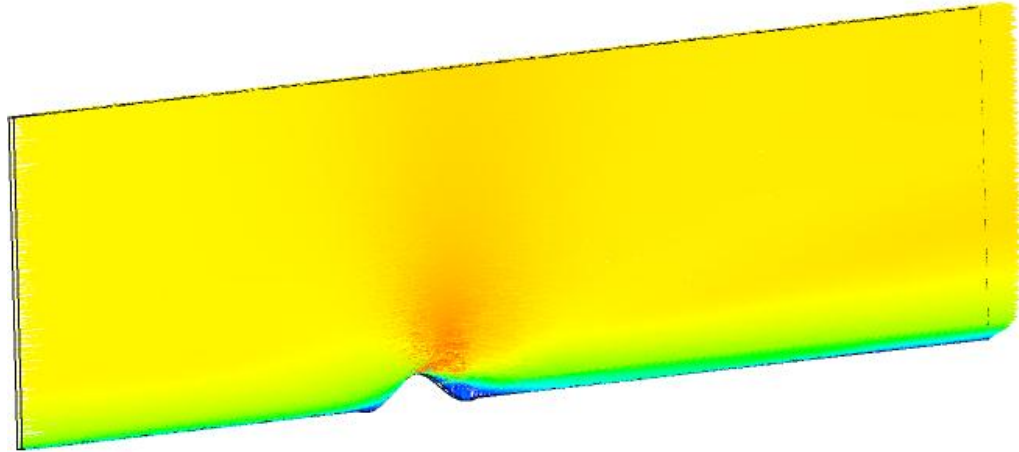




FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE
ENGENHARIA MECÂNICA



Modelação Computacional da Erosão Eólica de Pilhas Bidimensionais Constituídas Por um Material Granular

Dissertação apresentada para a obtenção do grau de Mestre em Engenharia
Mecânica na Especialidade de Sistemas de Produção

Autor

Vasco Rafael Paraíso Mateus

Júri

Presidente Professor Doutor Pedro de Figueiredo Vieira Carvalheira
Professor Auxiliar da Universidade de Coimbra

Vogal Professor Doutor António Manuel Gameiro Lopes
Professor Auxiliar da Universidade de Coimbra

Orientador Professor Doutor Almerindo Domingues Ferreira
Professor Auxiliar da Universidade de Coimbra

Coimbra, Setembro, 2012

“Scientists study the world as it is, engineers create the world that never has been.”

Theodore von Kármán

Aos meus pais.

Agradecimentos

Esta página é dedicada a todos aqueles que de alguma forma, direta ou indiretamente, contribuíram para a elaboração do trabalho que aqui se apresenta. Embora não me seja possível nomear todos, gostaria de agradecer a colaboração e apoio de algumas pessoas, às quais não posso deixar de prestar o meu reconhecimento.

Gostaria de agradecer:

- ao Professor Doutor Almerindo Domingues Ferreira, pela confiança em mim depositada, pelo fornecimento e dissertação de informação útil, bem como pelas condições de trabalho que me possibilitou;
- aos meus colegas Ricardo Trindade e Hugo Beja, pelo desafio lançado e pelo companheirismo partilhado durante a mobilidade ERASMUS na *Politechnika Łódzka* (Polónia), período de extraordinárias vivências e do qual guardo as melhores recordações;
- aos meus pais, pelos valores e apoio transmitidos, e pelo esforço investido para me proporcionar o percurso académico que culmina neste trabalho.

A todos, o meu muito obrigado!

Resumo

O presente trabalho foi levado a cabo no âmbito da Dissertação de Mestrado do MIEM da FCTUC.

O seu objectivo é o desenvolvimento de algoritmos e métodos numéricos, a implementar em rotinas FORTRAN, para a simulação numérica do fenómeno da erosão eólica de pilhas sinusoidais constituídas por um material granular, que contribuam para testar diversos modelos de erosão existentes. Os resultados computacionais obtidos deverão mostrar um abaixamento da superfície da colina. Serão considerados vários modelos. O primeiro modelo introduzido considera um abaixamento constante da colina ao longo de todo o sub-domínio e em cada intervalo de tempo. Apresentam-se também os códigos que utilizam outros dois modelos que reproduzem mais fielmente o processo da erosão, sendo eles o modelo de Bagnold (1941) e o de Lettau e Lettau (1978).

Os vários desenvolvimentos propostos são implementados no programa de simulação numérica de escoamentos ANSYS CFX, com recurso a rotinas FORTRAN programadas pelo utilizador.

Palavras-chave: ANSYS CFX, User Routines, Erosão Eólica, Pilhas Sinusoidais, Material Granular.

Abstract

This work was carried out in the framework of the FCTUC's Master Thesis in Mechanical Engineering.

Its aim is the development of numerical methods and algorithms to be implemented in FORTRAN routines for the numerical simulation of the phenomenon of wind erosion of sinusoidal piles made of a granular material, in order to customize and improve the existing numerical model. The computational results should show a lowering of the surface of the hill. Various models will be considered. The first model introduces a lowering of the hill constant throughout the sub-domain and in each timestep. It will also be presented the numerical codes which use two other models that more accurately simulate the erosion process, namely the one from Bagnold (1941) and the one from Lettau and Lettau (1978).

The various developments are implemented and tested in ANSYS CFX, a numerical simulation software for fluid flow, by means of FORTRAN user routines.

Keywords ANSYS CFX, User Routines, Wind Erosion, Sinusoidal Pile, Granular Material.

Índice

Índice de Figuras	vi
Índice de Tabelas	vii
SIMBOLOGIA E SIGLAS.....	viii
Simbologia.....	viii
Siglas	ix
1. INTRODUÇÃO.....	1
1.1. Erosão	1
1.2. CFD.....	2
1.3. Motivação	5
2. SIMULAÇÃO NUMÉRICA	6
2.1. O software utilizado e assunções efetuadas	6
2.2. Domínio computacional.....	6
2.3. Geração da malha.....	8
2.4. Condições de fronteira	10
2.4.1. <i>Default Domain</i>	10
2.4.2. <i>Inlet</i>	10
2.4.3. <i>Outlet</i>	11
2.4.4. <i>BottomWall</i>	12
2.4.5. <i>Top</i>	13
2.4.6. <i>Symmetry</i>	13
2.4.7. <i>MovingBoundary</i>	14
3. MÉTODOS NUMÉRICOS PARA A DEFORMAÇÃO DA GEOMETRIA	15
3.1. Modelo matemático	15
3.2. User Routines.....	16
3.2.1. Instalação prévia de software	17
3.2.2. Declaração das variáveis de ambiente	17
3.2.3. Criação da User Routine.....	18
3.2.4. Criação da <i>User Function</i>	18
3.2.5. Criação de Expressões	19
3.2.6. Definição do tipo de simulação	20
3.2.7. Definição do processamento.....	21
3.2.8. Definição dos resultados.....	22
4. SUBROTINA DO TIPO “USER CEL FUNCTION”	24
4.1. Aplicação ao caso em estudo	27
5. RESULTADOS	28
5.1. Discussão dos resultados	29
6. CONCLUSÕES	30
REFERÊNCIAS BIBLIOGRÁFICAS	31

ANEXO A – ROTINA FORTRAN – MODELO DE BAGNOLD (1941).....	32
ANEXO B – ROTINA FORTRAN – MODELO DE LETTAU E LETTAU (1978).....	36

ÍNDICE DE FIGURAS

Figura 1.1 Meios de transporte de materiais granulares por acção do vento (Gouveia e Dias, 1995)	2
Figura 1.2 Ciclo de <i>design</i> CFD	4
Figura 2.1 Domínio computacional, para o caso de duas colinas colocadas em tandem	6
Figura 2.2 Vista da malha obtida, utilizando uma Angular Resolution de 30 graus na generalidade do domínio, e 18 graus na zona da colina.....	9
Figura 3.1 Definições da User Routine	18
Figura 3.2 Definições da <i>User Function</i>	19
Figura 3.3 Definições de simulação	21
Figura 3.4 Definições do processador	22
Figura 3.5 Definição dos resultados	22
Figura 3.6 Definição dos resultados transientes	23
Figura 4.1 Ilustração do conceito de compilação de ficheiros no CFX.....	25
Figura 4.2 Ilustração do conceito de carregamento dinâmico da biblioteca partilhada por parte do CFX	25
Figura 4.3 Estrutura básica de uma função CEL	26
Figura 4.4 Ilustração da camada de células correspondentes à fronteira <i>MovingBoundary</i>	27
Figura 4.5 Esquema simplificado dos argumentos necessários ao modelo, e do resultado produzido.....	27
Figura 5.1 Deformação do perfil longitudinal da colina recorrendo à utilização de uma rotina FORTRAN.....	28

ÍNDICE DE TABELAS

Tabela 2.1 Definições da <i>Inflation Layer</i>	8
Tabela 2.2 Definições do <i>Face Spacing</i>	9
Tabela 2.3 Definições aplicadas nas fronteiras do domínio global	10
Tabela 2.4 Definições aplicadas como condições de fronteira à entrada (<i>Inlet</i>)	11
Tabela 2.5 Definições aplicadas como condições de fronteira à saída (<i>Outlet</i>)	12
Tabela 2.6 Definições aplicadas como condições de fronteira na superfície de Barlavento e Sotavento	12
Tabela 2.7 Definições aplicadas como condições de fronteira na superfície de topo (<i>Top</i>)	13
Tabela 2.8 Definições aplicadas como condições de fronteira nas superfícies laterais (<i>Back e Front</i>)	13
Tabela 2.9 Definições aplicadas como condições de fronteira na Colina	14
Tabela 3.1 Directorias contidas na variável PATH	17
Tabela 3.2 Directorias contidas na variável lib	17
Tabela 3.3 Directorias contidas na variável include	18
Tabela 3.4 Definições aplicadas para a criação da expressão DN1	19
Tabela 3.5 Definições aplicadas para a criação da expressão tStep	19
Tabela 3.6 Definições aplicadas para a criação da expressão tTotal	20
Tabela 3.7 Definições aplicadas para a criação da expressão MeshDisp	20
Tabela 4.1 Lista de argumentos padrão relativa a uma rotina do tipo <i>User CEL Function</i>	26
Tabela 4.2 Lista de argumentos de uma rotina para o caso da figura 4.4	27

SIMBOLOGIA E SIGLAS

Simbologia

ρ_a	– Massa volúmica do ar	[kg m ⁻³]
ρ_s	– Massa volúmica da areia	[kg m ⁻³]
α	– Ângulo de repouso	[rad]
α_i	– Constante adimensional	
δ	– Espessura da camada limite	[m]
τ	– Tensão na parede	[kg m ⁻¹ s ⁻²]
θ	– Ângulo de inclinação da colina	[rad]
ε	– Taxa de dissipação da energia cinética turbulenta	[m ² s ⁻³]
d	– Diâmetro médio do grão de areia	[m]
D	– Diâmetro de referência do grão de areia	[m]
g	– Aceleração da gravidade	[m s ⁻²]
H	– Altura inicial das colinas	[m]
L	– Comprimento das colinas	[m]
N_c	– Número de colinas	
k	– Constante de von Kármán	
K	– Energia cinética turbulenta	[m ² s ⁻²]
t	– Tempo	[s]
u	– Componente longitudinal média da velocidade	[m s ⁻¹]
u_*	– Velocidade de atrito	[m s ⁻¹]
u_{*t}	– Velocidade de atrito crítica	[m s ⁻¹]
$u_{*t,0}$	– Velocidade de atrito crítica para um leito plano horizontal	[m s ⁻¹]
$u_{*t,\theta}$	– Velocidade de atrito crítica para uma vertente de inclinação θ	[m s ⁻¹]
u_t	– Velocidade crítica do fluido	[m s ⁻¹]
U_0	– Componente longitudinal da velocidade do	[m s ⁻¹]

	escoamento não perturbado	
z	Distância (na vertical) ao solo	[m]
u_{exc}	Diferença entre a velocidade de atrito e a velocidade de atrito crítica, num determinado ponto do domínio	[m s ⁻¹]
L_{salt}	Comprimento de saltação	[m]
Q_B	Fluxo mássico total (Bagnold)	[kg m ⁻¹ s ⁻¹]
Q_L	Fluxo mássico total (Lettau e Lettau)	[kg m ⁻¹ s ⁻¹]
Φ_B	Fluxo mássico na direcção vertical (Bagnold)	[kg m ⁻² s ⁻¹]
Φ_L	Fluxo mássico na direcção vertical (Lettau e Lettau)	[kg m ⁻² s ⁻¹]
Δh_B	Altura de material arrancado da colina (Bagnold)	[m]
Δh_L	Altura de material arrancado da colina (Lettau e Lettau)	[m]
Δt	Intervalo de tempo	[s]
A	Constante adimensional	

Siglas

CAD – Desenho assistido por computador

CCL – *CFX Command Language*

CEL – *CFX Expression Language*

CFD - *Computational Fluid Dynamics*

D.C. – Distância entre colinas

DEM – Departamento de Engenharia Mecânica

FCTUC – Faculdade de Ciências e Tecnologia da Universidade de Coimbra

FORTTRAN - *IBM Mathematical Formula Translation System*

PDE – Equação diferencial parcia

1. INTRODUÇÃO

1.1. Erosão

O transporte de partículas sólidas mais ou menos coesas por um escoamento é um fenómeno corrente e que tem na natureza um papel importantíssimo, quer seja nos processos de erosão e sedimentação de solos ou mesmo nos de polinização das plantas. O escoamento pode efetuar-se tanto em meio líquido como gasoso.

O transporte das partículas pode ocorrer através de três mecanismos distintos: arrasto/rolamento, saltação e suspensão. No mecanismo de arrasto/rolamento, as partículas são transportadas sem que estas percam o contacto com a superfície. Tanto na saltação como na suspensão, as partículas são arrancadas da superfície, sendo que na saltação as partículas percorrem uma trajetória semelhante à de um projétil, sendo depositadas a uma distância relativamente curta, quando comparada com a distância percorrida pelas partículas em suspensão, que podem ficar suspensas durante dias e percorrer distâncias que vulgarmente podem atingir, nalguns casos milhares de quilómetros.

Consideremos uma superfície coberta de material granular, sobre a qual incide um escoamento de ar. À medida que a velocidade do escoamento aumenta, as partículas começam a mover-se, pois a força que o vento exerce nelas aumenta e tende a superar o peso dos grãos, começando a arrancar alguns deles da superfície em que se encontram, assim que se supere a velocidade crítica do fluido, u_t , que é a velocidade crítica para a qual se inicia o movimento das partículas. Bagnold (1941) observou, em experiências em túnel de vento, que o início do movimento ocorria predominantemente por **rolamento/deslizamento**. Verificou ainda que, se fossem introduzidas partículas no lado de barlavento do escoamento, estas iriam arrastar outras partículas cujo movimento não tinha sido ainda iniciado. Bisal e Nielson (1962) (citados por Pye e Tsoar, 2009) observaram que, antes de se movimentarem, algumas partículas começavam a vibrar, iniciando depois um movimento por saltos (**saltação**), fazendo um ângulo entre 30 e 90 graus com a horizontal. As partículas de menor dimensão ($d < 0.06\text{mm}$) são geralmente transportadas por **suspensão**, por a componente vertical do seu peso ser inferior à componente vertical do fluxo turbulento que se opõe.

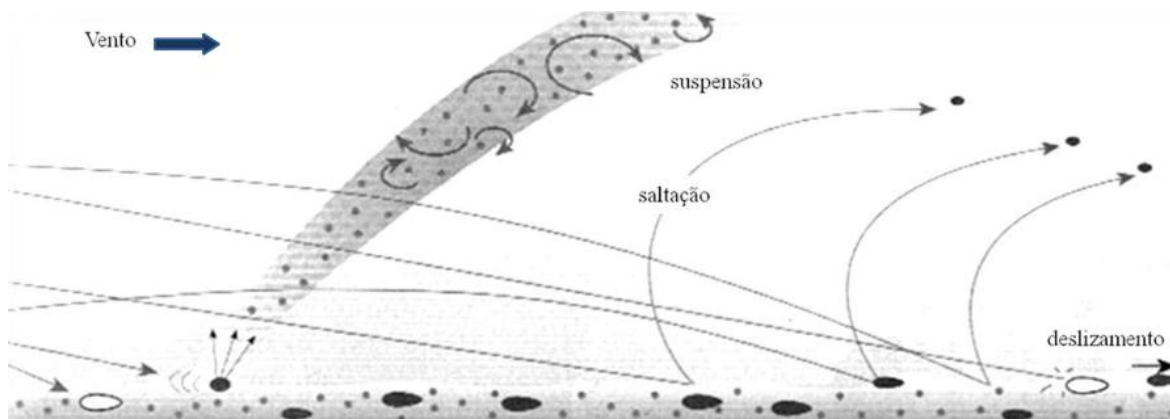


Figura 1.1 Meios de transporte de materiais granulares por acção do vento (Gouveia e Dias, 1995)

1.2. CFD

O escoamento de fluidos é governado por um sistema de equações diferenciais parciais (ou integro-diferenciais), que representam as leis de conservação de massa, de quantidade de movimento (*momentum*) e de energia, bem como de outras quantidades.

CFD (*Computational Fluid Dynamics*), é a arte de substituir tais sistemas de equações diferenciais parciais (PDE) por um conjunto de equações algébricas que possam ser resolvidas com o uso de computadores, utilizando um método de discretização (temporal e espacial). As aproximações são efetuadas em pequenos domínios de espaço e/ou tempo, pelo que as soluções são obtidas em pontos discretos de espaço e/ou tempo. É ainda necessária a especificação das condições de fronteira para a região de interesse em estudo.

Esta análise fornece previsões quantitativas do escoamento de fluidos através de:

- Modelação matemática (PDE's);
- Métodos numéricos (técnicas de discretização e resolução);
- Ferramentas de software (pré-processamento, processamento, pós-processamento).

Comparativamente à análise experimental, a análise computacional (CFD) apresenta as seguintes vantagens:

- Custos – a análise computacional apresenta normalmente custos inferiores aos da experimental, pelo facto de não ser necessário adquirir todo o equipamento necessário à realização da análise experimental. É espectável que os custos inerentes à simulação CFD tendam a diminuir, à medida que a capacidade computacional (velocidade de processamento e capacidade de armazenamento) aumenta;
- Rapidez – torna-se mais fácil alterar as condições do problema, permitindo testar um maior número de configurações relativas ao problema em análise;
- Capacidade em simular condições reais – seja pela dificuldade em testar certas condições (e.g. escoamento hipersónico) ou devido à grande dimensão dos sistemas, a análise computacional permite-nos simular condições que, pela via experimental, seriam difíceis de criar, se não mesmo impossíveis;
- Capacidade em simular condições ideais – devido ao grande controlo existente, na maioria dos casos, sobre o processo físico, torna-se possível isolar fenómenos específicos para estudo;
- Informação detalhada – mesmo os ensaios experimentais mais completos apenas permitem obter medições num número limitado de pontos do domínio. A análise computacional permite-nos efectuar previsões dos parâmetros de escoamento pretendidos num conjunto de pontos do domínio mais alargado.

Mas, neste tipo de análise, nem tudo são vantagens. O conhecimento das limitações da análise CFD reveste-se da maior importância. De entre as principais limitações deste método, destacam-se:

- Modelos físicos – as soluções obtidas pela via CFD são baseadas em modelos físicos de processos/fenómenos do mundo real. Como tal, a precisão dos resultados poderá ser apenas tão elevada quanto a do modelo físico do qual está dependente ;
- Erros numéricos – introduzidos pela resolução computacional das equações. O rigor dos resultados é limitado pela capacidade computacional, bem como pelas aproximações efectuadas nos modelos

numéricos. Estes últimos serão tão mais pequenos quanto mais refinada for a malha;

- Condições de fronteira – assim como no modelo físico, a qualidade da solução está fortemente dependente da qualidade com que as condições de fronteira reais do problema em estudo são reproduzidas numericamente.

É de referir ainda que a fiabilidade das simulações CFD é maior:

- para escoamentos laminares que para escoamentos turbulentos;
- para escoamentos monofásicos que para escoamentos multifásicos;
- para escoamentos quimicamente inertes que para escoamentos reactivos.

É tempo agora de explicar como são feitas as simulações de CFD. Os principais componentes de um ciclo de projecto são indicados na Figura 1.2 :

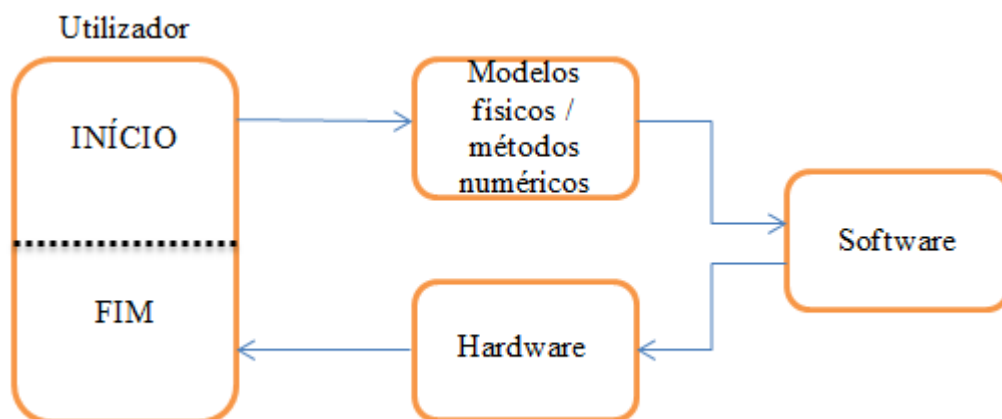


Figura 1.2 Ciclo de *design* CFD

Note-se que é o utilizador quem inicia e finaliza o ciclo. No início é ele quem define o problema em análise. Na fase seguinte, o conhecimento científico acerca dos processos físicos que governam a evolução do problema permite desenvolver os modelos físico-matemáticos que se considere serem os mais adequados para simular as condições do problema real, com recurso a ferramentas numéricas. O software incorpora depois tudo

isto para dar instruções específicas (algoritmos) ao computador, para que o hardware efectue os cálculos necessários. No final desta cadeia, o utilizador procede à interpretação e verificação dos resultados, normalmente com recurso a ferramentas de pós-processamento de dados. Caso os resultados obtidos não sejam satisfatórios, é o utilizador quem deve analisar quais as alterações necessárias a introduzir para melhorar a qualidade dos resultados.

1.3. Motivação

A motivação para a investigação levada a cabo acerca da erosão eólica decorreu da necessidade de prever e controlar os efeitos nefastos deste processo, nomeadamente a suspensão de partículas poluentes no ar, a destruição de plantações agrícolas (que têm um papel crucial no fornecimento de alimentos às populações) ou ainda a progressão das zonas áridas e a desertificação.

Compreender os mecanismos que governam este fenómeno torna-se crucial para conseguirmos prever o seu desenvolvimento e controlar os seus efeitos.

A análise computacional tem o intuito de complementar a análise experimental levada a cabo por Ferreira e Fino (2011). Pretende-se desenvolver uma metodologia que permita testar diferentes modelos de erosão, através da programação de rotinas, a integrar no software comercial CFX, e que permitam o cálculo da erosão de um leito granular e consequente modelação da deformação da superfície ao longo do tempo.

2. SIMULAÇÃO NUMÉRICA

2.1. O software utilizado e assunções efetuadas

Para efetuar as simulações, o código comercial de CFD escolhido foi o CFX (ANSYS, 2011). Este código resolve as equações Reynolds-averaged Navier-Stokes usando um método de discretização de volumes finitos. Como o objetivo principal é o de determinar a deformação do perfil da colina ao longo do tempo, foi assumido um regime transiente. Para o presente estudo, resolvem-se as equações de conservação da quantidade de movimento e de continuidade. O código usa uma formulação do tipo volume de controlo e o acoplamento entre pressão e velocidade é obtido usando o algoritmo SIMPLEC (Doormaal & Raithby, 1984). A intensidade de turbulência foi considerada elevada (10%) na região de entrada, sendo simulada utilizando o modelo $K-\varepsilon$, que efetua o cálculo da energia cinética turbulenta, K , e a sua taxa de dissipação, ε (Launder & Spalding, 1974). A convergência numérica assume-se como obtida assim que todos os resíduos forem menores que 10^{-6} .

2.2. Domínio computacional

A geometria utilizada na realização deste trabalho foi semelhante à utilizada por Fino (2011).

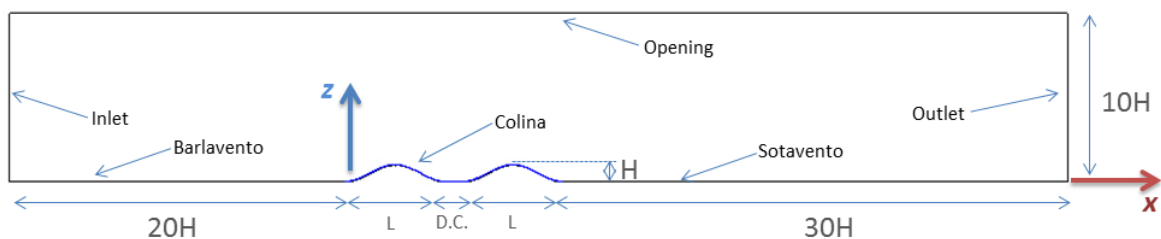


Figura 2.1 Domínio computacional, para o caso de duas colinas colocadas em tandem

O domínio computacional apresentado na Figura 2.1 pode ser subdividido em três zonas: a zona de barlavento, a zona das colinas e a zona de sotavento.

Nas zonas de barlavento e sotavento, as dimensões não variam, sendo de 20H a barlavento e de 30H a sotavento, onde H é a altura inicial da colina. A altura do domínio também não varia e tem o valor de 10H. Na zona das colinas, embora se tenha optado por apresentar o domínio computacional que apresenta uma maior extensão segundo a direção x, é de referir que existem várias configurações possíveis. Os parâmetros relativos a esta zona do domínio são: o número de colinas, N_c ; a altura máxima das colinas, H, dimensão fundamental do domínio, a qual se optou por manter constante; a distância entre colinas, D.C., que representa a distância entre o ponto mais a sotavento da primeira colina e o ponto mais a barlavento da colina de jusante; o comprimento da colina, L, que representa a distância entre o ponto mais a barlavento e o ponto mais a sotavento de uma mesma colina.

Considera-se que a altura máxima das colinas será de 60 mm, podendo o comprimento de cada colina ser de $L=5H$ ou de $L=6H$. No que respeita ao número de colinas, pode existir uma ou então duas colinas. Neste último caso, a distância entre elas poderá ser igual a zero ou igual a H. O perfil sinusoidal da colina é definido pela equação 2.1

$$z = \frac{H}{2} \left(1 + \sin \left[\pi \left(\frac{x}{L/2} + \frac{1}{2} \right) \right] \right) \quad (2.1)$$

Em que z é a distância ao solo, medida na vertical.

Por uma questão de simplicidade, começou por se analisar a configuração em que existia somente uma colina de altura máxima $H=60\text{mm}$ e comprimento $L=6H$. O domínio foi criado com recurso ao software comercial de CAD, Solidworks 2010. Para tal, procedeu-se à criação do perfil da colina através de um conjunto de pontos calculados usando o Microsoft Excel e guardados num ficheiro de texto (*.txt). Em seguida, no Solidworks, procedeu-se ao desenho do domínio destinado ao fluido. Para tal, foi necessário efectuar os seguintes passos:

1. Duplo clique no ficheiro executável do Solidworks;
2. Selecionar **File > New**;
3. Selecionar **Part** e clicar em **OK**;
4. Selecionar **Insert > Curve > Curve through XYZ points**;
5. Clicar em **Browse**;
6. Definir o tipo de ficheiro a importar como sendo do tipo ***.txt**, seleccionar o ficheiro e clicar em **Abrir**;

7. Criar um *sketch*, clicando em **Sketch** na barra de ferramentas;
8. Clicar em **Convert Entities**;
9. Clicar na curva e, em seguida, clicar **OK**;
10. Desenhar os restantes contornos do domínio respeitante ao fluido;
11. Fazer um **Extrude** do sketch, com uma profundidade de 5mm;
12. Guardar a peça num ficheiro de formato *.sldprt ou *.igs.

2.3. Geração da malha

A geração da malha é um processo relativamente rápido, devido ao facto de existirem algumas opções para a geração de malha de forma automática usando o CFX-Mesh. No entanto é necessário efetuar alguns ajustes na definição do processo de geração da malha.

Tendo em conta os elevados gradientes de velocidade esperados junto à superfície, foi aplicada uma *Inflation Layer* nesta superfície. Para tal, na *Tree View*, clicar com o botão direito do rato em **Inflation**, e seleccionar **Insert > Inflated Boundary**. Seleccionar as superfícies aonde se pretende aplicar a *Inflation Layer*, nomeadamente as superfícies de nome Barlavento, Colina e Sotavento (ver figura 2.1), e em seguida clicar em OK. As definições usadas constam na Tabela 2.1 :

Tabela 2.1 Definições da *Inflation Layer*

<i>Inflation</i>		<i>Inflation Option</i>	
<i>Number of Inflated Layers</i>	10	<i>Option</i>	<i>First Layer Thickness</i>
<i>Expansion Factor</i>	1.2	<i>Define First Layer By</i>	<i>First Prism Height</i>
<i>Number of Spreading Iterations</i>	0	<i>First Prism Height [mm]</i>	1
<i>Minimum Internal Angle [Degrees]</i>	2.5	<i>Extended Layer Growth</i>	Yes
<i>Minimum External Angle [Degrees]</i>	5	<i>Layer by Layer Smoothing</i>	No

De forma a evitar que na zona adjacente à *Inflation Layer* fossem gerados elementos demasiadamente refinados, que conduziriam a uma geração de malha e

posteriores simulações desnecessariamente mais lentas, foi aplicado um Default Face Spacing em todo o domínio e um outro apenas para a zona da colina. As definições usadas constam da Tabela 2.2 :

Tabela 2.2 Definições do Face Spacing

Default Face Spacing	
<i>Option</i>	Angular Resolution
<i>Angular Resolution [Degrees]</i>	30
<i>Minimum Edge Length [mm]</i>	15
<i>Maximum Edge Length [mm]</i>	30
<i>Radius of Influence [mm]</i>	0
<i>Expansion Factor</i>	1.2
Face Spacing 1	
<i>Option</i>	Angular Resolution
<i>Angular Resolution [Degrees]</i>	18
<i>Minimum Edge Length [mm]</i>	10
<i>Maximum Edge Length [mm]</i>	20
<i>Radius of Influence [mm]</i>	0
<i>Expansion Factor</i>	1.2
<i>Location</i>	1 2D Region (colina)

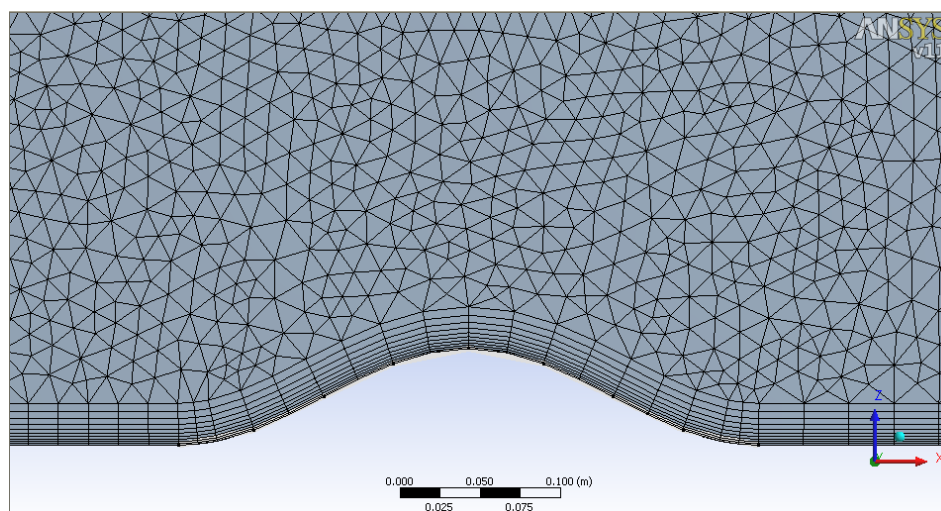


Figura 2.2 Vista da malha obtida, utilizando uma Angular Resolution de 30 graus na generalidade do domínio, e 18 graus na zona da colina

2.4. Condições de fronteira

Nas sub-seções que se seguem, são indicadas as condições de fronteira necessárias, especificando a condição em cada uma.

A especificação das condições de fronteira é feita na plataforma CFX-Pre.

2.4.1. Default Domain

Sendo esta fronteira automaticamente criada, interessa nela definir algumas propriedades, que são indicadas na tabela 2.3 :

Tabela 2.3 Definições aplicadas nas fronteiras do domínio global

Barra	Definição	Valor
<i>General</i>	<i>Basic Settings > Fluids List</i>	<i>Air at 25 C</i>
<i>Options</i>	<i>Domain Models > Reference Pressure</i>	<i>1 [atm]</i>
	<i>Mesh Deformation > Option</i>	<i>Regions of Motion Specified</i>
<i>Fluid Models</i>	<i>Heat Transfer > Option</i>	<i>Isothermal</i>
	<i>Heat Transfer > Fluid Temperature</i>	<i>25 [C]</i>
	<i>Turbulence > Option</i>	<i>k-Epsilon</i>
	<i>Turbulence > Wall Function</i>	<i>Scalable</i>

2.4.2. Inlet

A condição de fronteira na entrada (*Inlet*) foi o perfil de velocidade que obedece à equação 2.2 :

$$\frac{u}{U_0} = \left(\frac{z}{\delta}\right)^\alpha \quad (2.2)$$

sendo u a componente média longitudinal da velocidade, U_0 a velocidade do escoamento não perturbado, z a distancia vertical ao solo, e $\alpha=0,11$. A espessura da camada limite δ é de 0,1m, a 2 metros de distância do início da câmara de ensaios. A intensidade de turbulência da componente longitudinal da velocidade pode considerar-se aproximadamente igual a 10%. Os valores de $u(z)$ foram calculados com recurso a uma folha de cálculo e posteriormente guardados em ficheiro de texto.

Uma vez no CFX-Pre, para criar o perfil de velocidades teve de se criar uma *User Function*. Para tal, no *Outline*, usando o botão direito do rato, clicar em *User*

Function > Insert > User Function. Em seguida, definir o nome da função. Para o caso do perfil de velocidades em que a velocidade do escoamento não perturbado é de 9,1 m/s, o nome atribuído foi U91. Nos detalhes da função, definir **Option** como **Interpolation (Data Input)**, **Argument Units** como [m], **Result Units** como [m s⁻¹] e **Interpolation Data > Option** como **One Dimensional**. Em seguida, clicar com o botão direito do rato na tabela destinada aos valores da função e escolher **Import**. Selecionar o ficheiro onde estão guardados os pontos relativos ao perfil de velocidades e clicar em **Open**. Para finalizar, clicar em **Apply** e, em seguida, **Ok**.

Para proceder à criação da condição de fronteira, clicar **Insert > Boundary Condition**. Definir o nome como **Inlet**. Em seguida, duplo clique em **Inlet** para configurar as suas definições. As seguintes definições foram aplicadas, conforme é apresentado na tabela 2.4 :

Tabela 2.4 Definições aplicadas como condições de fronteira à entrada (**Inlet**)

Barra	Definição	Valor
<i>Basic Settings</i>	<i>Boundary Type</i>	<i>Inlet</i>
	<i>Location</i>	<i>Inlet</i>
<i>Boundary Details</i>	<i>Flow Regime > Option</i>	<i>Subsonic</i>
	<i>Mass And Momentum > Option</i>	<i>Cart. Vel. Components</i>
	<i>Mass And Momentum > U</i>	<i>U91(z)</i>
	<i>Mass And Momentum > V</i>	<i>0 [m s⁻¹]</i>
	<i>Mass And Momentum > W</i>	<i>0 [m s⁻¹]</i>
<i>Boundary Details</i>	<i>Turbulence > Option</i>	<i>High (Intensity = 10%)</i>
	<i>Mesh Motion > Option</i>	<i>Stationary</i>

Para aplicar, clicar em **Apply** e, em seguida, **Ok**

2.4.3. Outlet

Clicar **Insert > Boundary Condition**. Definir o nome como **Outlet**. Em seguida, duplo clique em **Outlet** para configurar as suas definições. As seguintes definições foram aplicadas, conforme é apresentado na tabela 2.5 :

Tabela 2.5 Definições aplicadas como condições de fronteira à saída (*Outlet*)

Barra	Definição	Valor
<i>Basic Settings</i>	<i>Boundary Type</i>	<i>Outlet</i>
	<i>Location</i>	<i>Outlet</i>
<i>Boundary Details</i>	<i>Flow Regime > Option</i>	<i>Subsonic</i>
	<i>Mass And Momentum > Option</i>	<i>Static Pressure</i>
	<i>Mass And Momentum > Relative Pressure</i>	<i>0 [Pa]</i>
	<i>Mesh Motion > Option</i>	<i>Stationary</i>

Para aplicar, clicar em **Apply** e, em seguida, **Ok**.

2.4.4. BottomWall

Clicar **Insert > Boundary Condition**. Definir o nome como *BottomWall*. Em seguida, duplo clique em *BottomWall* para configurar as suas definições. As seguintes definições foram aplicadas, conforme é apresentado na tabela 2.6 :

Tabela 2.6 Definições aplicadas como condições de fronteira na superfície de Barlavento e Sotavento

Barra	Definição	Valor
<i>Basic Settings</i>	<i>Boundary Type</i>	<i>Wall</i>
	<i>Location</i>	<i>Barlavento, Sotavento</i>
<i>Boundary Details</i>	<i>Flow Regime > Option</i>	<i>Subsonic</i>
	<i>Wall Influence on Flow > Option</i>	<i>No Slip</i>
	<i>Wall Roughness > Option</i>	<i>Rough Wall</i>
<i>Boundary Details</i>	<i>Wall Roughness > Roughness Height</i>	<i>5e-005 [m]</i>
	<i>Mesh Motion > Option</i>	<i>Stationary</i>

Para aplicar, clicar em **Apply** e, em seguida, **Ok**.

2.4.5. Top

Clicar *Insert > Boundary Condition*. Definir o nome como *Top*. Em seguida, duplo clique em *Top* para configurar as suas definições. As seguintes definições foram aplicadas, conforme é apresentado na tabela 2.7 :

Tabela 2.7 Definições aplicadas como condições de fronteira na superfície de topo (*Top*)

Barra	Definição	Valor
<i>Basic Settings</i>	<i>Boundary Type</i>	<i>Opening</i>
	<i>Location</i>	<i>Opening</i>
<i>Boundary Details</i>	<i>Flow Regime > Option</i>	<i>Subsonic</i>
	<i>Mass And Momentum > Option</i>	<i>Static Pres. (Entrain)</i>
	<i>Mass And Momentum > Relative Pressure</i>	<i>0 [Pa]</i>
	<i>Turbulence > Option</i>	<i>High (Intensity = 10%)</i>
	<i>Mesh Motion > Option</i>	<i>Stationary</i>

Para aplicar, clicar em *Apply* e, em seguida, **Ok**.

2.4.6. Symmetry

Clicar *Insert > Boundary Condition*. Definir o nome como *Symmetry*. Em seguida, duplo clique em *Symmetry* para configurar as suas definições. As seguintes definições foram aplicadas, conforme é apresentado na tabela 2.8 :

Tabela 2.8 Definições aplicadas como condições de fronteira nas superfícies laterais (*Back e Front*)

Barra	Definição	Valor
<i>Basic Settings</i>	<i>Boundary Type</i>	<i>Symmetry</i>
	<i>Location</i>	<i>Back, Front</i>
<i>Boundary Details</i>	<i>Mesh Motion > Option</i>	<i>Unspecified</i>

Para aplicar, clicar em *Apply* e, em seguida, **Ok**.

O facto de não se especificar o movimento da malha nesta fronteira deve-se ao facto de esta superfície ter de acompanhar o movimento da fronteira “MovingBoundary”.

2.4.7. *MovingBoundary*

Para proceder à criação da condição de fronteira, clicar **Insert > Boundary Condition**. Definir o nome como *MovingBoundary*. Em seguida, duplo clique em *MovingBoundary* para configurar as suas definições. As seguintes definições foram aplicadas, conforme é apresentado na tabela 2.9 :

Tabela 2.9 Definições aplicadas como condições de fronteira na Colina

Barra	Definição	Valor
<i>Basic Settings</i>	<i>Boundary Type</i>	<i>Wall</i>
	<i>Location</i>	<i>Colina</i>
<i>Boundary Details</i>	<i>Wall Influence on Flow > Option</i>	<i>No Slip</i>
	<i>Wall Roughness > Option</i>	<i>Rough Wall</i>
	<i>Wall Roughness > Roughness Height</i>	<i>5e-005 [m]</i>
	<i>Mesh Motion > Option</i>	<i>Specified Displacement</i>
	<i>Mesh Motion > X Component</i>	<i>0 [m]</i>
	<i>Mesh Motion > Y Component</i>	<i>0 [m]</i>
	<i>Mesh Motion > Z Component</i>	<i>MeshDisp</i>

Para aplicar, clicar em **Apply** e, em seguida, **Ok**.

Note-se que, até à criação desta fronteira, a função *MeshDisp* não se encontra ainda criada, razão pela qual aparecem alguns conflitos na janela. Esses conflitos desaparecerão assim que for criada a função *MeshDisp*.

3. MÉTODOS NUMÉRICOS PARA A DEFORMAÇÃO DA GEOMETRIA

3.1. Modelo matemático

A velocidade de atrito crítica $u_{*t,0}$ [m s^{-1}] para um material granular depositado num leito plano pode ser determinada pela equação 3.1 (Bagnold, 1941).

$$u_{*t,0} = A \sqrt{\frac{(\rho_s - \rho_a)}{\rho_a} * g * d} \quad (3.1)$$

em que A toma o valor de 0,1, ρ_s é a densidade do material granular, ρ_a a densidade do fluido, g a aceleração da gravidade e d o diâmetro médio dos grãos.

Iversen & Rasmussen (1994) propuseram a equação 3.2, que permite determinar a velocidade de atrito crítica $u_{*t,\theta}$ [m s^{-1}], para que partículas depositadas numa vertente de ângulo θ iniciem o movimento.

$$u_{*t,\theta} = \sqrt{u_{*t,0}^2 \left(\cos \theta + \frac{\sin \theta}{\tan \alpha} \right)} \quad (3.2)$$

em que α é o ângulo de fricção interna do material granular, que pode ser considerado igual ao ângulo de repouso (≈ 32 graus).

Por uma questão de conveniência, interessa agora definir o excesso da velocidade de atrito $u_{*t,0}$ relativamente ao seu valor crítico $u_{*t,\theta}$, através da equação 3.3.

$$u_{exc} = (u_* - u_{*t,\theta}) \quad (3.3)$$

O comprimento de saltação, L_{salt} [m], que é a distância percorrida pelas partículas durante a saltação é, segundo de Almeida et al., 2008, para um escoamento de ar na Terra, a uma temperatura de 25 [°C], dado por:

$$L_{salt} = 4,549 * (u_* - u_{*t}) \quad (3.4)$$

As equações que modelam o fluxo mássico total, Q_B e Q_L , das partículas arrancadas da superfície da colina por ação de um escoamento com velocidade u_* são, segundo Bagnold (1941) (equação 3.5)

$$Q_B = c_B \sqrt{\frac{d \rho}{D g}} u_*^3 \left[\frac{kg}{m s} \right] \quad (3.5)$$

e, segundo Lettau e Lettau (1978) (equação 3.6)

$$Q_L = c_L \sqrt{\frac{d \rho}{D g}} u_*^2 (u_* - u_{*t}) \left[\frac{kg}{m s} \right] \quad (3.6)$$

sendo c_B e c_L constantes adimensionais de valores iguais a 1,8 e 6,7 (Sherman e Li, 2012), respetivamente. d é o diâmetro médio de um grão de areia e D o diâmetro de grão de areia de referência ($D=250\mu m$), u_{*t} é a velocidade crítica de arranque de partículas.

Desta forma, o fluxo vertical de partículas é expresso pela equação 3.7 .

$$\Phi = \frac{Q}{L_{sat}} \left[\frac{kg}{m^2 \cdot s} \right] \quad (3.7)$$

Ao serem arrancadas partículas da superfície da colina, essa mesma superfície vai baixar uma distância Δh , em metros, definida através da equação 3.8 como sendo:

$$\Delta h = \frac{\Phi}{\rho_a} \cdot \Delta t \quad [m] \quad (3.8)$$

3.2. User Routines

User Routines são rotinas programadas pelo utilizador e que podem ser carregadas de forma dinâmica pelo processador do ANSYS CFX., para aumentar as suas funcionalidades. São programadas em linguagem FORTRAN. São escritas em *source-code*, através de qualquer editor de texto que suporte o formato FORTRAN (.for). Os parâmetros de entrada (“variáveis” de entrada) são carregados a partir do ANSYS e os parâmetros de saída (“variáveis” de saída) são exportados para o ANSYS, para que possam ser utilizados por outros processadores (pós-processamento, por exemplo).

3.2.1. Instalação prévia de *software*

Antes de começar a trabalhar com *User Routines* é necessário instalar os pacotes que contêm o seguinte *software*:

- *Microsoft Visual Studio 9.0*;
- *Microsoft Windows SDK*;
- *Intel Fortran Compiler 11.0.061*.

3.2.2. Declaração das variáveis de ambiente

Após ter instalado correctamente o *software* mencionado atrás, dever-se-há proceder à declaração das variáveis de ambiente do *Windows*. Para tal é necessário aceder a **Computador > Propriedades do Sistema > Definições Avançadas do Sistema > Variáveis de Ambiente**.

Nas variáveis que são apresentadas nas tabelas 3.1 a 3.3, incluir as directorias indicadas:

Tabela 3.1 Directorias contidas na variável PATH

PATH	C:\Programas\Intel\Compiler\11.0\061\fortran\Bin\IA32 C:\Programas\Microsoft Visual Studio 9.0\Common7\IDE C:\Programas\Microsoft Visual Studio 9.0\VC\bin C:\Programas\Microsoft Visual Studio 9.0\Common7\Tools C:\Programas\Microsoft Visual Studio 9.0\Common7\Tools\bin C:\Programas\Microsoft SDKs\Windows\v6.0A\bin C:\Windows\System32 C:\Programas\Intel\Compiler\11.0\061\fortran\lib\ia32 C:\Programas\Intel\Compiler\11.0\061\fortran\mkl\ia32\bin
------	--

Tabela 3.2 Directorias contidas na variável lib

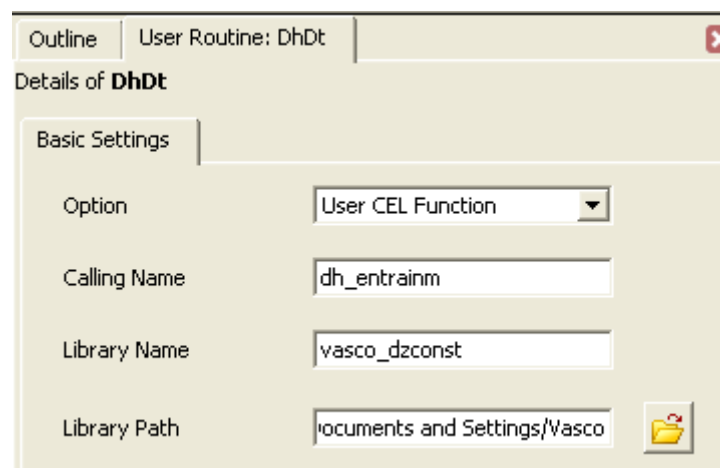
lib	C:\Programas\Intel\Compiler\11.0\061\fortran\lib\ia32 C:\Programas\Microsoft Visual Studio 9.0\VC\atlmfc\lib C:\Programas\Microsoft Visual Studio 9.0\VC\lib C:\Programas\Microsoft SDKs\Windows\v5.0A\Lib C:\Programas\ANSYS Inc\v110\ANSYS\custom\lib\intel C:\Programas\Intel\Compiler\11.0\061\fortran\mkl\ia32\lib
-----	--

Tabela 3.3 Directorias contidas na variável include

include	C:\Programas\Intel\Compiler\11.0\061\fortran\include C:\Programas\Intel\Compiler\11.0\061\fortran\include\ia32 C:\Programas\Microsoft Visual Studio 9.0\VC\include C:\Programas\Microsoft SDKs\Windows\v5.0A\include C:\Programas\ANSYS Inc\v110\CFX\include C:\Programas\Intel\Compiler\11.0\061\fortran\mkl\include
---------	--

3.2.3. Criação da *User Routine*

Uma vez criada a rotina para incorporar no ANSYS, é necessário articulá-la com o CFX. Estando no CFX-Pre, será necessário clicar com o botão direito do rato em *User Routines > Insert > User Routine*, e, em seguida preencher os campos da forma que se apresenta na figura 3.1 :

**Figura 3.1** Definições da *User Routine*

3.2.4. Criação da *User Function*

Criar uma *User Function* para importar os valores de uma rotina ou de um ficheiro de texto, definindo as unidades tanto dos argumentos como dos resultados.

3.2.4.1. *MeshMotion*

A função *MeshMotion*, cujos detalhes são apresentados na figura 3.2, é criada para articular a rotina FORTRAN externa com o processador do CFX. Esta função serve de intermediária entre a rotina FORTRAN e a expressão *MeshDisp*, que será exposta no sub-sub-capítulo seguinte.

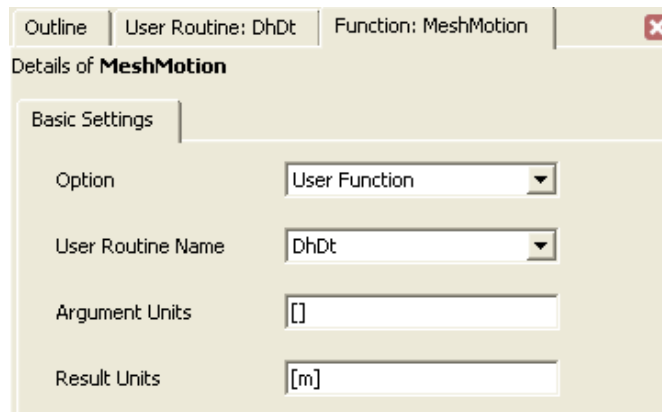


Figura 3.2 Definições da *User Function*

3.2.5. Criação de Expressões

A criação da expressão prende-se com o facto de utilizar os valores da *User Function* nos campos das definições do movimento da fronteira, assim como o de definir alguns parâmetros. As expressões criadas são indicadas a seguir. São criadas clicando com o botão direito do rato em *Expressions*, seleccionando depois *Insert > Expression*, definindo depois o nome a atribuir à expressão, seguido da própria expressão.

3.2.5.1. DN1

A expressão DN1 foi criada, utilizando as definições apresentadas na tabela 3.4, com o intuito de indicar o diâmetro médio das partículas, d .

Tabela 3.4 Definições aplicadas para a criação da expressão DN1

Nome	Expressão
DN1	0.0005 [m]

3.2.5.2. tStep

A expressão tStep, cujas definições são apresentadas na tabela 3.5, indica o intervalo de tempo entre iterações, Δt .

Tabela 3.5 Definições aplicadas para a criação da expressão tStep

Nome	Expressão
tStep	1. [s]

3.2.5.3. t_{Total}

A expressão t_{Total} indica o tempo total de simulação. As suas definições são apresentadas na tabela 3.6.

Tabela 3.6 Definições aplicadas para a criação da expressão t_{Total}

Nome	Expressão
t_{Total}	10. [s]

3.2.5.4. $MeshDisp$

A expressão $MeshDisp$, cujas definições são apresentadas na tabela 3.7, indica o deslocamento da malha recorrendo à função $MeshMotion$, assim como a outros parâmetros da simulação, sendo “*Shear Velocity*” e “*meshdisptot z*” obtidos através da simulação, e as restantes definidas pelo utilizador.

Esta expressão indica que o deslocamento da malha na direção vertical, para um determinado intervalo de tempo, será uma função da velocidade de atrito, diâmetro médio das partículas e do intervalo de tempo em questão, assim como do deslocamento acumulado durante os intervalos de tempo anteriores.

Tabela 3.7 Definições aplicadas para a criação da expressão $MeshDisp$

Nome	Expressão
$MeshDisp$	$MeshMotion(Shear\ Velocity, tStep, DNI) + meshdisptot\ z$

3.2.6. Definição do tipo de simulação

No que respeita aos detalhes da simulação, interessa referir que este é um processo transiente, na medida em que as condições, tanto da colina como do escoamento, se alteram ao longo do tempo. Ao ser alterada, decorrente da erosão, a geometria do domínio, os campos vectoriais da velocidade, bem como da pressão, por exemplo, também variam. A variação do campo de velocidades está diretamente relacionado com a variação da taxa de erosão da colina.

A simulação tem início no instante de tempo $t=0$ e decorre enquanto o tempo virtual da simulação não igualar ou exceder o tempo definido anteriormente como sendo o tempo total da simulação, t_{Total} , com incrementos de tempo de valor $tStep$ entre iterações.

Os seus detalhes são apresentados na figura 3.3

Details of **Simulation Type**

Basic Settings

External Solver Coupling

Option: None

Simulation Type

Option: Transient

Time Duration

Option: Total Time

Total Time: tTotal

Time Steps

Option: Timesteps

Timesteps: tStep

Initial Time

Option: Automatic with Value

Time: 0 [s]

Figura 3.3 Definições de simulação

3.2.7. Definição do processamento

Para que o processamento produza resultados satisfatórios, será necessário definir alguns parâmetros. A figura 3.4 mostra as opções que é necessário definir. O esquema de advecção utilizado foi definido como “*High Resolution*” e o esquema de análise transiente como “*Second Order Backward Euler*”. É neste passo que se define o critério de convergência introduzido anteriormente, o qual se optou por fixar o resíduo máximo de cada uma das equações no valor de $1E-6$.

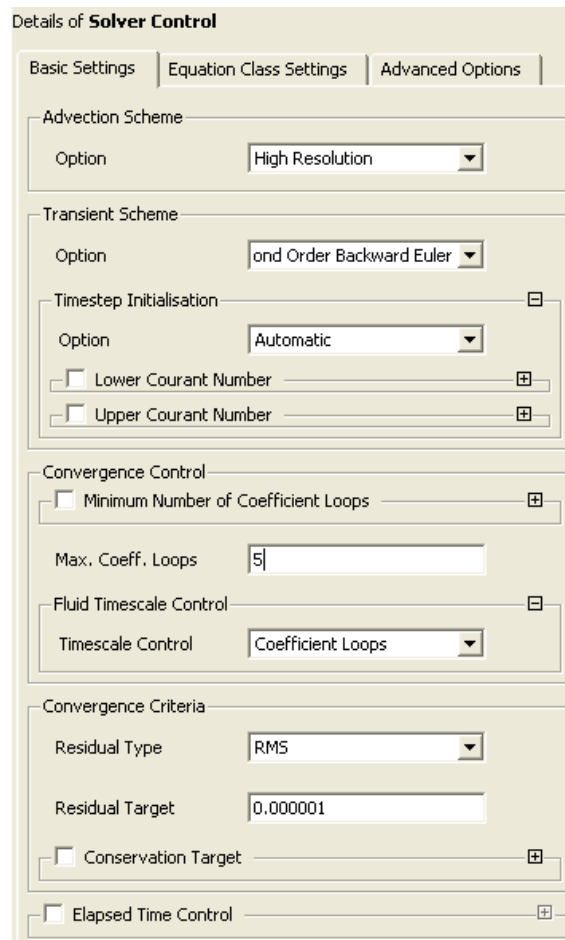


Figura 3.4 Definições do processador

3.2.8. Definição dos resultados

No que respeita à obtenção de resultados, optou-se por listar as variáveis relativas à componente longitudinal da velocidade do escoamento, bem como o deslocamento total da malha segundo a direcção vertical, conforme se mostra na figura 3.5.

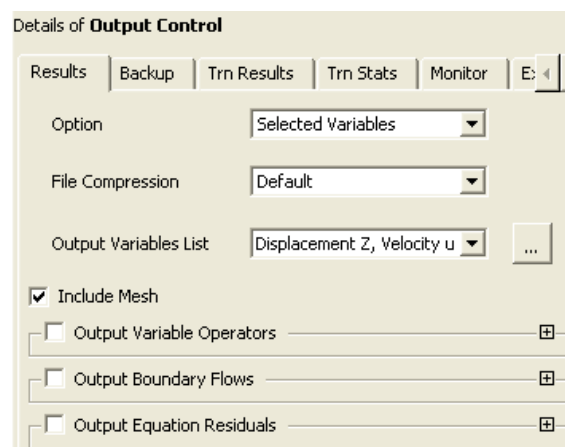


Figura 3.5 Definição dos resultados

3.2.8.1. Resultados transientes

Os resultados transientes mostram as mesmas variáveis que os resultados finais, sendo que um novo ficheiro de resultados transientes (*.trn) é gerado em cada intervalo de tempo. A figura 3.6 mostra as definições aplicadas neste campo.

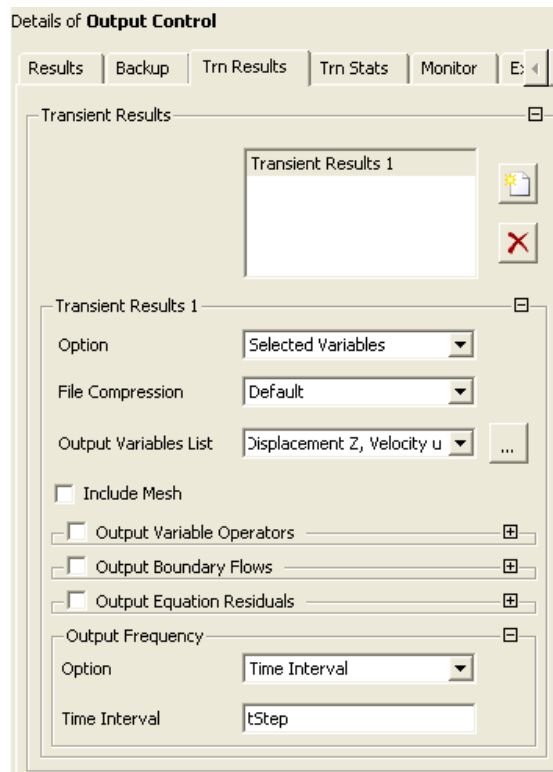


Figura 3.6 Definição dos resultados transientes

4. SUBROTINA DO TIPO “USER CEL FUNCTION”

As rotinas FORTRAN possuem um conjunto variado de aplicações. Exemplos de onde estas rotinas podem ser utilizadas incluem: *input* de dados externos para o CFX, definição das condições de fronteira, definição de termos-fonte, controlo avançado do processamento, assim como, no presente trabalho, de definição do movimento relativo das fronteiras. As rotinas FORTRAN são utilizadas quando as ferramentas *standard* do CFX tornam difícil, quer por terem capacidades limitadas ou pelo facto de a lógica inerente ao processo não estar contemplada no CFX, a modelação dos processos físicos de interesse. As suas capacidades excedem o que poderia ser atingido com recurso ao CEL ou ao CCL.

Existem dois tipos principais de rotinas, que podem ser utilizadas no âmbito deste trabalho, sendo elas: “*User CEL Function*” e a “*Junction Box Routine*”. As primeiras são utilizadas para escrever funções CEL próprias, enquanto que as segundas são utilizadas quando se pretende que estas sejam chamadas em pontos específicos do processamento. Ambas são programadas em linguagem FORTRAN, e têm acesso a todas as estruturas de dados do processador do CFX. Estas subrotinas podem chamar outras subrotinas definidas pelo utilizador, bem como rotinas ou funções do CFX. Quando uma rotina tem necessidade de chamar outra rotina, diz-se que esta é uma “rotina de alto nível”, sendo que, quando essa necessidade não existe, a rotina pode ser designada de “rotina de baixo nível).

Uma vez definidas e preparadas as rotinas FORTRAN, será necessário compilar os respectivos ficheiros. O CFX não tem a capacidade de reconhecer o código-fonte (em linguagem FORTRAN). O que realmente se passa é que o CFX, à medida que o processamento se vai desenrolando, interpreta dinamicamente a biblioteca criada através da compilação, e que por sua vez será partilhada com o CFX, ou seja, sempre que a expressão que utiliza a função, que por sua vez contém a rotina FORTRAN, seja chamada. Assim sendo, é necessário compilar os ficheiros de código-fonte, por forma a convertê-los para uma linguagem que o CFX consiga interpretar. Para tal, é necessário correr o utilitário “*cfx5mkext*” através da introdução do seguinte comando no editor de comandos do CFX-Pre: “! System (“*cfx5mkext* ficheiro1.F ficheiro2.F ...”)==0 or die;”; sendo esta uma

introdução em linguagem Perl que levará o CFX a executar o utilitário `cfx5mkext` nos ficheiros FORTRAN indicados, criando então uma biblioteca partilhada com o processador do CFX (Figuras 4.1 e 4.2).

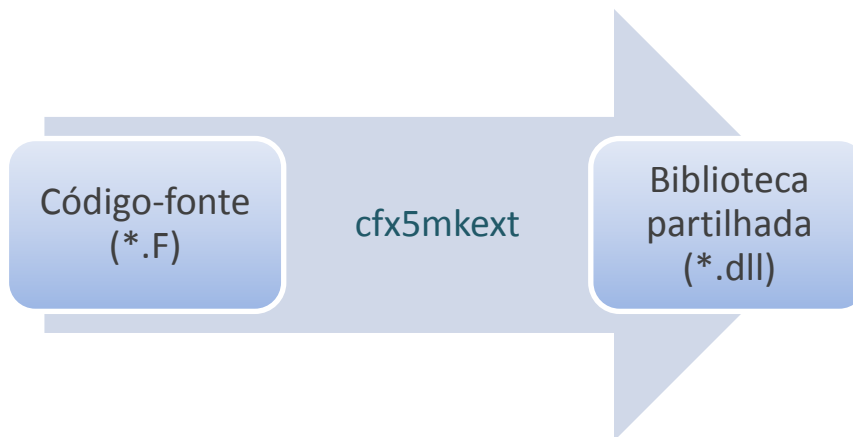


Figura 4.1 Ilustração do conceito de compilação de ficheiros no CFX

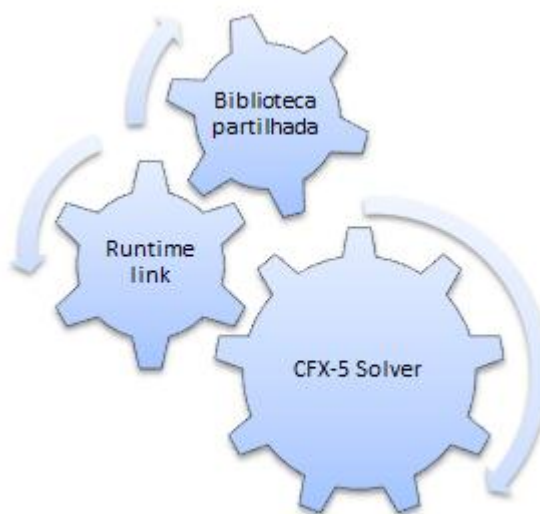


Figura 4.2 Ilustração do conceito de carregamento dinâmico da biblioteca partilhada por parte do CFX

Uma rotina FORTRAN definida pelo utilizador pode conter outras rotinas, também elas definidas pelo utilizador, e que podem ser chamadas a partir do processador CFX-5, assim como rotinas secundárias que são chamadas somente a partir de outras rotinas (que não a principal).

A estrutura básica de uma função CEL encontra-se ilustrada na figura 4.3.

```

#include "cfx5ext.h"
dllexport(callingname)
  SUBROUTINE CALLINGNAME (
    & NLOC, NRET, NARG, RET, ARGS, CRESLT, CZ,DZ,IZ,LZ,RZ )
C
  INTEGER NLOC,NARG,NRET
  CHARACTER CRESLT*(*)
  REAL ARGS(NLOC,NARG), RET(NLOC,NRET)
C
  INTEGER IZ(*)
  CHARACTER CZ(*)*(1)
  DOUBLE PRECISION DZ(*)
  LOGICAL LZ(*)
  REAL RZ(*)
C
  .... executable statements
  END

```

Figura 4.3 Estrutura básica de uma função CEL

A *macro* `dllexport()` é usada para assegurar que o nome de chamada da rotina é conhecido externamente. A *macro* é definida através de um ficheiro *include*, pelo que `#include "cfx5ext.h"` deve estar presente na primeira linha do ficheiro FORTRAN. Deve ser especificado ainda um `dllexport()` por cada rotina que o processador do CFX (CFX-5) pode chamar a partir do ficheiro FORTRAN. Cada `dllexport()` deve preceder a declaração *SUBROUTINE* à qual se refere, e deve começar na primeira coluna. O seu argumento deve ser o nome da subrotina, em minúsculas e sem espaçamentos.

Os argumentos padrão da subrotina são descritos na tabela 4.1.

Tabela 4.1 Lista de argumentos padrão relativa a uma rotina do tipo *User CEL Function*

<i>Argumento</i>	<i>Tipo de dados</i>	<i>I/O</i>	<i>Descrição</i>
<i>NLOC</i>	<i>INTEGER</i>	<i>IN</i>	Número de pontos do espaço
<i>NRET</i>	<i>INTEGER</i>	<i>IN</i>	Número de variáveis de saída
<i>NARG</i>	<i>INTEGER</i>	<i>IN</i>	Número de argumentos
<i>RET(NLOC,NRET)</i>	<i>REAL array</i>	<i>OUT</i>	Campo de dados de saída
<i>ARGS(NLOC,NARGS)</i>	<i>REAL array</i>	<i>IN</i>	Campo de argumentos
<i>CRESLT</i>	<i>CHARACTER*4</i>	<i>OUT</i>	Resultado da chamada
<i>CZ, DZ, IZ, LZ, RZ</i>	Vários	<i>MOD</i>	Stacks – aceder a dados de processamento

4.1. Aplicação ao caso em estudo

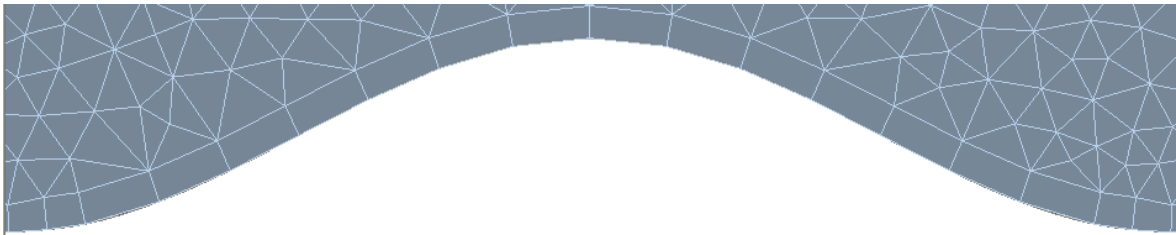


Figura 4.4 Ilustração da camada de células correspondentes à fronteira *MovingBoundary*

Para o caso ilustrado na figura **Figura 4.4 Ilustração da camada de células correspondentes à fronteira *MovingBoundary***, procedeu-se à definição dos argumentos patentes nas rotinas FORTRAN, considerando que, ao ser chamada a rotina FORTRAN que procede ao cálculo da altura de partículas arrancadas da superfície da colina, seria aplicada a função descrita pelo seguinte modelo:



Figura 4.5 Esquema simplificado dos argumentos necessários ao modelo, e do resultado produzido

Desta forma, a Tabela 4.1 tomará a seguinte forma, apresentada na tabela 4.2:

Tabela 4.2 Lista de argumentos de uma rotina para o caso da figura 4.4

<i>Argumento</i>	<i>Tipo de dados</i>	<i>I/O</i>	<i>Valor</i>
NLOC	INTEGER	IN	18
NRET	INTEGER	IN	1
NARG	INTEGER	IN	3
RET(NLOC,NRET)	REAL array	OUT	(18,1)
ARGS(NLOC,NARGS)	REAL array	IN	(18,3)
CRESLT	CHARACTER*4	OUT	'GOOD' se a chamada tiver sucesso
CZ, DZ, IZ, LZ, RZ	Vários	MOD	ARGS() importados do processador

5. RESULTADOS

Os seguintes resultados, ilustrados através da figura 5.1 foram obtidos através de uma rotina que define um valor constante de deformação do perfil da colina de um milímetro por cada intervalo de tempo, o que produziu um deslocamento de quatro milímetros, considerando que se simulou um tempo total de quatro segundos, com intervalos de tempo de um segundo.

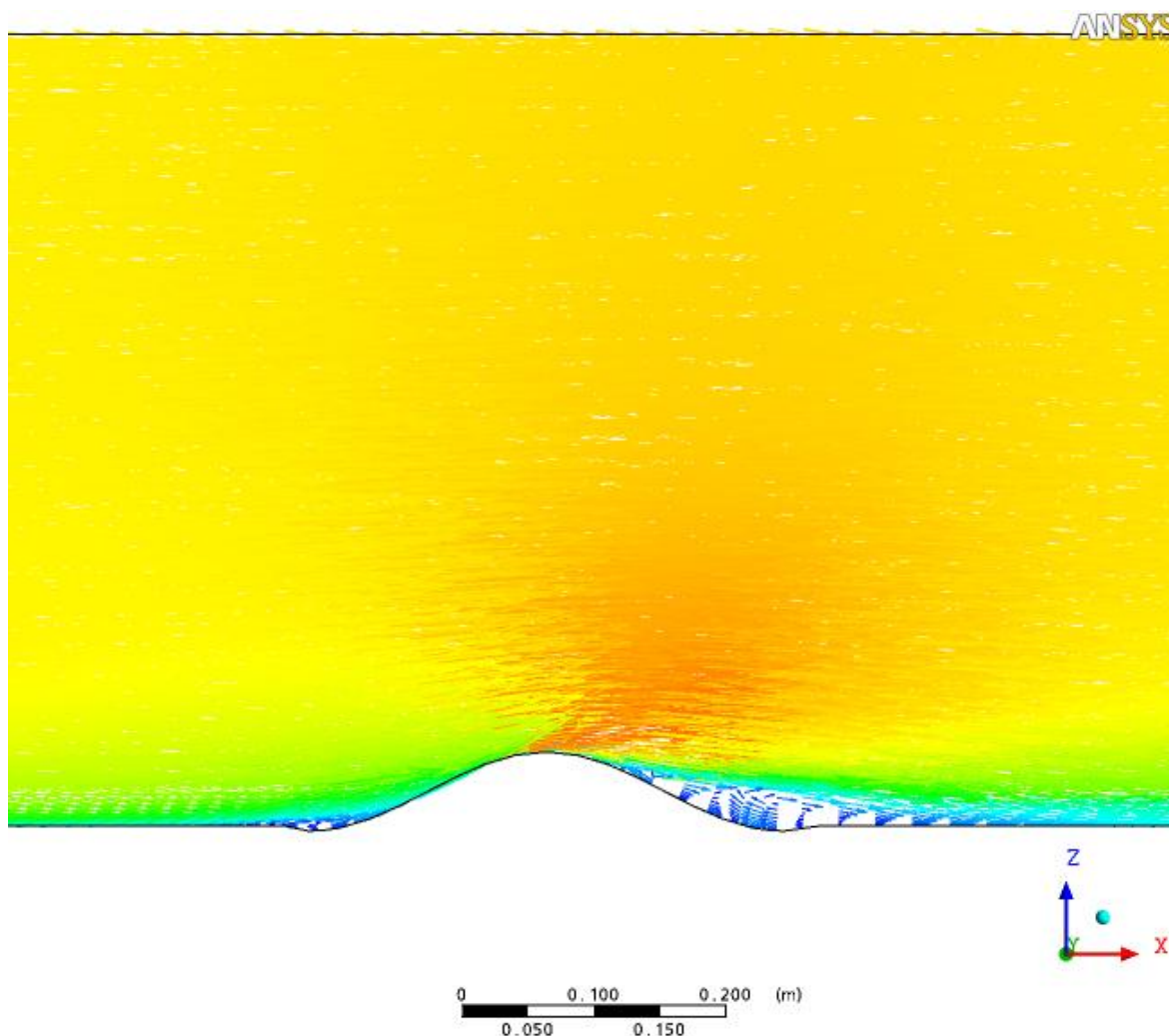


Figura 5.1 Deformação do perfil longitudinal da colina recorrendo à utilização de uma rotina FORTRAN

5.1. Discussão dos resultados

Os resultados obtidos, embora não reproduzam de forma fiel o que o fenómeno da erosão, para a rotina que simula um abaixamento constante da colina em cada intervalo de tempo, ilustra de forma correta aquilo que se esperava que ela fizesse. Contudo, é de referir que se deve corrigir a rotina de forma a impedir que se verifique erosão abaixo do nível do solo ($z=0$, coincidente com a base da colina), assim como um abaixamento acumulado da colina superior à altura máxima da colina.

6. CONCLUSÕES

O objetivo principal deste trabalho era simular, com recurso a rotinas FORTRAN definidas pelo utilizador, o abaixamento da superfície da colina, sendo que este abaixamento resulta do processo de erosão. Nessa medida, pode-se dizer que o objetivo foi parcialmente atingido, mas apenas para o caso da simulação de uma abaixamento uniforme ao longo de toda a colina, e constante em cada intervalo de tempo.

Para trabalhos futuros, que eventualmente venham a ser realizados na prossecução do trabalho que aqui se apresenta, recomenda-se a implementação dos modelos de Bagnold (1941) e de Lettau e Lettau (1978). As rotinas necessárias para tal foram parcialmente desenvolvidas e são apresentadas nos anexos A e B. Espera-se que, quando implementadas no software CFX, se consiga simular de forma mais realista o fenómeno da erosão.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ansys CFX (2010). Retrieved May 10, 2011, from <http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/ANSYS+CFX>.
- Bagnold, R.A. 1937, “The transport of sand by wind”. *Geographical Journal* 89, 409-438.
- Bagnold, R.A. 1941, “The Physics of Blown Sand and Desert Dunes”, Methuen, London, pp. 265 (re-impreso em 1954).
- Bisal, F., e Nielson, K. F. (1962), “Movement of Soil Particles in Saltation”, *Canadian Journal of Soil Science*, 42, 81-86.
- Doormaal, J. P. V. e Raithby, G. D. (1984). Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows. *Numerical Heat Transfer* 7 (2), p. 147-163
- Ferreira, A.D. e Fino, M.R.M. (2011), “A wind tunnel study of wind erosion and profile reshaping of transverse sand piles in tandem”, *Geomorphology*, 139–140, 230–241
- Gouveia, J. e Dias, A. (1995). *Geologia 12*. (A. Editores, Ed.). Porto.
- Houaiss, A. e Villar, M. de S. (2005). *Dicionário Houaiss da Língua Portuguesa*. (T. e Debates, Ed.). Porto: Global Notícias Publicações, SA.
- Iversen, J. D. e Rasmussen, K. R. (1994). The effect of surface slope on saltation threshold. *Sedimentology*, 41(4), 721-728. doi: 10.1111/j.1365-3091.1994.tb01419.x.
- Lauder, B. e Spalding, D. (1974). The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3, 269-289.
- Lettau, K. e Lettau, H. (1978), Experimental and micrometeorological field studies of dune migration. In H. H. Lettau and K. Lettau, editors, *Exploring the world's driest climate*. Madison, Center for Climatic Research, Univ. Wisconsin-Madison, pp.110-147, IES Report 101.
- Pye, K. e Tsoar, H. (2009), *Aeolian Sand and Sand Dunes*, Springer, Berlin, p. 458.
- Sherman, D.J. e Li, B. (2012), *Aeolian Research*, 3, 371-378.

ANEXO A – ROTINA FORTRAN – MODELO DE BAGNOLD (1941)

Os resultados apresentados obedeceriam às equações calculadas posteriormente à definição da velocidade de atrito crítica, apresentadas no sub-capítulo 3.1 se, em vez de ter sido utilizado a rotina atual, fosse utilizada uma rotina que tivesse como modelo a seguinte estrutura:

```
#include "cfx5ext.h"
dlllexport(dh_entrainm)
  SUBROUTINE DH_ENTRAINM2 (
    & NLOC, NRET, NARG, RET, ARGS, CRESLT, CZ,DZ,IZ,LZ,RZ )
CC User routine: calculates the timestep deformation of the loose boundary
CC The total mesh displacement was implemented in conjunction with each time step displacement
CC -----
CC   Input
CC -----
CC NLOC  - size of current locale
CC NRET  - number of components in result
CC NARG  - number of arguments in call
CC ARGS() - (NLOC,NARG) argument values
CC -----
CC   Arguments [Input] List
CC -----
CC [1] Shear velocity           [m s-1]
CC [2] Time step                [s]
CC [3] Particle diameter        [m]
CC -----
CC   Return [Output] List
CC -----
CC [1] Mesh displacement        [m]
CC RET() - (NLOC,NRET) return values
CC CRESLT - 'GOOD' for success
CC -----
CC   USER VARIABLES
CC -----
CC NDHPOS - number of locals with a displacement different from zero
```

```

CC=====
C -----
C   Preprocessor includes
C -----
#include "MMS.h"
#include "stack_point.h"
#include "locale_params.h"
#include "cfd_constants.h"
C -----
C   Argument list
C -----
    INTEGER NLOC,NARG,NRET
    CHARACTER CRESLT*(*)
    REAL ARGS(NLOC,NARG), RET(NLOC,NRET)
    INTEGER IZ(*)
    CHARACTER CZ(*)*(1)
    DOUBLE PRECISION DZ(*)
    LOGICAL LZ(*)
    REAL RZ(*)
C -----
C   External routines
C -----
    CHARACTER CFROMR*20
    EXTERNAL CFROMR
    CHARACTER CFROMI*5
    EXTERNAL CFROMI
C -----
C   Local Variables
C -----
    INTEGER ILOC, NDHPOS
    REAL DELTATIME, DN1, DREF, PI, ALPHAB1, RHOAIR, RHOSAND, GRAV
    & DHTOT, DHB, MASSFLUXENT, SHEARVEL, SHEARVELCT, SHEARVELEXC
    & LSALT, TOTMASSFLUXENT, MASSFLUXENT

    REAL DHTOT
C -----
C   Executable Statements
C -----

C Initialise RET() to zero.

```

```

CALL SET_A_0( RET, NLOC*NRET )
C -----
C   Constants
C -----
      ILOC = 1
      DELTATIME = ARGS(ILOC,2)           !Time step
      DN1 = ARGS(ILOC,3)                 !Particle diameter [m]
      DREF = 0.00025                     !Reference sand grain diameter [m]
      PI = ACOS(0.0)                     !Set Pi
      ALPHAB1 = 3.14159/6                !considering alpha to be equal to Pi/6 [rad]
      RHOSAND = 2650                     !for each grain of sand [kg m**3]
      RHOAIR = 1.18                       !for air [kg m**3]
      GRAV = 9.81                         ![m s**-2]
C -----
C   Expressions
C -----
      DHTOT = 0
      DHT = 0
      NDHPOS = 0
      DO ILOC = 1,NLOC
      SHEARVEL = ARGS (ILOC, 1)           !Shear velocity = u*
      SHEARVELCT = 0.33
      SHEARVELEXC = (SHEARVEL-SHEARVELCT)
CC   Velocity difference between the shear velocity and its threshold

      LSALT = 4.549*SHEARVELEXC           !Saltation length

      IF (SHEARVELEXC>0) THEN
TOTMASSFLUXENT = 1.8*SQRT(DN1/DREF)*RHOAIR*SHEARVEL**3/GRAV
CC Total sand flux
      MASSFLUXENT = TOTMASSFLUXENT/LSALT   !Sand flux
      DHB = MASSFLUXENT*DELTATIME/RHOAIR   !Decay in height
      ELSE
      DHB = 0                               !Decay in height
      END IF

      DHTOT = DHTOT - DHB
      RET(ILOC,1) = DHB
      IF (DHB.LT.0.) THEN
      CALL MESSAGE( 'WRITE', CFROMI(ILOC) // ' DHB = ' // CFROMR (DHB))

```

```
        NDHPOS = NDHPOS +1
    ENDIF
END DO
IF (NDHPOS.GT.0) THEN
    DHTOT = DHTOT / NDHPOS
    CALL MESSAGE( 'WRITE', '***** Entrain 1.1.1 *****')
    CALL MESSAGE( 'WRITE', 'NDHPOS = ' // CFROMI (NDHPOS) //
    &   ' DHTTOT MEDIO = ' // CFROMR (DHTOT) )
    CALL MESSAGE ( 'WRITE-ASYS', ' ')
ENDIF
C Set success flag.
CRESLT = 'GOOD'
C=====
END
```

ANEXO B – ROTINA FORTRAN – MODELO DE LETTAU E LETTAU (1978)

Por sua vez, os resultados apresentados obedeceriam às equações calculadas posteriormente à definição da velocidade de atrito crítica apresentadas no sub-capítulo 3.1, utilizando o modelo baseado na equação 3.6 de Lettau e Lettau (1978) se, considerando o modelo de rotina baseado no modelo de Bagnold (1941, equação 3.5), substituíssemos a expressão relativa à equação 3.5 por uma outra que expressasse a equação 3.6:

```

C -----
C   Expressions
C -----
      DHTOT = 0
      DHT = 0
      NDHPOS = 0
      DO ILOC = 1,NLOC
      SHEARVEL = ARG5 (ILOC, 1)           !Shear velocity = u*
      SHEARVELCT = 0.33
      SHEARVELEXC = (SHEARVEL-SHEARVELCT)
CC   Velocity difference between the shear velocity and its threshold

      LSALT = 4.549*SHEARVELEXC          !Saltation length

      IF (SHEARVELEXC>0) THEN
TOTMASSFLUXENT = 6.7*SQRT(DN1/DREF)*RHOAIR*SHEARVEL**2*SHEARVELEXC/GRAV
CC Total sand flux
          MASSFLUXENT = TOTMASSFLUXENT/LSALT          !Sand flux
          DHB = MASSFLUXENT*DELTATIME/RHOAIR          !Decay in height
      ELSE
          DHB = 0                                       !Decay in height
      END IF

      DHTOT = DHTOT - DHB
      RET(ILOC,1) = DHB
      IF (DHB.LT.0.) THEN
          CALL MESSAGE( 'WRITE', CFROMI(ILOC) // ' DHB = ' // CFFROMR (DHB))
          NDHPOS = NDHPOS +1
      ENDIF
      END DO

```

```
IF (NDHPOS.GT.0) THEN
  DHTOT = DHTOT / NDHPOS
  CALL MESSAGE( 'WRITE', '***** Entrain 1.1.1 *****')
  CALL MESSAGE( 'WRITE', 'NDHPOS = ' // CFROMI (NDHPOS) //
    & ' DHTTOT MEDIO = ' // CFROMR (DHTOT) )
  CALL MESSAGE ( 'WRITE-ASYS', ' ')
ENDIF
C Set success flag.
CRESLT = 'GOOD'
C=====
END
```