

```

#include <stdio.h>
#include <stdlib.h>
#include<string.h>
void GraficosMatlab(char nomePasta[20],int numX,int numY);
int main(int argc, char *argv[])
{
    FILE *saida,*perfil,*encargos,*config,*batch,*macroPre,*fluent,*matlab;
    FILE *eixoX,*eixoY,*fidCorda;
    char NomeFicheiro[20],nomePerfil[20],output[8],nomeProjecto[20],directoria[121];
    char lixo[121],caminhoPasta[141],path_original[121];//nomeScript[30];
    char caminhoFluent[141],caminhoICEM[141],caminhoMATLAB[141],nomeSaida[141];
    char nomeMalha[27],nomeFicheiroPre[35],caminhoSolverFluent[120];
    char *grandeza[]={ "Drag", "Lift", "Momento", "X_CP", "Y_CP"};
    char *tipo[]={ "Pressao", "Viscoso", "Total"};
    float *coordCimaX,*coordCimaY,*coordBaixoX,*coordBaixoY;
    float corda,*incidencia,*velocidadeU;
    int numeroPontos,ponto,curva,indice,i,numeroCasos, caso,numeroVelocidades;
    int numeroIncidencias,indiceIncidencia,indiceVelocidade,figura=1,intAuxiliar;

    if(argc==2)
    {
        sprifclose(matlab);o
    }
    else if (argc==1)
    {
        printf("Introduza o nome do ficheiro:\n\t");
        gets(NomeFicheiro);
    }
    else
    {
        printf("\nErro: Demasiados argumentos inseridos!\n");
        system("PAUSE");
        return 0;
    }

    encargos=fopen(NomeFicheiro,"r");
    if(encargos==NULL)
    {
        printf("\nErro: Ficheiro %s nao encontrado ou inacessivel!\n\n",NomeFicheiro);
    }

```

```

        system("PAUSE");
        return 0;
    }
else
{
    fscanf(encargos, "%d", &numeroCasos);
    if(numeroCasos<1)
    {
        printf("\nErro: Numero de casos deve ser um inteiro positivo!\n\n");
        system("pause");
        return 0;
    }
    fscanf(encargos, "%c", &lixo[0]);
    fgets(directoria, 121, encargos);
    //tirar alguma mudança de linha e espaços
    i=strlen(directoria)-1;
    while((directoria[i]!='\n')||(directoria[i]==' '))
    {
        directoria[i]='\0';
        i--;
    }

    fgets(lixo, 120, encargos);
    //analisar barra
    i=strlen(directoria)-1;
    if((i!=4)&&(directoria[i]=='\'))
    {
        directoria[i]='\0';
    }

    for(i=0;i<=strlen(directoria);i++)
    {
        path_original[i]=directoria[i];
    }

    sprintf(lixo, "cd \"%s\"", path_original);
    system(lixo);

    sprintf(lixo, "%s\\Configuracoes.ini", path_original);

```

```

config=fopen(lixo,"r");
if(config==NULL)
{
    printf("\nErro: Nao foi possivel aceder ao ficheiro 'Configuracoes.ini'\n\n");
    system("PAUSE");
    return 0;
}

sprintf(lixo,"%s\\Zzad.bat",path_original);
batch=fopen(lixo,"w");
if(batch==NULL)
{
    printf("\nErro: Nao foi possivel criar o ficheiro Zzad.bat!\n\n");
    system("PAUSE");
    return 0;
}

fgets(lixo,120,config);
fgets(caminhoFluent,120,config);
camifclose(matlab);n(caminhoFluent)-

fgets(lixo,120,config);
fgets(caminhoICEM,120,config);
caminhoICEM[strlen(caminhoICEM)-1]='\0'; //Tirar mudana de linha

fgets(lixo,120,config);
fgets(caminhoSolverFluent,120,config);
caminhoSolverFluent[strlen(caminhoSolverFluent)-1]='\0'; //Tirar mudana de linha

fgets(lixo,120,config);
fgets(caminhoMATLAB,120,config);
caminhoMATLAB[strlen(caminhoMATLAB)-1]='\0'; //Tirar mudana de linha

fprintf(batch,"@echo off\n");

fprintf(batch,"cd \"%s\"\n",directoria);

for(caso=0;caso<numeroCasos;caso++)
{

```

```

fscanf(encargos, "%s", &nomePerfil);
fscanf(encargos, "%f", &corda);
fscanf(encargos, "%s", &nomeProjecto);
fscanf(encargos, "%d", &numeroIncidencias);
fscanf(encargos, "%d", &numeroVelocidades);

sprintf(caminhoPasta, "mkdir \"%s\\%s\"", directoria, nomeProjecto);
system(caminhoPasta);
sprintf(lixo, "mkdir \"%s\\%s\\Graficos\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Lift\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Drag\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\X_CP\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Y_CP\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Momento\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Macros\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Simulacoes\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Resultados\"", directoria, nomeProjecto);
system(lixo);
sprintf(lixo, "mkdir \"%s\\%s\\Convergencia\"", directoria, nomeProjecto);
system(lixo);

sprintf(lixo, "%s\\%s\\Graficos\\Corda.dat", path_original, nomeProjecto);
fidCorda=fopen(lixo, "w");
if(fidCorda==NULL)
{
    printf("\nErro: Nao foi possivel criar o ficheiro 'Corda.dat'!\n\n");
    system("PAUSE");
    return 0;
}
fprintf(fidCorda, "%.3f\n", corda);
fclose(fidCorda);

```

```

sprintf(lixo, "%s\\%s\\Graficos\\Incidencia.dat", path_original, nomeProjecto);
eixoX=fopen(lixo, "w");
if(eixoX==NULL)
{
    printf("\nErro: Nao foi possivel criar o ficheiro 'Incidencia.dat'!\n\n");
    system("PAUSE");
    return 0;
}

sprintf(lixo, "%s\\%s\\Graficos\\Velocidade.dat", path_original, nomeProjecto);
eixoY=fopen(lixo, "w");
if(eixoY==NULL)
{
    printf("\nErro: Nao foi possivel criar o ficheiro 'Velocidade.dat'!\n\n");
    system("PAUSE");
    return 0;
}

sprintf(lixo, "%s\\%s\\Correr_FLUENT_%s.bat", path_original, nomeProjecto, nomeProjecto);
fluent=fopen(lixo, "w");
if(fluent==NULL)
{
    printf("\nErro: Nao foi possivel criar '%s'!\n\n", lixo);
    system("PAUSE");
    return 0;
}
fprintf(fluent, "@echo off\n");

if(numeroIncidencias<1)
{
    printf("\nErro: Numero de Incidencias deve ser um inteiro positivo!\n\n");
    system("pause");
    return 0;
}
if(numeroVelocidades<1)
{
    printf("\nErro: Numero de Velocidades deve ser um inteiro positivo!\n\n");
    system("pause");
}

```

```

    return 0;
}

fscanf(encargos, "%c", &lixo[0]); //descartar mudança de linha

incidencia=(float *)malloc(sizeof(float)*numeroIncidencias);
velocidadeU=(float *)malloc(sizeof(float)*numeroVelocidades);

for(indiceIncidencia=0;indiceIncidencia<numeroIncidencias;indiceIncidencia++)
{
    fscanf(encargos, "%f", &incidencia[indiceIncidencia]);
    fprintf(eixoX, "%.1f\n", incidencia[indiceIncidencia]);
}
fclose(eixoX);

fgets(lixo, 10, encargos); //descartar mudança de linha

for(indiceVelocidade=0;indiceVelocidade<numeroVelocidades;indiceVelocidade++)
{
    fscanf(encargos, "%f", &velocidadeU[indiceVelocidade]);
    fprintf(eixoY, "%.1f\n", velocidadeU[indiceVelocidade]);
}
fclose(eixoY);

fscanf(encargos, "%c", &lixo[0]); //descartar mudança de linha

for(i=0;i<strlen(caminhoPasta);i++)
{
    if(caminhoPasta[i]=='\\')
    {
        caminhoPasta[i]='/'; //virar as barras ao contrário
    }
}
for(i=0;i<strlen(directoria);i++)
{
    if(directoria[i]=='\\')
    {
        directoria[i]='/'; //virar as barras ao contrário
    }
}

```

```

    }
}
sprintf(caminhoPasta, "%s/%s", directoria, nomeProjecto);

//=====================================================
for(indiceIncidencia=0; indiceIncidencia<numeroIncidencias; indiceIncidencia++)
{
    sprintf(nomeSaida, "%s\\%s\\Macros\\Script_ICEM_%s_%.0f_.rpl", path_original, nomeProjecto, nomeProjecto, ...
    incidencia[indiceIncidencia]*10);
    saida=fopen(nomeSaida, "w");
    if(saida==NULL)
    {
        printf("\nErro: Nao foi possivel criar o ficheiro '%s'!\n\n", nomeSaida);
        system("PAUSE");
        return 0;
    }

    fprintf(batch, "call \"%s\" -batch -script %s \n", caminhoICEM, nomeSaida); //correr ICEM em background
    sprintf(nomeMalha, "%s_%.0f_", nomeProjecto, incidencia[indiceIncidencia]*10);

    //Pontos FRONTEIRA
    fprintf(saida, "ic_geo_new_family GEOM\n");
    fprintf(saida, "ic_boco_set_part_color GEOM\n");
    fprintf(saida, "ic_point {} GEOM pnt.0 -11,0,0\n");
    fprintf(saida, "ic_point {} GEOM pnt.1 1,-12,0\n");
    fprintf(saida, "ic_point {} GEOM pnt.2 21,-12,0\n");
    fprintf(saida, "ic_point {} GEOM pnt.3 21,12,0\n");
    fprintf(saida, "ic_point {} GEOM pnt.4 1,12,0\n");

    //Ler Coordenadas Perfil
    sprintf(lixo, "%s\\%s", path_original, nomePerfil);
    perfil=fopen(lixo, "r");
    if(perfil==NULL)
    {
        printf("\nFicheiro %s nao encontrado ou inacessivel!\n\n", lixo);

        system("PAUSE");
    }
}

```

```

        return 0;
    }

    fscanf(perfil, "%d", &numeroPontos);
    coordBaixoX=(float *)malloc(sizeof(float)*numeroPontos);
    coordBaixoY=(float *)malloc(sizeof(float)*numeroPontos);
    coordCimaX=(float *)malloc(sizeof(float)*numeroPontos);
    coordCimaY=(float *)malloc(sizeof(float)*numeroPontos);
    for(indice=0;indice<numeroPontos;indice++)
    {
        fscanf(perfil, "%f %f", &coordCimaX[indice], &coordCimaY[indice]);
    }
    for(indice=0;indice<numeroPontos;indice++)
    {
        fscanf(perfil, "%f %f", &coordBaixoX[indice], &coordBaixoY[indice]);
    }
    fclose(perfil);

    //Desenhar pontos perfil
    ponto=5;
    for(indice=0;indice<numeroPontos;indice++)
    {
        fprintf(saida, "ic_point {} GEOM pnt.%d %.5f,%.5f,0\n", ponto, coordCimaX[indice], coordCimaY[indice]);
        ponto++;
    }
    for(indice=0;indice<numeroPontos;indice++)
    {
        fprintf(saida, "ic_point {} GEOM pnt.%d %.5f,%.5f,0\n", ponto, coordBaixoX[indice], coordBaixoY[indice]);
        ponto++;
    }

    //Curvas
    fprintf(saida, "ic_geo_new_family ENTRADA\n");
    fprintf(saida, "ic_boco_set_part_color ENTRADA\n");
    fprintf(saida, "ic_curve point ENTRADA crv.00 {pnt.3 pnt.4}\n");
    fprintf(saida, "ic_curve arc ENTRADA crv.01 {pnt.4 pnt.0 pnt.1}\n");
    fprintf(saida, "ic_curve point ENTRADA crv.02 {pnt.1 pnt.2}\n");

    fprintf(saida, "ic_geo_new_family SAIDA\n");

```

```

fprintf(saida, "ic_boco_set_part_color SAIDA\n");
fprintf(saida, "ic_curve point SAIDA crv.03 {pnt.2 pnt.3}\n");

//Desenhar curvas do perfil com os pontos
fprintf(saida, "ic_geo_new_family PERFIL\n");
fprintf(saida, "ic_boco_set_part_color PERFIL\n");
ponto=5;
for(i=4;i<6;i++)
{
    fprintf(saida, "ic_curve point PERFIL crv.0%d {" ,i);
    for(indice=0;indice<numeroPontos;indice++)
    {
        fprintf(saida, " pnt.%d", ponto);
        ponto++;
    }
    fprintf(saida, "}\n");
}

//Rodar perfil
fprintf(saida, "ic_move_geometry curve names {crv.04 crv.05} rotate %.2f rotate_axis {0 0 1} cent...
{1.0 0.0 0.0}\n", -1*incidencia[indiceIncidencia]);

//Criar superfície FLUIDO
fprintf(saida, "ic_geo_new_family FLUIDO\n");
fprintf(saida, "ic_boco_set_part_color FLUIDO\n");
fprintf(saida, "ic_surface 2-4crvs FLUIDO srf.00 {0.01 {crv.00 crv.01 crv.02 crv.03}}\n");
fprintf(saida, "ic_set_dormant_pickable point 0 {}\n");
fprintf(saida, "ic_set_dormant_pickable curve 0 {}\n");

//Criar blocking
fprintf(saida, "ic_hex_initialize_mesh 2d new_numbering new_blocking FLUIDO\n");
fprintf(saida, "ic_hex_switch_blocking root\n");
fprintf(saida, "ic_hex_unblank_blocks \n");
fprintf(saida, "ic_hex_multi_grid_level 0\n");
fprintf(saida, "ic_hex_projection_limit 0\n");
fprintf(saida, "ic_hex_default_bunching_law default 2.0\n");
fprintf(saida, "ic_hex_floating_grid off\n");
fprintf(saida, "ic_hex_transfinite_degree 1\n");
fprintf(saida, "ic_hex_unstruct_face_type one_tri\n");

```

```

fprintf(saida, "ic_hex_set_n_tetra_smoothing_steps 20\n");
fprintf(saida, "ic_hex_set_mesh_params GEOM ENTRADA SAIDA PERFIL FLUIDO -version 110\n");
fprintf(saida, "ic_hex_error_messages off_minor\n");
fprintf(saida, "ic_hex_switch_blocking root\n");

//Associar arestas e vértices do blocking à fronteira
fprintf(saida, "ic_hex_set_edge_projection 19 21 0 1 crv.03\n");
fprintf(saida, "ic_hex_create_composite {crv.00 crv.01 crv.02}\n");
fprintf(saida, "ic_hex_set_edge_projection 13 21 0 1 crv.00\n");
fprintf(saida, "ic_hex_set_edge_projection 11 13 0 1 crv.00\n");
fprintf(saida, "ic_hex_set_edge_projection 11 19 0 1 crv.00\n");
fprintf(saida, "ic_hex_project_to_surface GEOM PERFIL SAIDA ENTRADA FLUIDO\n");

//O-Grid
fprintf(saida, "ic_hex_mark_blocks unmark\n");
fprintf(saida, "ic_hex_mark_blocks superblock 4\n");
fprintf(saida, "ic_hex_mark_blocks numbers 19 21 edge_neighbors\n");
fprintf(saida, "ic_hex_ogrid 1 m GEOM ENTRADA SAIDA PERFIL FLUIDO -version 50\n");
fprintf(saida, "ic_hex_mark_blocks unmark\n");
fprintf(saida, "ic_hex_mark_blocks unmark\n");
fprintf(saida, "ic_hex_mark_blocks unmark\n");
fprintf(saida, "ic_hex_mark_blocks unmark\n");

//Split block
fprintf(saida, "ic_hex_split_grid 33 35 pnt.%d m GEOM ENTRADA SAIDA PERFIL FLUIDO VORFN\n", ...
4+2*numeroPontos);

//Apagar block dentro do perfil
fprintf(saida, "ic_hex_mark_blocks unmark\n");
fprintf(saida, "ic_hex_mark_blocks superblock 4\n");
fprintf(saida, "ic_hex_change_element_id VORFN\n");

//Mover vértices do bloco sob a fronteira
fprintf(saida, "ic_hex_place_node 41 curve:crv.01 0.0\n");
fprintf(saida, "ic_hex_place_node 39 curve:crv.02 0.0\n");
fprintf(saida, "ic_hex_place_node 13 curve:crv.01 0.45\n");
fprintf(saida, "ic_hex_place_node 11 curve:crv.01 0.55\n");

//Associar arestas ao perfil

```



```

fprintf(saida, "ic_hex_project_to_surface 32 33\n");

//Cortar bloco para criar zona refinada junto ao perfil
fprintf(saida, "ic_hex_split_grid 13 33 0.8 m GEOM ENTRADA SAIDA PERFIL FLUIDO VORFN\n");

//Ajustar a malha na zona da camada limite
fprintf(saida, "ic_hex_set_mesh 48 33 n 61 hlrel 0.0 h2rel 0.000622082046401 r1 1.2 r2 1.1 lmax...
0 default copy_to_parallel unlocked\n");

//Uniformizar malha (direcção vertical)
fprintf(saida, "ic_hex_set_mesh 13 48 n 50 hlrel 0.0 h2rel 0.0152827340368 r1 2 r2 2 lmax...
0 default copy_to_parallel unlocked\n");

//Ajustar malha na zona sob o perfil
fprintf(saida, "ic_hex_set_mesh 13 41 n 45 hlrel 0.0 h2rel 0.0 r1 2 r2 2 lmax...
1 default copy_to_parallel unlocked\n");

//Ajustar malha sobre o bordo de ataque
fprintf(saida, "ic_hex_set_mesh 11 13 n 11 hlrel 0.0 h2rel 0.111407856752 r1 2 r2 2 lmax...
1 uniform copy_to_parallel unlocked\n");

//Ajustar malha a jusante do perfil
fprintf(saida, "ic_hex_set_mesh 42 35 n 78 hlrel 0.0005 h2rel 0.015 r1 1.1 r2 1.5 lmax...
0 default copy_to_parallel unlocked\n");

fprintf(saida, "ic_chdir {%s}\n", caminhoPasta);

//Tornar a malha não estruturada
fprintf(saida, "ic_hex_write_file hex.uns GEOM ENTRADA SAIDA PERFIL FLUIDO proj 2 dim_to_mesh...
3 -family_boco family_boco.fbc\n");
fprintf(saida, "ic_uns_load hex.uns 3 0 { } 1\n");
fprintf(saida, "ic_uns_update_family_type visible {GEOM ORFN PERFIL SAIDA FLUIDO ENTRADA}...
{!NODE !LINE_2 QUAD_4} update 0\n");
fprintf(saida, "ic_uns_diag_reset_degen_min_max \n");
fprintf(saida, "ic_boco_solver \n");
fprintf(saida, "ic_uns_update_family_type visible {GEOM ORFN PERFIL SAIDA FLUIDO ENTRADA}...
{!NODE !LINE_2 QUAD_4} update 0\n");
fprintf(saida, "ic_boco_clear_icons \n");
fprintf(saida, "ic_csystem_display all 0\n");

```

```

fprintf(saida,"ic_csystem_set_current global\n");
fprintf(saida,"ic_boco_nastran_csystem reset\n");
fprintf(saida,"ic_uns_update_family_type visible {GEOM ORFN PERFIL SAIDA FLUIDO ENTRADA}...
{!NODE !LINE_2 QUAD_4} update 0\n");

//Ajustar ortogonalidade da malha
for(i=0;i<2;i++)
{
    fprintf(saida,"ic_unstruct_elliptical_smooth 10 10 0.0 0.0 1.0 2.0 0 0 202 0 -1 {PERFIL 0 0 }...
    1 1 {} {} {GEOM ORFN PERFIL SAIDA FLUIDO ENTRADA } 2 0\n");
}

//Guardar ficheiro contendo a malha e definir tipo de fronteiras para FLUENT
fprintf(saida,"ic_chdir {%s}\n",caminhoPasta);
fprintf(saida,"ic_save_tetin %s.tin 0 0 {} {} 0 0 1\n",nomeMalha);
fprintf(saida,"ic_uns_check_duplicate_numbers \n");
fprintf(saida,"ic_save_unstruct %s.uns 1 {} {} {} \n",nomeMalha);
fprintf(saida,"ic_uns_set_modified 1\n");
fprintf(saida,"ic_hex_save_blocking %s.blk\n",nomeMalha);
fprintf(saida,"ic_boco_solver \n");
fprintf(saida,"ic_boco_save %s.fbc\n",nomeMalha);
fprintf(saida,"ic_boco_solver Nastran\n");
fprintf(saida,"ic_solver_mesh_info Nastran\n");
fprintf(saida,"ic_boco_save_atr %s.atr\n",nomeMalha);
fprintf(saida,"ic_cart_is_loaded \n");
fprintf(saida,"ic_save_project_file {%s/%s.prj} ",caminhoPasta,nomeMalha);
    fprintf(saida,"{array\\ set\\ file_name\\ \\{ {   catia_dir .} {   parts_dir .}...
    {   domain_loaded 0} {   cart_file_loaded 0} {   cart_file {} } ");
    fprintf(saida,"{   domain_saved %s.uns} {   archive {} } {   med_replay {} }...
    {   topology_dir .} {   ugparts_dir .} ",nomeMalha);
    fprintf(saida,"{   icons {{$env(ICEM_ACN)/lib/ai_env/icons}...
    {{$env(ICEM_ACN)/lib/va/EZCAD/icons} {{$env(ICEM_ACN)/lib/icons}...
    {{$env(ICEM_ACN)/lib/va/CABIN/icons}} } ");
    fprintf(saida,"{   tetin %s.tin} {   family_boco %s.fbc} {   iges_dir .}...
    {   solver_params_loaded 0} {   attributes_loaded 0} {   project_lock {} } ",nomeMalha,nomeMalha);
    fprintf(saida,"{   attributes %s.atr} {   domain %s.uns} {   domains_dir .}...
    {   settings_loaded 0} {   settings %s.prj} ",nomeMalha,nomeMalha,nomeMalha);
    fprintf(saida,"{   blocking %s.blk} {   hexa_replay {} } {   transfer_dir .}...
    {   mesh_dir .} {   family_topo {} } {   gemsparts_dir .} ",nomeMalha);

```

```

fprintf(saida,"{ family_boco_loaded 0} { tetin_loaded 0} { project_dir .}...
{ topo_mulcad_out {} } { solver_params $s.nastran.par} \\} ",nomeMalha);
fprintf(saida,"array\\ set\\ options\\ \\{ { expert 1} { remote_path {} }...
{ tree_disp_quad 2} { tree_disp_pyra 0} { evaluate_diagnostic 0} ");
fprintf(saida,"{ histo_show_default 1} { select_toggle_corners 0} { remove_all 0}...
{ keep_existing_file_names 0} { record_journal 0} ");
fprintf(saida,"{ edit_wait 0} { face_mode all} { select_mode all}...
{ med_save_emergency_tetin 1} { user_name Administrator} { diag_which all} ");
fprintf(saida,"{ uns_warn_if_display 500000} { bubble_delay 1000} { external_num 1}...
{ tree_disp_tri 2} { apply_all 0} { default_solver {ANSYS CFX}} ");
fprintf(saida,"{ flood_select_angle 0} { home_after_load 1} { project_active 0}...
{ histo_color_by_quality_default 1} { undo_logging 1} { tree_disp_hexa 0} ");
fprintf(saida,"{ histo_solid_default 1} { host_name nietzsche} { xhidden_full 1}...
{ editor notepad} { mouse_color orange} { clear_undo 1} { remote_acn {} } ");
fprintf(saida,"{ remote_sh csh} { tree_disp_penta 0} { n_processors 1} { remote_host {} }...
{ save_to_new 0} { quality_info Quality} { tree_disp_node 0} ");
fprintf(saida,"{ med_save_emergency_mesh 1} { redtext_color red} { tree_disp_line 0}...
{ select_edge_mode 0} { use_dlremote 0} { max_mesh_map_size 1024} ");
fprintf(saida,"{ show_tris 1} { remote_user {} } { auto_save_views 1} { max_cad_map_size 512}...
{ display_origin 0} { uns_warn_user_if_display 1000000} ");
fprintf(saida,"{ detail_info 0} { win_java_help 0} { show_factor 1} { boundary_mode all}...
{ clean_up_tmp_files 1} { med_save_emergency_blocking 1} ");
fprintf(saida,"{ max_binary_tetin 0} { tree_disp_tetra 0} \\} array\\ set\\ disp_options\\ \\...
{ { uns_dualmesh 0} { uns_warn_if_display 500000} ");
fprintf(saida,"{ uns_normals_colored 0} { uns_icons 0} { uns_locked_elements 0}...
{ uns_shrink_npos 0} { uns_node_type None} { uns_icons_normals_vol 0} ");
fprintf(saida,"{ uns_bcfield 0} { backup Wire} { uns_nodes 0} { uns_only_edges 0}...
{ uns_surf_bounds 0} { uns_wide_lines 0} { uns_vol_bounds 0} ");
fprintf(saida,"{ uns_displ_orient Triad} { uns_orientation 0} { uns_directions 0}...
{ uns_thickness 0} { uns_shell_diagnostic 0} { uns_normals 0} ");
fprintf(saida,"{ uns_couplings 0} { uns_periodicity 0} { uns_single_surfaces 0}...
{ uns_midside_nodes 1} { uns_shrink 100} { uns_multiple_surfaces 0} ");
fprintf(saida,"{ uns_no_inner 0} { uns_enums 0} { uns_disp Wire} { uns_bcfield_name {} }...
{ uns_color_by_quality 0} { uns_changes 0} ");
fprintf(saida,"{ uns_cut_delay_count 1000} \\} {set use_xwindows 0} {set icon_size1 24}...
{set icon_size2 35} {set undo_save_whole_mesh 1} {set undo_max_memory 0} ");
fprintf(saida,"{set gsys_display_list_max_size 100} {set geom_fast_dynamics 1}...
{set med_fast_dynamics 1} {set tdv_dynamics_speed_thresh 300} {set med_float_precision 0} ");
fprintf(saida,"{set tdv_quadratic_accuracy 9} {set med_index_base 1}...

```

```

{set mesh_min_moving_element_aspect 0.1} {set thickness_defined 0} {set solver_type 1} ");
fprintf(saida,"{set solver_setup -1} {set tdv_background_color {0 #000000 #000000}}...
{set prompt_for_bld_topo 0} {set prompt_for_bld_topo_opts 0} {set named_sels_as_parts 1} ");
fprintf(saida,"array\\ set\\ prism_values\\ \\{ { n_triangle_smoothing_steps 5}...
{ min_smoothing_steps 6} { first_layer_smoothing_steps 1} { new_volume {} }");
fprintf(saida,"{ height {} } { prism_height_limit {} } { n_tetra_smoothing_steps 10}...
{ do_checks {} } { delete_standalone 1} { ortho_weight 0.50} ");
fprintf(saida,"{ max_aspect_ratio {} } { ratio_max {} } { total_height {} }...
{ use_prism_v10 0} { intermediate_write 1} { delete_base_triangles {} }");
fprintf(saida,"{ ratio_multiplier {} } { refine_prism_boundary 1} { max_size_ratio {} }...
{ triangle_quality {} } { max_prism_angle 180} { tetra_smooth_limit 0.3} ");
fprintf(saida,"{ max_jump_factor 5} { use_existing_quad_layers 0} { layers 3}...
{ fillet 0.10} { into_orphan 0} { init_dir_from_prev {} } { blayer_2d 0} ");
fprintf(saida,"{ do_not_allow_sticking {} } { top_family {} } { law exponential}...
{ min_smoothing_val 0.1} { auto_reduction 0} { stop_columns 1} ");
fprintf(saida,"{ stair_step 1} { smoothing_steps 12} { side_family {} }...
{ min_prism_quality 0.01} { ratio 1.2} \\} array\\ set\\ curve_properties\\ \\");
fprintf(saida,"{ emin_incr 0.1} { ewid_incr 1} { brat_incr 0.1} { hrat_incr 0.1}...
{ emax_incr 1.0} { edev_incr 0.1} { ehgt_incr 0.1} { nmax_incr 1} \\} ");
fprintf(saida,"array\\ set\\ mesh_option\\ \\{ { project_bspline 0} \\}...
{set aie_current_flavor {} } array\\ set\\ vid_options\\ \\{ { wb_NS_to_subset 0} { auxiliary 0} ");
fprintf(saida,"{ show_name 0} { inherit 0} { default_part GEOM} { new_srf_topo 1}...
{ DelPerFlag 0} { show_item_name 0} { composite_tolerance 1.0} ");
fprintf(saida,"{ replace 0} { same_pnt_tol 1e-4} { tdv_axes 1} { vid_mode 0}...
{ DelBlkPerFlag 0} \\} {set savedTreeVisibility }");
fprintf(saida,"{geomNode 1 geom_subsetNode 2 geomPointNode 0 geomCurveNode 0 geomSurfNode 0 meshNode 1...
mesh_subsetNode 2 meshPointNode 0 meshLineNode 2 meshShellNode 2 }");
fprintf(saida,"meshQuadNode 2 blockingNode 1 block_subsetNode 2 block_vertNode 0 block_edgeNode 2...
block_faceNode 0 block_blockNode 0 block_meshNode 0 topoNode 2 topo-root 2 ");
fprintf(saida,"partNode 2 part-ENTRADA 2 part-FLUIDO 2 part-GEOM 2 part-PERFIL 2 part-SAIDA 2...
part-VORFN 0} } {set last_view {rot {0 0 0 1} }");
fprintf(saida,"scale {313.460514767 313.460514767 313.460514767} center {0 0 0} pos {0.0 0.0 0} }...
array\\ set\\ cut_info\\ \\{ { active 0} ");
fprintf(saida,"{ whole 1} \\} array\\ set\\ hex_option\\ \\{ { default_bunching_ratio 2.0}...
{ floating_grid 0} { n_tetra_smoothing_steps 20} { trfDeg 1} ");
fprintf(saida,"{ wr_hexa7 0} { hexa_verbos_mode 0} { old_eparams 0} { multigrid_level 0}...
{ uns_face_mesh one_tri} { check_blk 1} { proj_limit 0} ");
fprintf(saida,"{ check_inv 0} { hexa_update_mode 1} { default_bunching_law BiGeometric} \\}...
array\\ set\\ saved_views\\ \\{ { views {} } \\} {set tdv_start_spaceball 1} ");

```

```

    fprintf(saida, "{ICEM CFD}\n");
    fprintf(saida, "ic_boco_solver Fluent_V6\n");
    fprintf(saida, "ic_solver_mesh_info Fluent_V6\n");
    fprintf(saida, "ic_boco_set GEOM { { 1 {color} 16663866 } }\n");
    fprintf(saida, "ic_boco_set ORFN {} \n");
    fprintf(saida, "ic_boco_set SAIDA {{1 PRESO 0} { 1 {color} 16601907 }}\n");
    fprintf(saida, "ic_boco_set PERFIL {{1 WALL 0} { 1 {color} 3406683 }}\n");
    fprintf(saida, "ic_boco_set ENTRADA {{1 VELI 0} { 1 {color} 16663866 }}\n");
    fprintf(saida, "ic_boco_set FLUIDO {{1 FLUID 0} { 1 {color} 9168691 }}\n");
    fprintf(saida, "ic_boco_set VORFN {} \n");
    fprintf(saida, "ic_boco_solver \n");
    fprintf(saida, "ic_boco_save %s.fbc\n", nomeMalha);
    fprintf(saida, "ic_boco_save_atr %s.atr\n", nomeMalha);
    fprintf(saida, "ic_exec {s} -dom {s/%s.uns} -b %s.fbc -dim2d ./%sfluent\n", caminhoFluent, caminhoPasta, ...
    nomeMalha, nomeMalha, nomeMalha);

```

//Criar journal files

```

for(indiceVelocidade=0; indiceVelocidade<numeroVelocidades; indiceVelocidade++)
{
    sprintf(nomeFicheiroPre, "%s%.0f_", nomeMalha, velocidadeU[indiceVelocidade]*10);

    sprintf(lixo, "%s\\%s\\%s.jou", path_original, nomeProjecto, nomeFicheiroPre);
    macroPre=fopen(lixo, "w");
    if(macroPre==NULL)
    {
        printf("\nErro: Nao foi possivel criar %s!\n\n", nomeFicheiroPre);

        system("PAUSE");
        return 0;
    }

    fprintf(macroPre, "/file/read-case %sfluent.msh\n", nomeMalha);
    fprintf(macroPre, "/grid/reorder/reorder-domain\n");
    fprintf(macroPre, "/grid/scale %.2f %.2f\n", corda, corda);
    fprintf(macroPre, "/define/boundary-conditions/velocity-inlet ");
    fprintf(macroPre, "entrada n y y n %.3f n 0.0\n", velocidadeU[indiceVelocidade]);

```

```
fprintf(macroPre, "/define/models/viscous/ke-standard yes\n");
fprintf(macroPre, "/solve/set/p-v-coupling 24\n");
fprintf(macroPre, "/solve/initialize/initialize-flow\n");
fprintf(macroPre, "/solve/patch (fluido) x-velocity yes %.3f\n", velocidadeU[indiceVelocidade]);
fprintf(macroPre, "/solve/set/reporting-interval 5\n");
fprintf(macroPre, "/solve/monitors/residual/check-convergence yes yes yes yes yes\n");
fprintf(macroPre, "/solve/monitors/residual/convergence-criteria 1e-5 1e-5 1e-5 1e-4 1e-4\n");
fprintf(macroPre, "/solve/monitors/force/drag yes (perfil) no yes \"drag_coeff_history_%s.dat\" no...
yes 1 0\n", nomeFicheiroPre);
fprintf(macroPre, "/solve/monitors/force/lift yes (perfil) no yes \"lift_coeff_history_%s.dat\" no...
yes 0 1\n", nomeFicheiroPre);
fprintf(macroPre, "/solve/monitors/force/moment yes (perfil) no yes \"moment_coeff_history_%s.dat\" no...
yes 0 0\n", nomeFicheiroPre);
fprintf(macroPre, "/solve/iterate 100\n");

fprintf(macroPre, "/adapt/adapt-to-y+ 0 1 0 200000 yes\n");
fprintf(macroPre, "/solve/iterate 50\n");

fprintf(macroPre, "/define/models/viscous/ke-realizable yes\n");
fprintf(macroPre, "/solve/set/discretization-scheme/pressure 14\n");
fprintf(macroPre, "/solve/set/discretization-scheme/epsilon 1\n");
fprintf(macroPre, "/solve/set/discretization-scheme/k 1\n");
fprintf(macroPre, "/solve/set/discretization-scheme/mom 1\n");
fprintf(macroPre, "/solve/iterate 50\n");

fprintf(macroPre, "/adapt/adapt-to-y+ 0 1 0 200000 yes\n");
fprintf(macroPre, "/solve/iterate 50\n");

fprintf(macroPre, "/adapt/adapt-to-y+ 0 1 0 200000 yes\n");
fprintf(macroPre, "/solve/iterate 50\n");

fprintf(macroPre, "/define/models/viscous/kw-sst yes\n");
fprintf(macroPre, "/solve/iterate/50\n");
fprintf(macroPre, "/adapt/adapt-to-y+ 0 1 0 200000 yes\n");
fprintf(macroPre, "/solve/iterate 300\n");

fprintf(macroPre, "/file/write-case-data %s\n", nomeFicheiroPre);
fprintf(macroPre, "/report/forces/wall-forces yes 1 0 yes Drag_%s.dat\n", nomeFicheiroPre);
```

```

        fprintf(macroPre, "/report/forces/wall-forces yes 0 1 yes Lift_%s.dat\n", nomeFicheiroPre);
        fprintf(macroPre, "/report/forces/wall-moments yes 0.0 0.0 yes Momento_%s.dat\n", nomeFicheiroPre);
        fprintf(macroPre, "/report/forces/pressure-center yes no yes 0.0 yes X_CP_%s.dat\n", nomeFicheiroPre);
        fprintf(macroPre, "/report/forces/pressure-center yes yes 0.0 yes Y_CP_%s.dat\n", nomeFicheiroPre);
        fprintf(macroPre, "/exit\n");

        fclose(macroPre);

        fprintf(fluent, "echo Calculando caso %s\n", nomeFicheiroPre);
        fprintf(fluent, "\"%s\" 2d -hidden -i %s.jou\n", caminhoSolverFluent, nomeFicheiroPre);
    }
}

free(coordBaixoX);
free(coordBaixoY);
free(coordCimaX);
free(coordCimaY);

}
fprintf(saida, "exit\n");
fclose(saida);

fprintf(batch, "cd \"%s\\%s\" \n", path_original, nomeProjecto);

close(fluent);
fprintf(batch, "echo Calculando processo %s ... \n", nomeProjecto);
fprintf(batch, "call Correr_FLUENT_%s.bat\n", nomeProjecto);
fprintf(batch, "echo Processo %s terminado. \n", nomeProjecto);

//Criar m-file para fazer Gráficos em Matlab
sprintf(lixo, "%s\\%s\\Graficos\\Graficos_Matlab_%s.m", path_original, nomeProjecto, nomeProjecto);
matlab=fopen(lixo, "w");
if(matlab==NULL)
{
    printf("Erro: Nao foi possivel criar o ficheiro '%s'!\n\n", lixo);
    return 0;
}

if(numeroIncidencias==1)

```

```

{
  if(numeroVelocidades==1)
  {
    fprintf(matlab,"%c Valores pontuais: nenhum gráfico para mostrar!\n",37);
    goto FimFicheiroMatlab;
  }
  else
  {
    fprintf(matlab,"load Incidencia.dat;\n");
    fprintf(matlab,"load Velocidade.dat;\n");
    fprintf(matlab,"load Corda.dat;\n");

    fprintf(matlab,"load Cl.dat;\n");
    fprintf(matlab,"figure\n");
    fprintf(matlab,"h%d = plot(Velocidade,Cl);\n",figura);
    fprintf(matlab,"xlabel('Velocidade')\n");
    fprintf(matlab,"ylabel('Cl')\n");
    fprintf(matlab,"title('%s')\n",nomeProjecto);
    sprintf(lixo,"legend('Incidencia=%.1f')\n",incidencia[0]);
    fprintf(matlab,lixo);
    fprintf(matlab,"saveas(gcf,['Cl'],'.jpg')\n");
    fprintf(matlab,"saveas(gcf,['Cl'],'.fig')\n\n");
    figura++;

    fprintf(matlab,"load Cd.dat;\n");
    fprintf(matlab,"figure\n");
    fprintf(matlab,"h%d = plot(Velocidade,Cd);\n",figura);
    fprintf(matlab,"xlabel('Velocidade')\n");
    fprintf(matlab,"ylabel('Cd')\n");
    fprintf(matlab,"title('%s')\n",nomeProjecto);
    sprintf(lixo,"legend('Incidencia=%.1f')\n",incidencia[0]);
    fprintf(matlab,lixo);
    fprintf(matlab,"saveas(gcf,['Cd'],'.jpg')\n");
    fprintf(matlab,"saveas(gcf,['Cd'],'.fig')\n\n");
    figura++;

    fprintf(matlab,"load Cm.dat;\n");
    fprintf(matlab,"figure\n");
    fprintf(matlab,"h%d = plot(Velocidade,Cm);\n",figura);
  }
}

```

```

fprintf(matlab, "xlabel('Velocidade')\n");
fprintf(matlab, "ylabel('Cm')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
sprintf(lixo, "legend('Incidencia=%.1f')\n", incidencia[0]);
fprintf(matlab, lixo);
fprintf(matlab, "saveas(gcf, ['Cm', '.jpg'])\n");
fprintf(matlab, "saveas(gcf, ['Cm', '.fig'])\n\n");
figura++;

fprintf(matlab, "load Eficiencia.dat\n");
fprintf(matlab, "figure\n");
fprintf(matlab, "h%d = plot(Velocidade, Eficiencia);\n", figura);
fprintf(matlab, "xlabel('Velocidade')\n");
fprintf(matlab, "ylabel('Eficiencia')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
sprintf(lixo, "legend('Incidencia=%.1f')\n", incidencia[0]);
fprintf(matlab, lixo);
fprintf(matlab, "saveas(gcf, ['Eficiencia', '.jpg'])\n");
fprintf(matlab, "saveas(gcf, ['Eficiencia', '.fig'])\n\n");
figura++;

for(indice=0; indice<3; indice++)
{
    for(i=0; i<3; i++)
    {
        fprintf(matlab, "load %s_%s.dat\n", grandeza[indice], tipo[i]);
        fprintf(matlab, "figure\n");
        fprintf(matlab, "h%d = plot(Velocidade, %s_%s);\n", figura, grandeza[indice], tipo[i]);
        fprintf(matlab, "xlabel('Velocidade')\n");
        fprintf(matlab, "ylabel('%s-%s')\n", grandeza[indice], tipo[i]);
        fprintf(matlab, "title('%s')\n", nomeProjecto);
        sprintf(lixo, "legend('Incidencia=%.1f')\n", incidencia[0]);
        fprintf(matlab, lixo);
        fprintf(matlab, "saveas(gcf, [%s_%s, '.jpg'])\n", grandeza[indice], tipo[i]);
        fprintf(matlab, "saveas(gcf, [%s_%s, '.fig'])\n\n", grandeza[indice], tipo[i]);
        figura++;
    }
}
for(i=3; i<5; i++)

```

```

    {
        fprintf(matlab, "load %s.dat;\n", grandeza[i]);
        fprintf(matlab, "figure\n");
        fprintf(matlab, "h%d = plot(Velocidade,%s);\n", figura, grandeza[i]);
        fprintf(matlab, "xlabel('Velocidade')\n");
        fprintf(matlab, "ylabel('%s')\n", grandeza[i]);
        fprintf(matlab, "title('%s')\n", nomeProjecto);
        sprintf(lixo, "legend('Incidencia=%.1f')\n", incidencia[0]);
        fprintf(matlab, lixo);
        fprintf(matlab, "saveas(gcf,['%s','.jpg']);\n", grandeza[i]);
        fprintf(matlab, "saveas(gcf,['%s','.fig']);\n\n", grandeza[i]);
        figura++;
    }
}
else
{
    fprintf(matlab, "load Incidencia.dat;\n");
    fprintf(matlab, "load Velocidade.dat;\n");
    fprintf(matlab, "load Corda.dat;\n");

    if(numeroVelocidades==1)
    {
        fprintf(matlab, "load Cl.dat;\n");
        fprintf(matlab, "figure\n");
        fprintf(matlab, "h%d=plot(Incidencia,Cl);\n", figura);
        fprintf(matlab, "xlabel('Incidencia')\n");
        fprintf(matlab, "ylabel('Cl')\n");
        fprintf(matlab, "title('%s')\n", nomeProjecto);
        sprintf(lixo, "legend('Velocidade=%.1f')\n", velocidadeU[0]);
        fprintf(matlab, lixo);
        fprintf(matlab, "saveas(gcf,['Cl','.jpg']);\n");
        fprintf(matlab, "saveas(gcf,['Cl','.fig']);\n\n");
        figura++;

        fprintf(matlab, "load Cd.dat;\n");
        fprintf(matlab, "figure\n");
        fprintf(matlab, "h%d=plot(Incidencia,Cd);\n", figura);
        fprintf(matlab, "xlabel('Incidencia')\n");
    }
}

```

```

fprintf(matlab, "ylabel('Cd')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
sprintf(lixo, "legend('Velocidade=%.1f')\n", velocidadeU[0]);
fprintf(matlab, lixo);
fprintf(matlab, "saveas(gcf, ['Cd', '.jpg']);\n");
fprintf(matlab, "saveas(gcf, ['Cd', '.fig']);\n\n");
figura++;

fprintf(matlab, "load Cm.dat;\n");
fprintf(matlab, "figure\n");
fprintf(matlab, "h%d=plot(Incidencia, Cm);\n", figura);
fprintf(matlab, "xlabel('Incidencia')\n");
fprintf(matlab, "ylabel('Cm')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
sprintf(lixo, "legend('Velocidade=%.1f')\n", velocidadeU[0]);
fprintf(matlab, lixo);
fprintf(matlab, "saveas(gcf, ['Cm', '.jpg']);\n");
fprintf(matlab, "saveas(gcf, ['Cm', '.fig']);\n\n");
figura++;

fprintf(matlab, "load Eficiencia.dat;\n");
fprintf(matlab, "figure\n");
fprintf(matlab, "h%d=plot(Incidencia, Eficiencia);\n", figura);
fprintf(matlab, "xlabel('Incidencia')\n");
fprintf(matlab, "ylabel('Eficiencia')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
sprintf(lixo, "legend('Velocidade=%.1f')\n", velocidadeU[0]);
fprintf(matlab, lixo);
fprintf(matlab, "saveas(gcf, ['Eficiencia', '.jpg']);\n");
fprintf(matlab, "saveas(gcf, ['Eficiencia', '.fig']);\n\n");
figura++;

for(indice=0; indice<3; indice++)
{
    for(i=0; i<3; i++)
    {
        fprintf(matlab, "load %s_%s.dat;\n", grandeza[indice], tipo[i]);
        fprintf(matlab, "figure\n");
        fprintf(matlab, "h%d=plot(Incidencia, %s_%s);\n", figura, grandeza[indice], tipo[i]);
    }
}

```

```

        fprintf(matlab, "xlabel('Incidencia')\n");
        fprintf(matlab, "ylabel('%s-%s')\n", grandeza[indice], tipo[i]);
        fprintf(matlab, "title('%s')\n", nomeProjecto);
        sprintf(lixo, "legend('Velocidade=%.1f')\n", velocidadeU[0]);
        fprintf(matlab, lixo);
        fprintf(matlab, "saveas(gcf, ['%s_%s', '.jpg'])\n", grandeza[indice], tipo[i]);
        fprintf(matlab, "saveas(gcf, ['%s_%s', '.fig'])\n\n", grandeza[indice], tipo[i]);
        figura++;
    }
}
for(i=3; i<5; i++)
{
    fprintf(matlab, "load %s.dat;\n", grandeza[i]);
    fprintf(matlab, "figure\n");
    fprintf(matlab, "h%d=plot('Incidencia, %s')\n", figura, grandeza[i]);
    fprintf(matlab, "xlabel('Incidencia')\n");
    fprintf(matlab, "ylabel('%s')\n", grandeza[i]);
    fprintf(matlab, "title('%s')\n", nomeProjecto);
    sprintf(lixo, "legend('Velocidade=%.1f')\n", velocidadeU[0]);
    fprintf(matlab, lixo);
    fprintf(matlab, "saveas(gcf, ['%s', '.jpg'])\n", grandeza[i]);
    fprintf(matlab, "saveas(gcf, ['%s', '.fig'])\n\n", grandeza[i]);
    figura++;
}
}
else
{
    fprintf(matlab, "load Cl.dat;\n");
    fprintf(matlab, "figure\n");
    fprintf(matlab, "h%d=surfc('Incidencia, Velocidade, Cl')\n", figura);
    fprintf(matlab, "xlabel('Incidencia')\n");
    fprintf(matlab, "ylabel('Velocidade')\n");
    fprintf(matlab, "zlabel('Cl')\n", grandeza[indice], tipo[i]);
    fprintf(matlab, "title('%s')\n", nomeProjecto);
    fprintf(matlab, "shading interp\n");
    fprintf(matlab, "colorbar\n");
    fprintf(matlab, "saveas(gcf, ['Cl', '.jpg'])\n");
    fprintf(matlab, "saveas(gcf, ['Cl', '.fig'])\n\n");
    figura++;
}
}

```

```

fprintf(matlab, "load Cd.dat;\n");
fprintf(matlab, "figure\n");
fprintf(matlab, "h%d=surfc(Incidencia, Velocidade, Cd);\n", figura);
fprintf(matlab, "xlabel('Incidencia')\n");
fprintf(matlab, "ylabel('Velocidade')\n");
fprintf(matlab, "zlabel('Cd')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
fprintf(matlab, "shading interp\n");
fprintf(matlab, "colorbar\n");
fprintf(matlab, "saveas(gcf, ['Cd', '.jpg']);\n");
fprintf(matlab, "saveas(gcf, ['Cd', '.fig']);\n\n");
figura++;

fprintf(matlab, "load Cm.dat;\n");
fprintf(matlab, "figure\n");
fprintf(matlab, "h%d=surfc(Incidencia, Velocidade, Cm);\n", figura);
fprintf(matlab, "xlabel('Incidencia')\n");
fprintf(matlab, "ylabel('Velocidade')\n");
fprintf(matlab, "zlabel('Cm')\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
fprintf(matlab, "shading interp\n");
fprintf(matlab, "colorbar\n");
fprintf(matlab, "saveas(gcf, ['Cm', '.jpg']);\n");
fprintf(matlab, "saveas(gcf, ['Cm', '.fig']);\n\n");
figura++;

fprintf(matlab, "load Eficiencia.dat;\n");
fprintf(matlab, "figure\n");
fprintf(matlab, "h%d=surfc(Incidencia, Velocidade, Eficiencia);\n", figura);
fprintf(matlab, "xlabel('Incidencia')\n");
fprintf(matlab, "ylabel('Velocidade')\n");
fprintf(matlab, "zlabel('Eficiencia')\n");
fprintf(matlab, "shading interp\n");
fprintf(matlab, "colorbar\n");
fprintf(matlab, "title('%s')\n", nomeProjecto);
fprintf(matlab, "saveas(gcf, ['Eficiencia', '.jpg']);\n");
fprintf(matlab, "saveas(gcf, ['Eficiencia', '.fig']);\n\n");
figura++;

```

```

for(indice=0;indice<3;indice++)
{
    for(i=0;i<3;i++)
    {
        fprintf(matlab,"load %s_%s.dat;\n",grandeza[indice],tipo[i]);
        fprintf(matlab,"figure\n");
        fprintf(matlab,"h%d=surfc(Incidencia,Velocidade,%s_%s);\n",figura,grandeza[indice],tipo[i]);
        fprintf(matlab,"xlabel('Incidencia')\n");
        fprintf(matlab,"ylabel('Velocidade')\n");
        fprintf(matlab,"zlabel('%s-%s')\n",grandeza[indice],tipo[i]);
        fprintf(matlab,"shading interp\n");
        fprintf(matlab,"colorbar\n");
        fprintf(matlab,"title('%s')\n",nomeProjecto);
        fprintf(matlab,"saveas(gcf,['s_%s','.jpg']);\n",grandeza[indice],tipo[i]);
        fprintf(matlab,"saveas(gcf,['s_%s','.fig']);\n\n",grandeza[indice],tipo[i]);
        figura++;
    }
}
for(i=3;i<5;i++)
{
    fprintf(matlab,"load %s.dat;\n",grandeza[i]);
    fprintf(matlab,"figure\n");
    fprintf(matlab,"h%d=surfc(Incidencia,Velocidade,%s);\n",figura,grandeza[i]);
    fprintf(matlab,"xlabel('Incidencia')\n");
    fprintf(matlab,"ylabel('Velocidade')\n");
    fprintf(matlab,"zlabel('%s')\n",grandeza[i]);
    fprintf(matlab,"title('%s')\n",nomeProjecto);
    fprintf(matlab,"shading interp\n");
    fprintf(matlab,"colorbar\n");
    fprintf(matlab,"saveas(gcf,['s','.jpg']);\n",grandeza[i]);
    fprintf(matlab,"saveas(gcf,['s','.fig']);\n\n",grandeza[i]);
    figura++;
}
}
}

```

FimFicheiroMatlab:

```

//fprintf(matlab,"%\nexit");
fclose(matlab);

```

```
    fprintf(batch, "cd \"%s\"\n", path_original);
    fprintf(batch, "call Plot_Graficos.exe %s\n", nomeProjecto);
    fprintf(batch, "call Calcula_Coeficientes %s\n", nomeProjecto);
    fprintf(batch, "cd \"%s\\%s\\Graficos\"\n", path_original, nomeProjecto);
    fprintf(batch, "call \"%s\" -nosplash -nodesktop -minimize -r Graficos_Matlab_%s\n", caminhoMATLAB, nomeProjecto);
    fprintf(batch, "cd \"%s\"\n", path_original);

    free(incidencia);
    free(velocidadeU);
}

fprintf(batch, "echo Calculos terminados com sucesso!\n");
fclose(batch);
fclose(encargos);
fclose(config);
printf("\nEscrita do ficheiro concluida com sucesso!\n\n");
return 0;
}
```