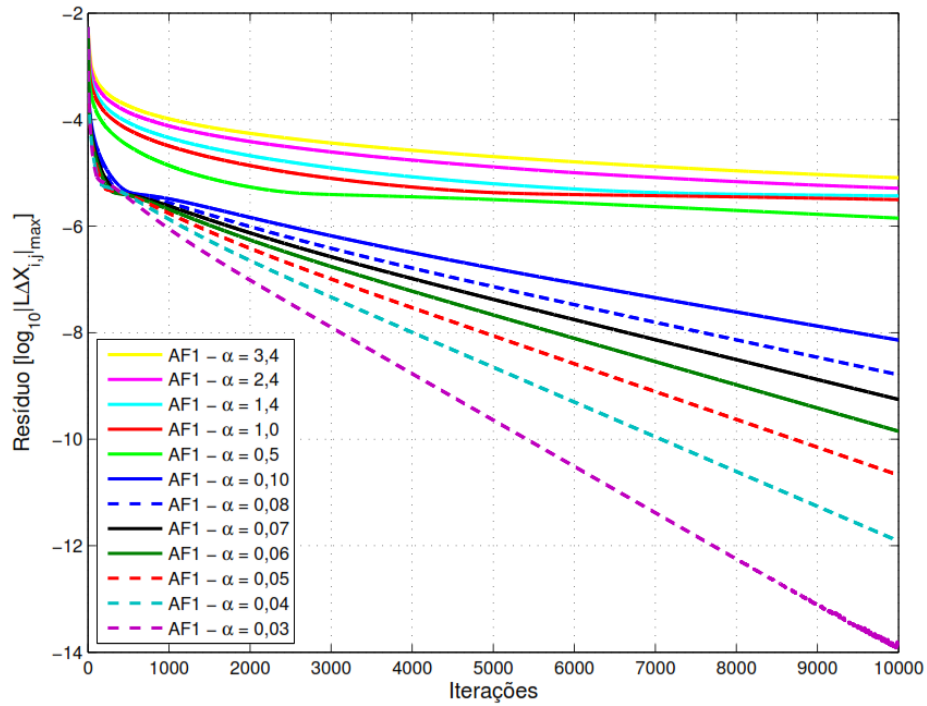


Figura 1: Influência do parâmetro α na razão de convergência para $\omega = 1.0$.

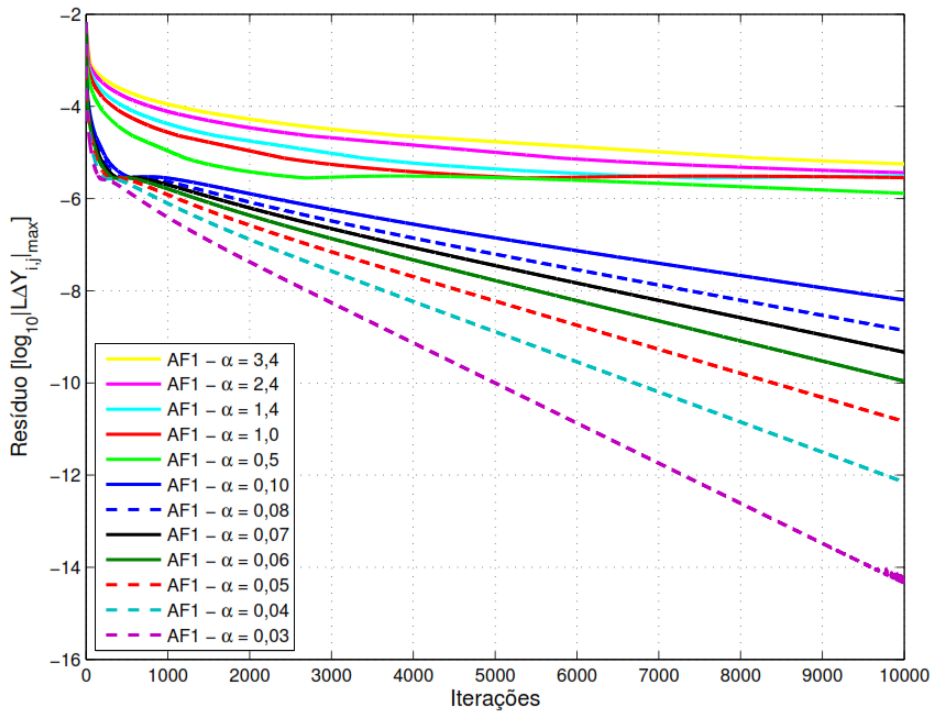


Figura 2: Influência do parâmetro α na razão de convergência para $\omega = 1.0$.

O que podemos observar de imediato nestas figuras acima é que para valores de α mais próximos de zero melhor é a taxa de convergência do método, isso nos dá um “sinal” que os valores de α devem ser pequenos. Percebe-se em uma primeira análise que para valores de

α inferiores a 0.03 o método divergiu, indicando que para cada valor de ω escolhido terá um respectivo valor de α para qual o método irá divergir. As curvas de redução de resíduo tanto para x quanto para y seguem um tendência praticamente igual.

Sendo assim é o momento de investigar a influência do parâmetro de relaxação ω , esta análise esta representada na Figura 3 e Figura 4. Procurou-se variar os valores de ω entre 0 e 2, sempre mantendo o parâmetro de estiramento fixo. Adotou-se o valor de $\alpha = 0.05$ que foi o valor mais adequado para fazer a variação do ω , uma vez que para valores de ω acima da unidade e valores de alfas inferiores a 0,05 o método divergia.

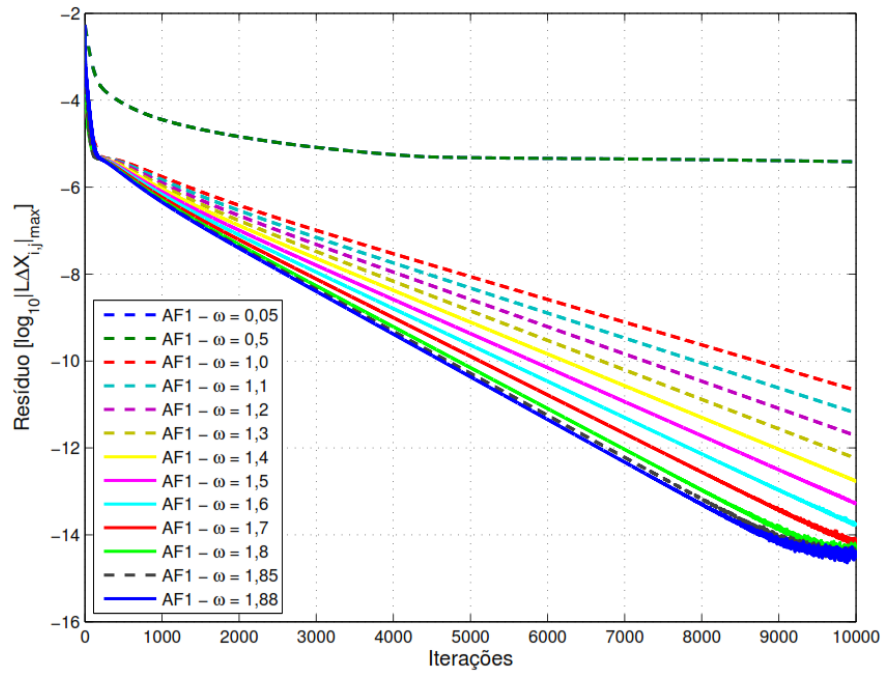


Figura 3: Influência do parâmetro ω na taxa de convergência do método AF1 .

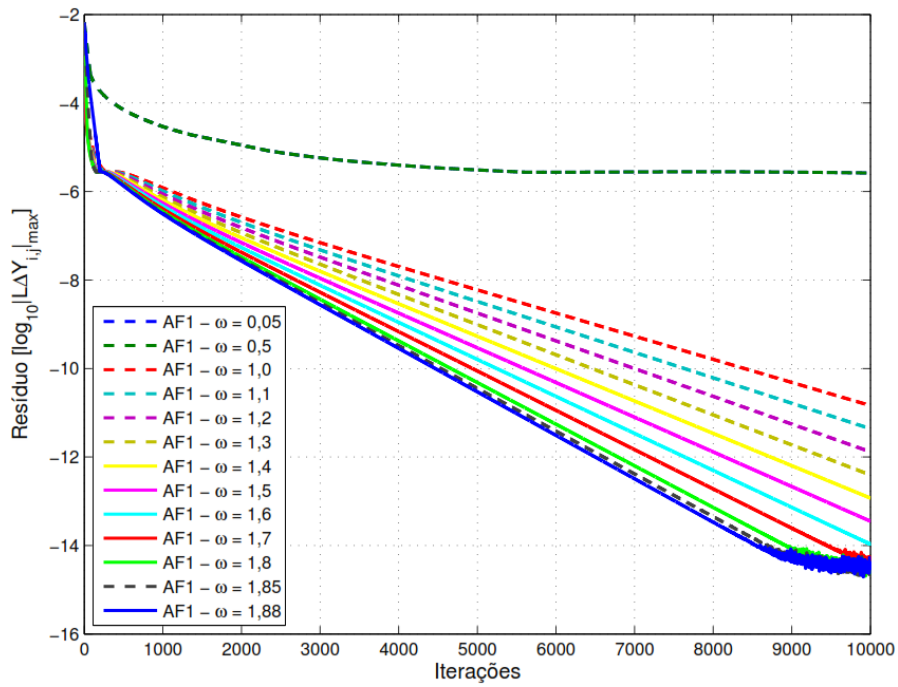


Figura 4: Influência do parâmetro ω na taxa de convergência do método AF1 .

Claramente, valores de ω mais próximos de zero o método até converge, mas se requer um número apreciável de iterações. As análises das figuras acima mostram que para valores de ω próximos ao limite de estabilidade o método apresenta melhores taxas de convergência. Para o valor de $\omega = 1.88$ o método atingiu convergência com aproximadamente 6600 iterações. Valores adequados de ω estão no intervalo entre $1.8 < \omega < 1.9$, curiosamente, estes são os mesmos valores sugeridos e encontrados no *paper* de MORGENSTERN e AZEVEDO (1989).

Deve-se observar neste momento que qualquer mudança nos parâmetros de estiramento pode levar a outras escolhas de α e ω ótimos. Neste sentido achou-se interessante fazer uma análise mantendo constantes $\alpha = 0.05$ e $\omega = 1.88$ e fazer uma variação do parâmetro de estiramento para direção x . O que se observa é que malhas mais estirada (ou seja, XSF próximos a 1,25) leva a um maior número de iterações até a convergência, à medida que se vai reduzindo o parâmetro de estiramento para a proximidade de XSF = 1.10 o método passa a requerer um menor número de iterações para atingir a convergência dentro do critério estabelecido, para aproximadamente 4000 iterações, como pode ser evidenciado por meio da Figura 5 e Figura 6.

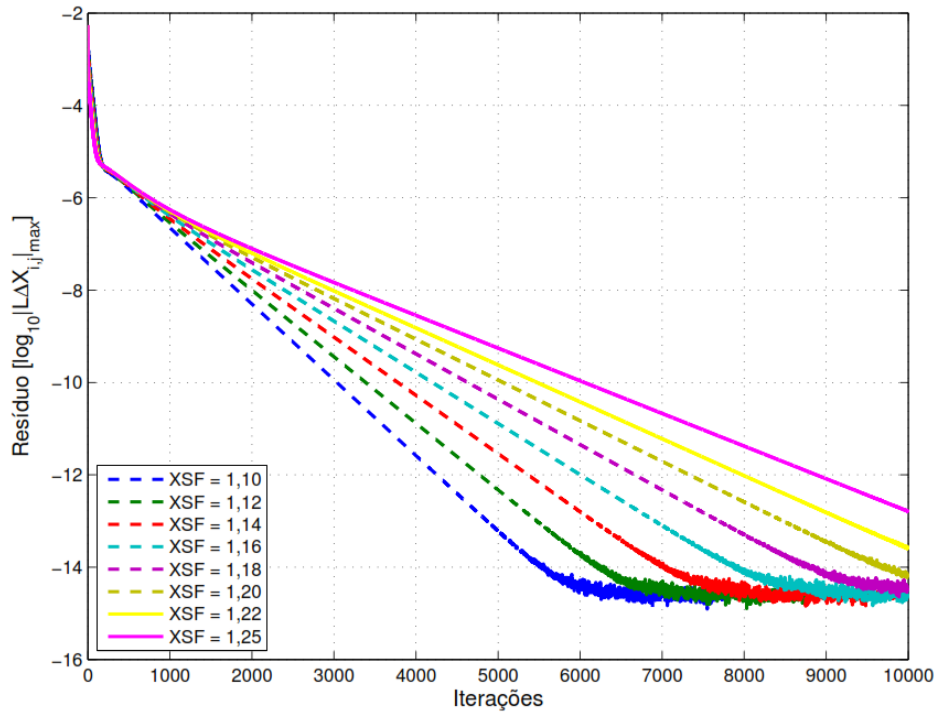


Figura 5: Influência do parâmetro de estiramento XSF na taxa de convergência geral do método AF1.

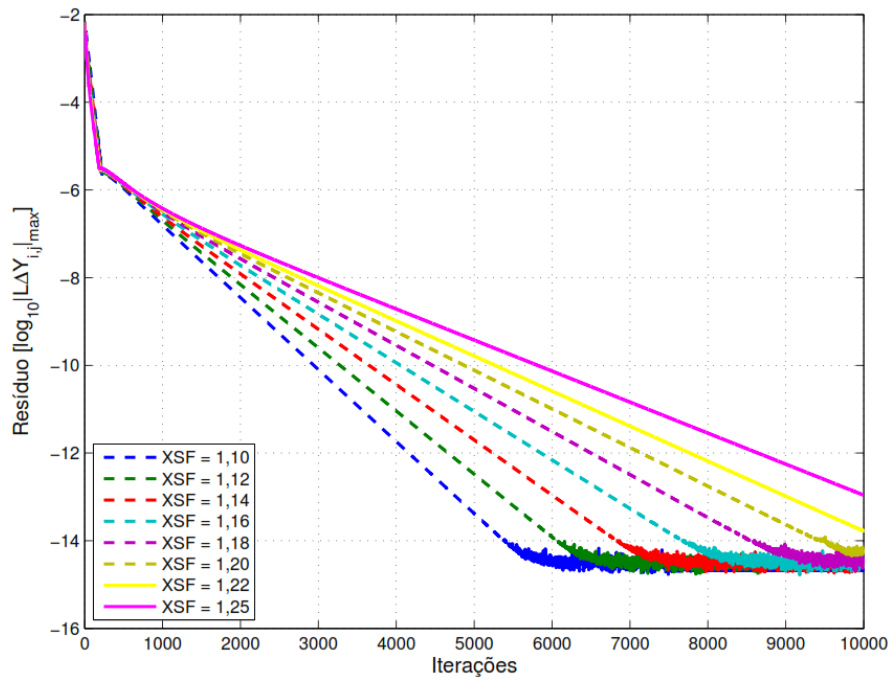


Figura 6: Influência do parâmetro de estiramento XSF na taxa de convergência geral do método AF1.

É momento de apresentar as malhas finais parabólicas e elípticas, considerando estiramento de $XSF = 1.10$. MORGENSTERN e AZEVEDO (1990, página 186), mostram visualizações de malhas, primeiro é apresentado uma “*General view of the mesh*”, depois uma figura mostrando os detalhes da malha próximo aos bordos de ataque e fuga e por final uma

comparação (por meio de uma sobreposição de malhas) cujo objetivo neste caso foi comparar a malha algébrica e a malha elíptica. Neste trabalho será sobreposto a malha parabólica com a malha Elíptica.

Sendo assim é possível observar por meio da Figura 7 a visão geral da malha parabólica para o perfil biconvexo, a Figura 8 uma vista do bordo de fuga e a Figura 9 a vista do bordo de ataque. As Figura 10Figura 11 Figura 12 mostram a malha elíptica para o mesmo perfil.

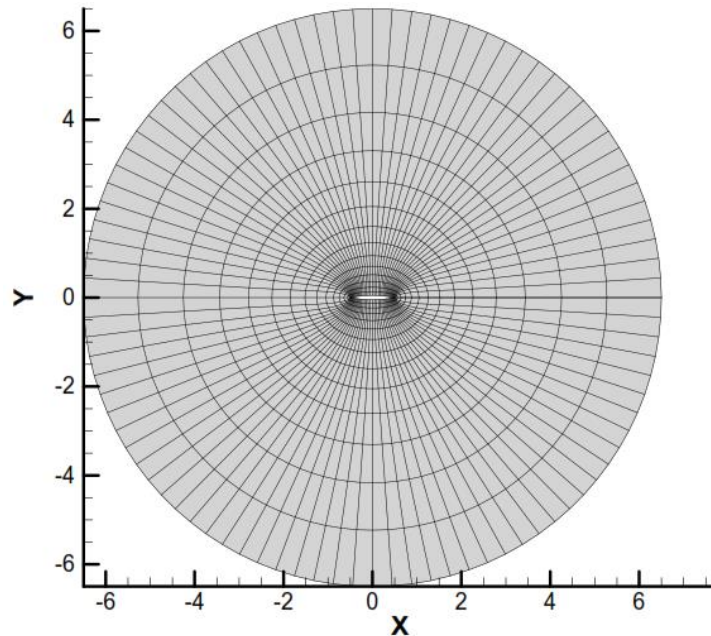


Figura 7: Visão Geral da malha para o perfil biconvexo obtida com o gerador parabólico.

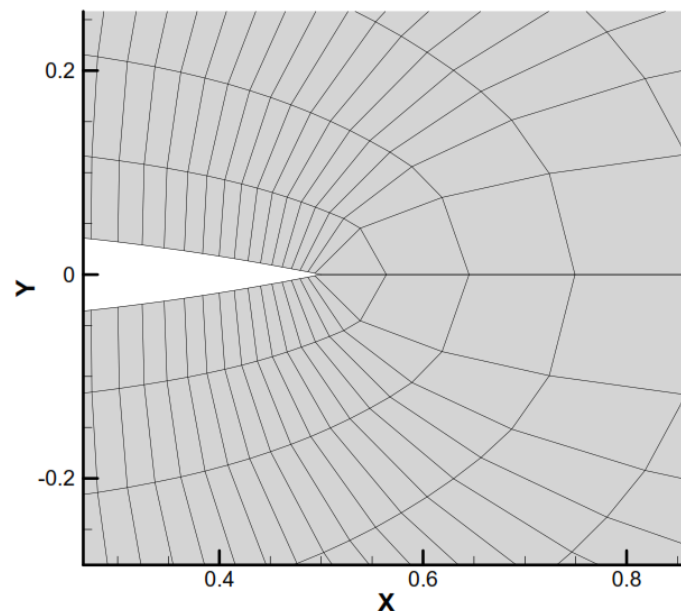


Figura 8: Visão específica do bordo de fuga para o perfil biconvexo obtida com o gerador parabólico.

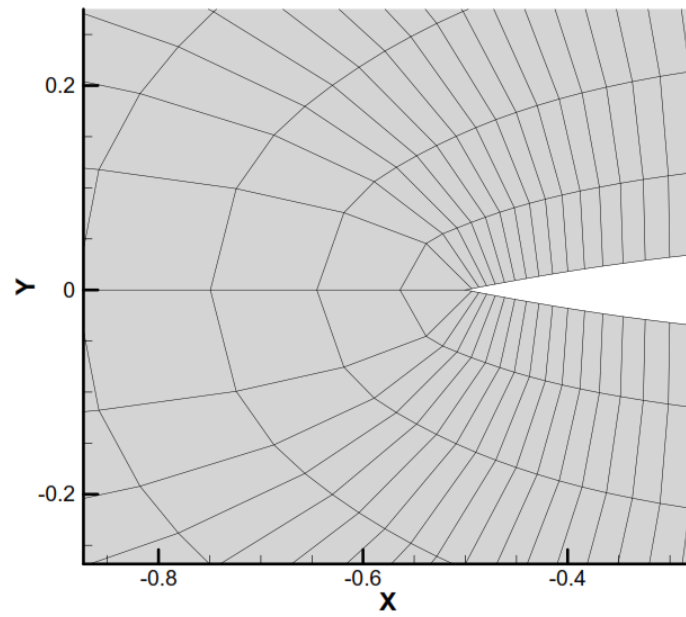


Figura 9: Visão específica do bordo de ataque para o perfil biconvexo obtida com o gerador parabólico.

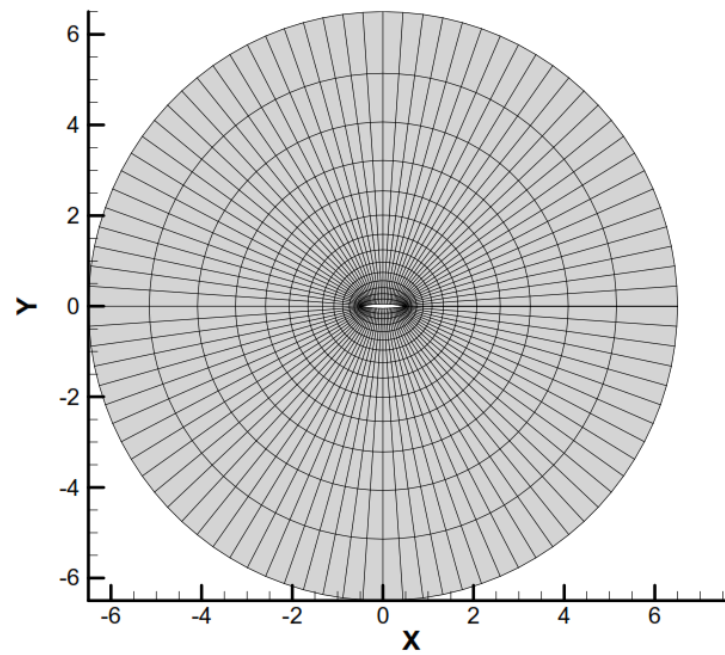


Figura 10: Visão Geral da malha para o perfil biconvexo obtida com o gerador Elíptico.

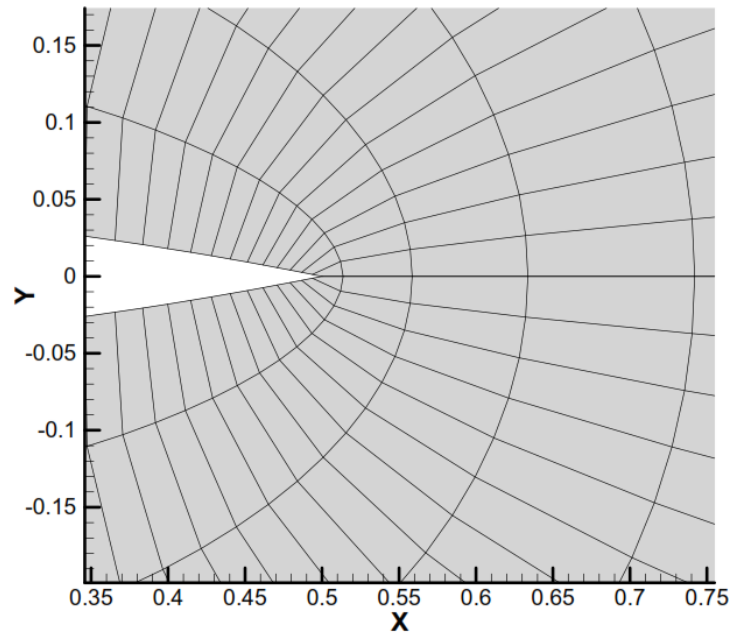


Figura 11: Visão específica do bordo de fuga para o perfil biconvexo obtida com o gerador Elíptico.

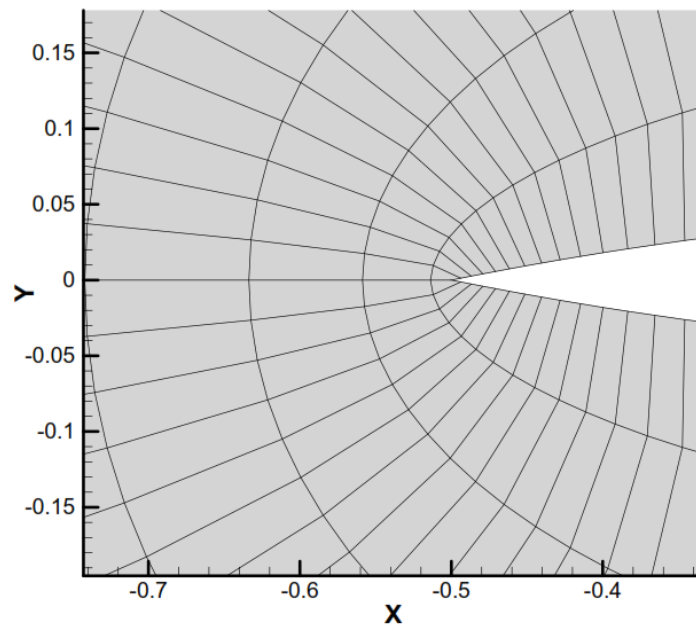


Figura 12: Visão específica do bordo de ataque para o perfil biconvexo obtida com o gerador Elíptico.

É observada por meio da Figura 13 a comparação da malha para o perfil biconvexo gerada por ambos os processos, tanto o elíptico quanto o parabólico. A malha elíptica esta representada por linhas do tipo *dash dot dot*.

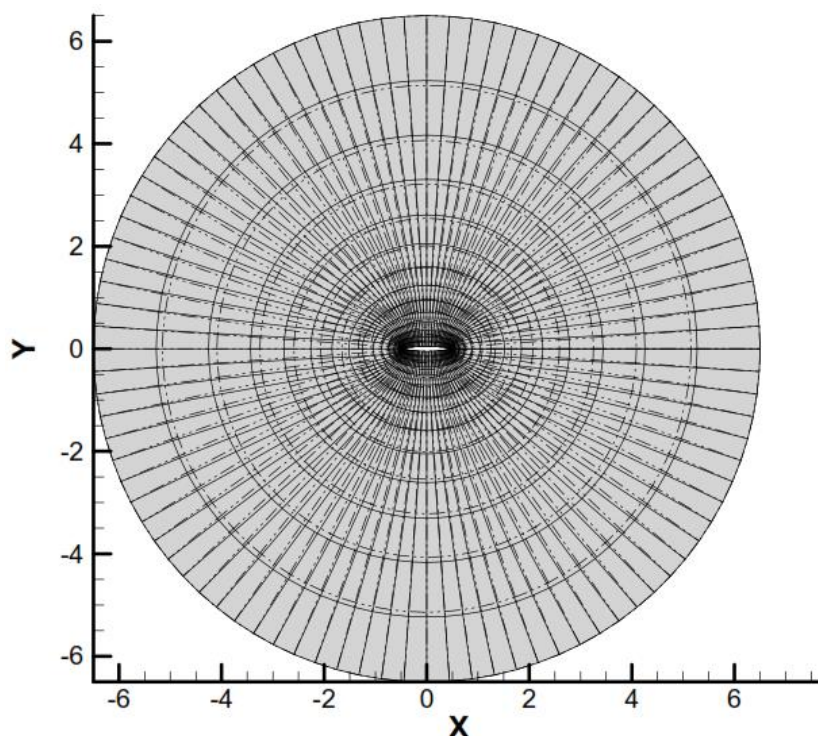


Figura 13: Comparação das malhas geradas pelos dois processos para o perfil biconvexo.

Percebemos que a distribuição dos pontos na malha obtida pelo gerador elíptico é mais suave quando comparada a malha parabólica, entretanto, é de se observar também a clara perda de ortogonalidade por parte da malha elíptica próxima a parede do perfil. Essa ortogonalidade é verificada com certa veemência na malha parabólica, que teve seu próprio processo de geração já pensado em se garantir ortogonalidade próximo ao perfil, como já discutido acima no relatório e também discutido no *paper* do NOACK e ANDERSON. Uma ampliação da área próxima ao bordo de fuga esta representada por meio da Figura 14. Nenhuma descontinuidade de linhas foi verificada.

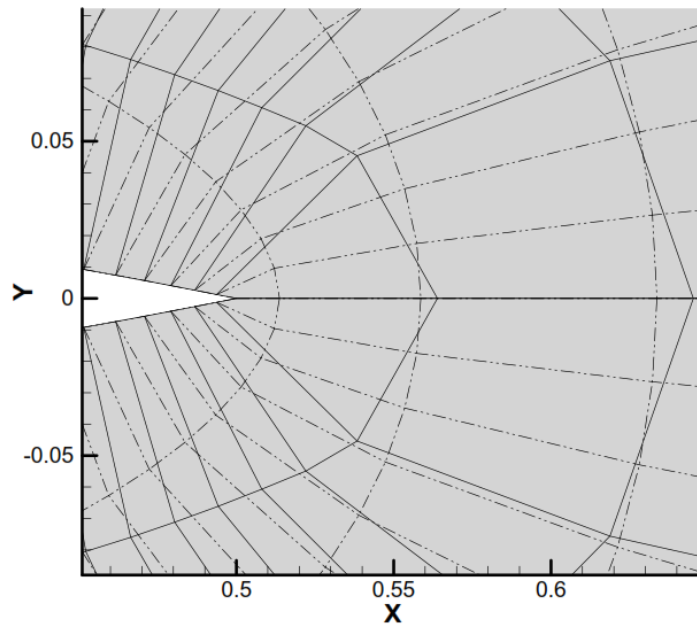


Figura 14: Comparação das malhas geradas pelos dois processos para o perfil biconvexo, bordo de fuga .

Embora a malha para o NACA 0012 seja requisitada por meio do item 4 da proposta do projeto, aproveitou-se de antemão para apresentar por meio da Figura 15 e Figura 16 uma visão geral da malha gerada pelo método parabólico e elíptico respectivamente.

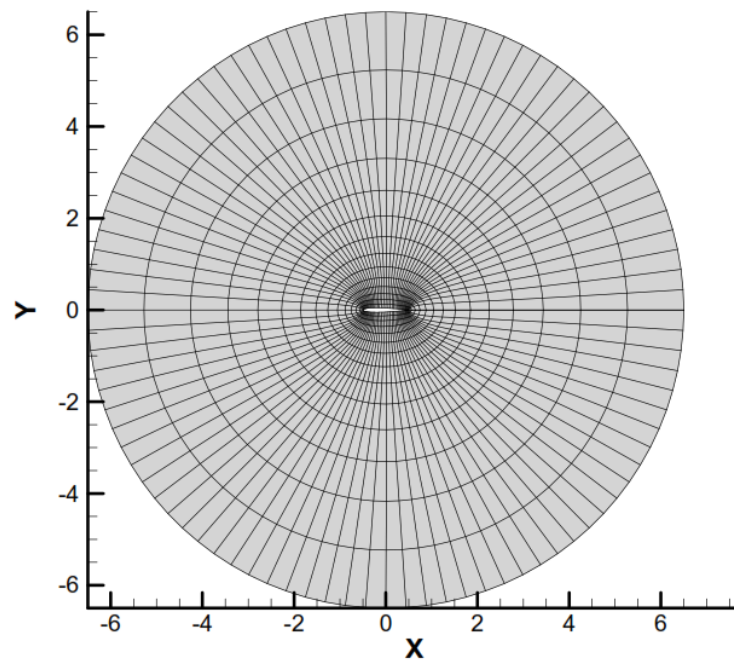


Figura 15: Visão Geral da malha para o perfil NACA 0012 obtida com o gerador parabólico.

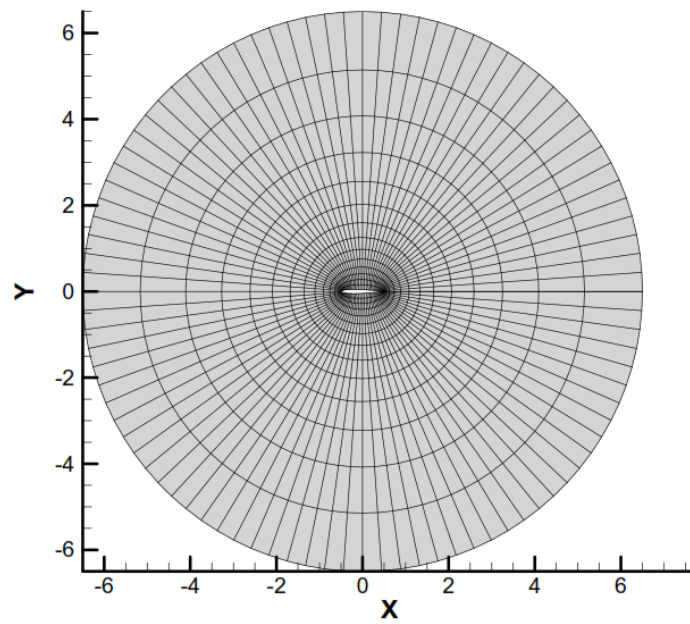


Figura 16: Visão Geral da malha para o perfil NACA 0012 obtida com o gerador Elíptico.

A Figura 17 representa a visão geral das malhas parabólicas e elípticas sobrepostas, enquanto a Figura 18 procura representar o detalhe da sobreposição próximo ao bordo de ataque.

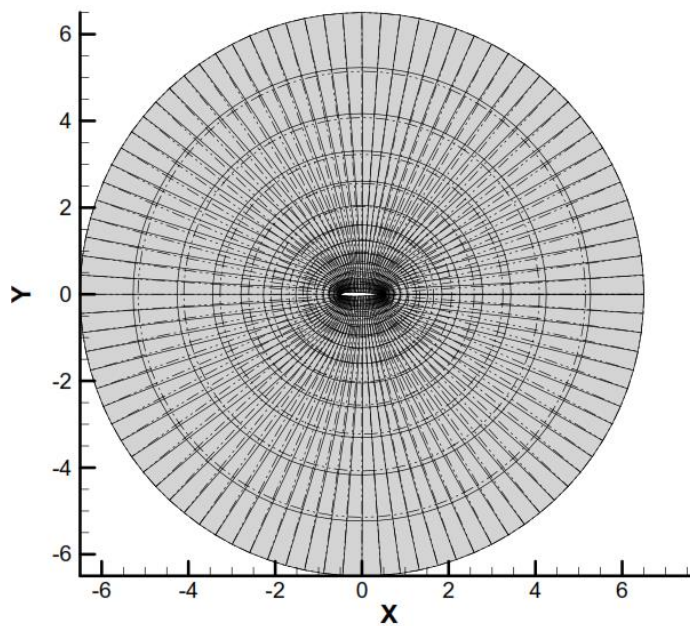


Figura 17: Comparação das malhas geradas pelos dois processos para o perfil NACA0012.

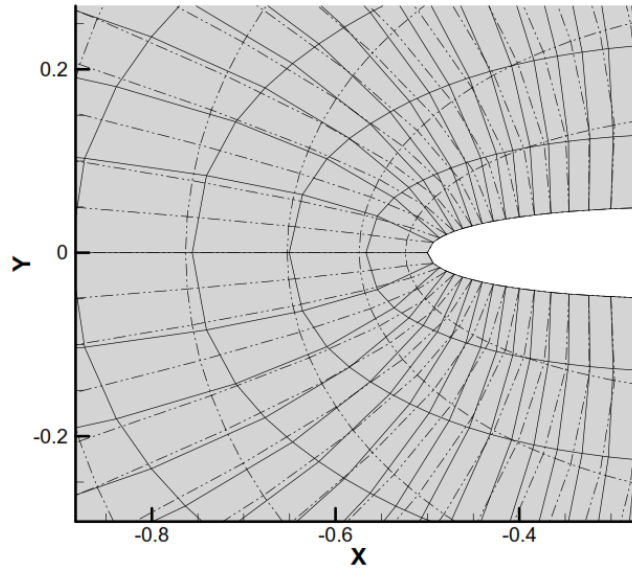


Figura 18: Comparação das malhas geradas pelos dois processos para o perfil NACA 0012, bordo de ataque .

2.4.2 Aplicação dos métodos AF2 e AF1 e comparação dos seus respectivos históricos de convergência e tempo computacional

Os históricos de convergência e otimização dos parâmetros α e ω para o método AF1 foram feitos levando em conta o perfil biconvexo. Uma comparação de mesma natureza será feita utilizando o método AF2, programou-se também um contador de tempo computacional, ao término será feita uma análise comparando o custo computacional destes dois métodos. O que será feito inicialmente é manter fixo o valor de $\omega = 1.88$ e variar o parâmetro α para o método AF2, para avaliar a influência deste parâmetro na taxa de convergência do método, conforme pode ser observado por meio da Figura 19.

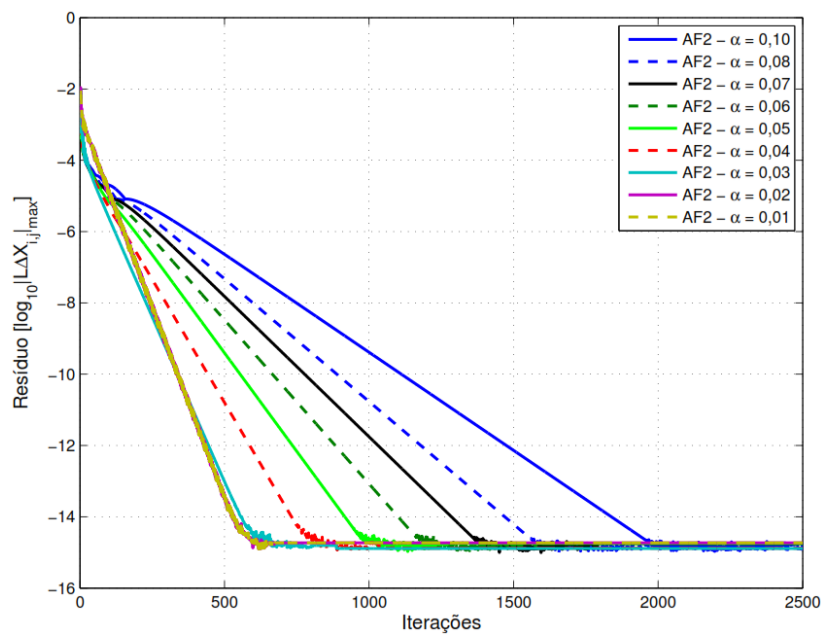


Figura 19: Influência do parâmetro α na razão de convergência do método AF2 para $\omega = 1.88$.

O método AF2 diverge para valores menores a $\alpha = 0.01$, entretanto o método apresenta melhores taxas de convergência quando comparado ao método AF1, quando considerado o valor de $\alpha = 0.05$ e $\omega = 1.88$. Enquanto o método AF1 necessitou de aproximadamente 6600 iterações para atingir a convergência o método AF2 utilizando parâmetro α e ω iguais convergiu com aproximadamente 737 iterações. Poderia também para o método AF2 utilizar um valor ainda menor de α e obter convergência com menos iterações.

A Figura 20 é um gráfico comparativo entre os métodos SLOR (utilizando $r = 1.9$), AF1 e AF2 utilizando $\alpha = 0.05$ e $\omega = 1.88$ (XSF = 1.18 e YSF = 1.25). Neste trabalho não se preocupou em explorar a influência do parâmetro r do método SLOR, pois uma análise mais detalhada já fora apresentada no relatório do projeto 1. Deve-se pautar que o método SLOR não atingiria convergência para um erro de $10E-12$, mesmo que se deixasse o programa rodar, por exemplo, 20 mil iterações ou mais.

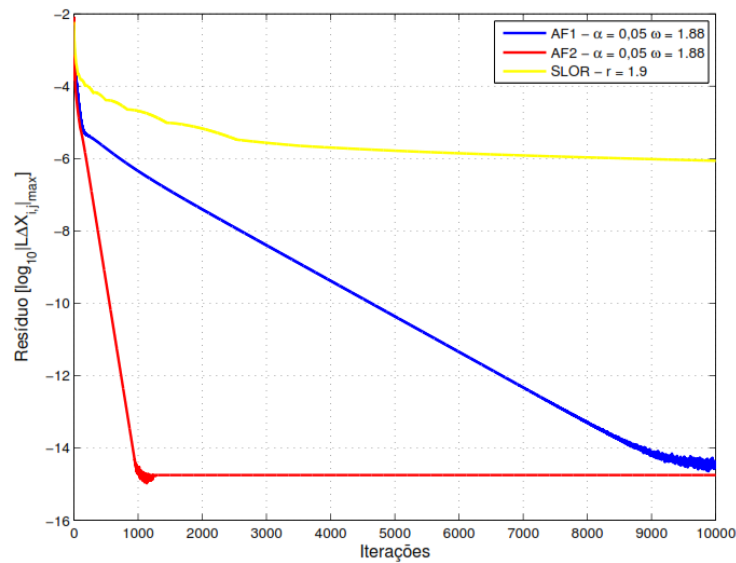


Figura 20: Comparativo dos métodos SLOR, AF1 e AF2.

As eficiências comparadas na tabela 1 abaixo entre o método AF1 e AF2 são em termos de iterações para convergência e tempo de CPU para convergência. Programou-se nesse caso um laço do tipo *while*, onde, o procedimento iterativo é interrompido quando se atinge a tolerância de $\epsilon_{ps} = 1 * 10E - 12$. Estes valores estão resumidos na tabela 1. Considerou-se também a variação do parâmetro ω para ambos os métodos mantido o parâmetro α constante igual a 0,05 e (XSF = 1.18 e YSF = 1.25).

AF1			
$\alpha = 0,05$			
ω	Número total de iterações	Tempo Total	Tempo gasto por iteração (s)
1,88	6669	4,295	0,000644025
1,78	7043	4,538	0,000644328
1,68	7463	4,782	0,000640761
1,58	7935	5,026	0,000633396
1,48	8472	5,321	0,000628069
AF2			
$\alpha = 0,05$			
ω	Número total de iterações	Tempo Total	Tempo gasto por iteração (s)
1,88	737	0,465	0,000630936
1,78	778	0,491	0,000631105
1,68	825	0,519	0,000629091
1,58	878	0,557	0,000634396
1,48	938	0,59	0,000628998

Tabela 1 – Comparativo do custo computacional entre os momentos AF1 e AF2

O que se pode perceber com essa análise do tempo computacional, é que de fato o método AF2 é mais eficiente que o método AF1 no que diz respeito tanto ao tempo total de execução e ao número de iterações, e caracteriza-se como mais vantajoso no sentido de ter sido o método mais “barato”, quando comparado o tempo gasto por iteração. Deve-se pautar que não é sempre (ou seja, para “qualquer” caso simulado) que o método AF2 será melhor que o AF1, e isso foi evidenciado quando se utilizou valores de ω menores (Figura 21: Tempo gasto por iteração métodos AF1 e AF2. Contudo essa análise nos dá um indicativo que para o AF1 e o AF2 operando nas suas respectivas regiões ótimas o método AF2 será necessariamente mais barato computacionalmente que o AF1.

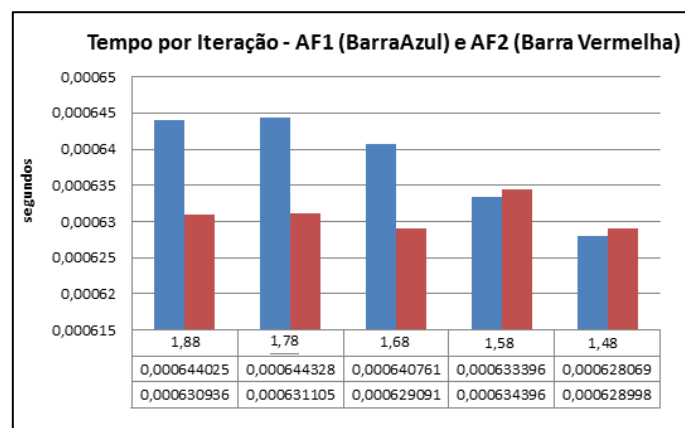


Figura 21: Tempo gasto por iteração métodos AF1 e AF2.

2.4.1 Estudo da aplicação da sequência de Alfas para os métodos AF2 e AF1

Observou-se de antemão neste trabalho que utilizar os valores de α_L e α_H conforme sugerido pelo BALLHAUS, JAMESON e ALBERT (1978), ou seja, atribuindo valores tanto para α_L quanto α_H dentro das ordens de grandezas estabelecidas pelos mesmos, não seria possível obter melhorias no quesito redução do número de iterações. Ao conversar com o Professor em sala de aula foi sugerido trabalhar com essas sequências adotando valores bem pequenos tanto para α_L e α_H próximo de 0.1, 0.01 ou 0.001. A Figura 22 mostra uma primeira análise do comparativo da curva de convergência para o método AF1 com os parâmetros considerados ótimos $\omega = 1.88$ e $\alpha = 0.05$ e adotando sequências de alfas.

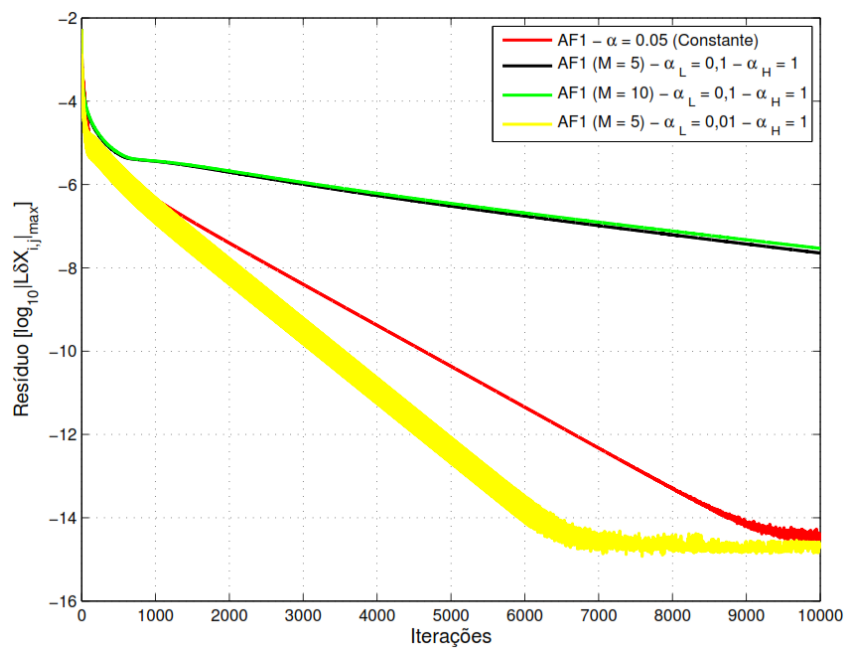


Figura 22: Aplicação da sequência de alfas para o método AF1.

A curva em vermelho é a curva para o método operando com um valor fixo de α , ou seja, α igual 0.05 e $\omega = 1.88$. Testou-se inicialmente uma sequência de α 's com $\alpha_L = 0.1$ e $\alpha_H = 1$ e utilizou-se uma sequência de $M = 5$ e $M = 10$. O que se percebe para este caso é que aumentar o número de α 's na sequência não levou a uma melhoria na convergência. Utilizando assim sequências de $M = 5$ e reduzindo-se em uma ordem de grandeza o valor de α_L para $\alpha_L = 0.01$ e manteve-se fixo o valor de $\alpha_H = 1$. O que se percebe é que ganhou em redução do número de iterações, embora a curva em amarelo apresente bastantes oscilações. Em termos de iterações para o critério de convergência pré-estabelecido, utilizando a sequência de α 's foi possível obter uma redução de mais de 2100 iterações.

Outra forma sugestiva consiste em manter uma sequência de α 's fixa e variar o parâmetro de relaxação ω para valores superiores a 1.88. Embora não se tenha apresentado o gráfico destes resultados, poucas melhorias foram obtidas variando o ω para valores superiores a 1.9 e para valores superiores a 1.94 o método divergiu.

A Figura 23 mostra o histórico de convergência do método AF2 utilizando seqüências de α 's conforme sugerido por BALLHAUS, JAMESON e ALBERT (1978). Em uma primeira análise procurou-se simular utilizando valores de α_L e α_H conforme sugerido por JAMESON (utilizando as mesmas ordens de grandeza)

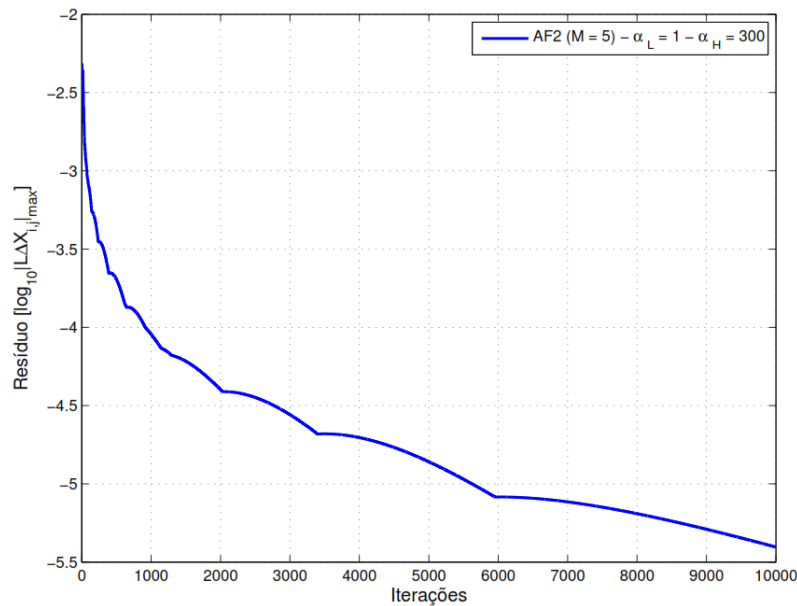


Figura 23: “Picos” na curva de resíduo devido ao processo de aniquilação de autovetores utilizando seqüência de alfas para o método AF2.

Percebe-se que esta solução não está devidamente convergida, mas o objetivo de gerar essa curva, está no fato que foi observado no *paper* do BALLHAUS, JAMESON e ALBERT (1978) na página 33 (Figura 3), quando o mesmo compara o método SLOR com o método AF2 utilizando seqüência de α 's que era de se esperar na curva do AF2 com seqüências de alfas certos “picos” durante o procedimento iterativo de cálculo do resíduo (obs: o fenômeno na curva de resíduo encontrado aqui é praticamente idêntico ao encontrado por JAMESON, embora se esteja resolvendo problemas muito diferentes do ponto de vista do equacionamento e física).

Por meio de uma citação direta do *paper* do BALLHAUS, JAMESON e ALBERT (1978) (página 30, décimo terceiro parágrafo-13º) tem-se: “*The peaks in the AF-2 convergence history correspond to the smaller values of α (i.e., larger values of Δt) in the eight-element sequence (no caso do presente projeto 2 no quinto elemento da seqüência).*”

A Figura 24 mostra os resultados obtidos para o método AF2 onde se buscou uma seqüência de alfas “ótima”. Para este problema em específico objetivou-se em buscar o menor número de iterações, para valores menores de α_L o problema diverge, e para valores maiores para α_L e ou α_H o método requerer um número maior de iterações, manteve-se fixo o $\omega = 1.88$.

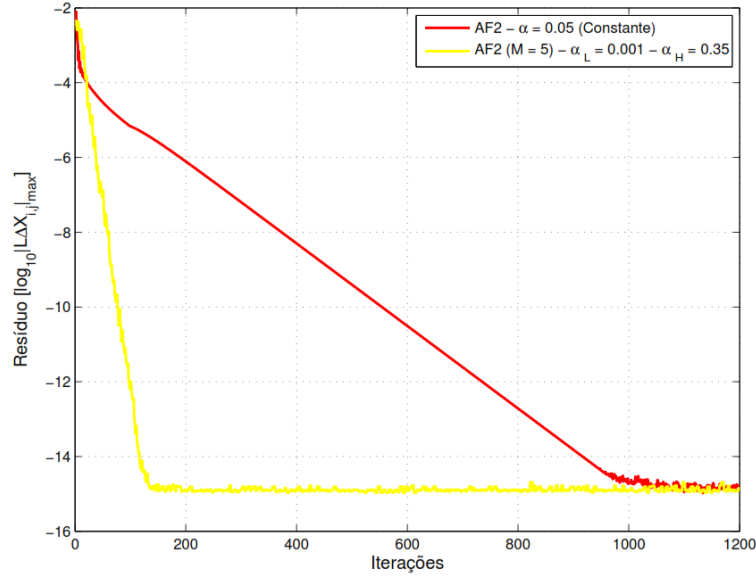


Figura 24: Sequência de alfas para o método AF2.

Utilizando esta sequência de alfas foi possível obter soluções convergidas para o perfil biconvexo com aproximadamente 97 iterações.

2.4.2 Considerações sobre a implementação das funções P e Q

A última parte requisitada neste Projeto 2 (dentro os 4 itens) é referente a utilização das funções de atração de pontos e linhas, funções P e Q. A utilização das funções P e Q é de grande valia durante a geração de malhas estruturadas, pois permitem atrair linhas para regiões de interesse, por exemplo, aproximar linhas na parede do perfil e/ou concentrar pontos nas proximidades dos bordos de ataque e fuga, entre outras.

Em THOMPSON (1977a) é apresentado às equações (68) e (69): (um equacionamento semelhante pode ser encontrado no livro do ANDERSON e PLETCHER (1984)), Thompson também desenvolveu um código chamado TOMCAT (NASA CR-2729) onde apresenta os estudos destas funções e por último, no website da *Mississippi State University (High Performance Computer Laboratory)* é possível encontrar o livro deste mesmo autor em formato *pdf*, onde esta escrito essas funções.

$$\begin{aligned}
 P(\xi, \eta) = & - \sum_{l=1}^L a_l \operatorname{sgn}(\xi - \xi_l) \exp(-c_l |\xi - \xi_l|) - \\
 & - \sum_{m=1}^M b_m \operatorname{sgn}(\xi - \xi_m) \exp \left[-d_m [(\xi - \xi_m)^2 + (\eta - \eta_m)^2]^{\frac{1}{2}} \right]
 \end{aligned} \tag{68}$$

$$Q(\xi, \eta) = - \sum_{l=1}^L a_l \text{sgn}(\eta - \eta_l) \exp(-c_l |\eta - \eta_l|) - \sum_{m=1}^M b_m \text{sgn}(\eta - \eta_m) \exp \left[-d_m [(\xi - \xi_m)^2 + (\eta - \eta_m)^2]^{\frac{1}{2}} \right] \quad (69)$$

Onde:

$$\text{sgn}(x) = 1 \text{ se } x \text{ é positivo} \quad (70)$$

$$\text{sgn}(x) = 0 \text{ se } x = 0 \quad (71)$$

$$\text{sgn}(x) = -1 \text{ se } x \text{ é negativo} \quad (72)$$

O primeiro termo do lado direito da equação da função P tem o efeito de mover $\xi =$ linhas constantes até a direção da linha $\xi = \xi_l$ e o primeiro termo para a função Q tem o efeito de mover linhas $\eta =$ linhas constante até a direção de $\eta = \eta_l$. O segundo termo em ambas as equações tem a função de atrair linhas ξ e η para os pontos $(\xi_m$ e $\eta_m)$, o segundo termo dessas equações que é utilizado para atrair pontos em direção aos bordos de fuga e de ataque.

De uma maneira geral valores mal dimensionados para P e Q podem levar facilmente ao *blow-up* do processo iterativo, além disso, o correto funcionamento dessas funções esta estritamente relacionado com adequados valores de chute inicial para x e y (THOMPSON (1977a)).

Neste presente trabalho tentou-se implementar essas funções por meio da *function AF2Attraction*, infelizmente não foi possível obter resultados expressivos, o que foi possível observar foi apenas resultados um tão pouco triviais, como por exemplo, a influência do sinal da função P ou Q na geração final do grid, mantendo-se alguns valores fixos durante o procedimento iterativo.

Percebe-se que a implementação dessas funções não é tão trivial “quanto” parece e ajustar os coeficientes não é uma tarefa tão simples, o procedimento deixa de convergir (“explode”) muito facilmente. Neste contexto, seria necessário um maior tempo para estudar a implementação dessa função e estudar mais a fundo a influência destes parâmetros, começando, por exemplo, com a implementação de um caso mais simples.

3 COMENTÁRIOS FINAIS

O código desenvolvido para este Projeto 2 e disponibilizado no anexo A, esta apresentando resultados finais para a geração de malha parabólica e elíptica (malha suavizada e/ou “trabalhada”) coerentes com aqueles encontrados nas principais referências bibliográficas indicadas pelo professor. Neste sentido acredita-se que o objetivo principal proposto foi atendido, ou seja, a oportunidade de praticar a implementação de geradores de malhas computacionais estruturadas baseados na solução de equações diferenciais.

Aproveitou-se para implementar e estudar a aplicação dos métodos AF1 e AF2, onde concluiu-se que para uma ampla faixa de escolha de ω e α o método AF2 se apresentou mais barato computacionalmente. Constatou-se que a escolha de um adequado parâmetro de relaxação e a busca por um valor ótimo de α é fundamental para acelerar a convergência dos métodos, e que todos os métodos quando trabalhados próximos aos limites de estabilidade acabam por divergir.

Com esse código é possível avaliar 3 dos 4 estudos sugeridos, ou seja, gerar uma malha com topologia tipo “O” em um perfil NACA 0012, implementar o método AF2 e comparar sua eficiência em relação ao método AF1 e estudar a utilização das sequências de alfas em um contexto de geração de malhas estruturadas, a implementação das funções P e Q ainda carecem de uma melhor análise. Neste aspecto, foi possível atender a exigência de realizar dois estudos de caso para o projeto 2.

Sendo assim, o(s) resultado(s) deste projeto, ou seja, a(s) malha(s) computacional sobre o perfil biconvexo será(ão) utilizada(s) no Projeto 3 e toda as análises da variação dos parâmetros ω e α , a utilização da sequência de α 's e a geração de malha no NACA 0012 poderão posteriormente serem estendidas em um contexto da solução da equação do potencial completa. Será possível realmente avaliar a influência das questões relacionadas com a ortogonalidade e a não ortogonalidade de malha, suavidade de malhas elípticas, refinamento de malha, entre outros, nos resultados finais de C_p .

4 REFERÊNCIAS BIBLIOGRÁFICAS

Anderson, D.A., Tannehill, J.C., and Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer*, McGraw-Hill, New York, 1984.

Uller, M., e Azevedo, J.L.F., “Grid Generation Technique Effects on Transonic Full Potential Solutions of Airfoil Flows,” *Anais do XI Congresso Brasileiro de Engenharia Mecânica– XI COBEM*, Vol. “Azul”, São Paulo, SP, Dez. 1991, pp. 197-200.

Noack, R.W., and Anderson, D.A., “Solution-Adaptive Grid Generation Using Parabolic Partial Differential Equations,” *AIAA Journal*, Vol. 28, No. 6, June 1990, pp. 1016-1023.

Holst, T.L., “Approximate-Factorization Schemes for Solving the Transonic Full-Potential Equation,” In *Advances in Computational Transonics*, W.G. Habashi, editor, Pineridge Press, Swansea, UK, 1985, pp. 59-82.

Holst, T.L., “Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh,” *AIAA Journal*, Vol. 17, No. 10, Oct. 1979, pp. 1038-1045.

Morgenstern, A., Jr., Azevedo, J.L.F., and de Mattos, B.S., “Two Dimensional Full Potential Solutions of Lifting Airfoil Flows,” *Proceedings of the 3rd Brazilian Thermal Sciences Meeting, Vol. I, Itapema, SC*, Dec. 1990, pp.183-188.

Morgenstern, A., Jr., Azevedo, J.L.F., and de Mattos, B.S., “Influence of Convergence Parameters and Mesh Refinement on Full-Potential Solutions of Airfoil Flows,” *Anais do X Congresso Brasileiro de Engenharia Mecânica*, Vol. I, Rio de Janeiro, Dez. 1989, pp. 149-152.

Fletcher, C.A.J., *Computational Techniques for Fluid Dynamics 2 – Specific Techniques for Different Flow Categories*, Springer-Verlag, New York, 1988

Thompson, J.F., Thames, F.C and Mastin, C.M. “Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies”. *Journal of Computational Physics*. Vol. 15, 1974, pp. 299-319

5 ANEXO A