Description

Objectif

L'intérêt du cas test est de vérifier le calcul TRUST pour la conduction de chaleur anisotrope dans une géométrie assimilé de la couche de diffusion de gaz de PEMFC.

Physique

La conduction de la chaleur dans GDL (pemfc) est anisotrope. Le coefficient de conductivité dans le plan (parallèle) de GDL est beaucoup plus élevé que celui perpendiculaire au plan (dans l'ordre de grandeur d'une centaine). De plus, la conduction est réduit lors de l'écrasement du GDL: zone non écrasé est plus conductrice que la zone écrasée. La conductivité dans la zone transitoire est interpolé linéairement.

Géométrie La géométrie utilisée est modélisé d'une partie de GDL écrasé par la plaque bipolaire (en métal) comme montrant la Figure 1.

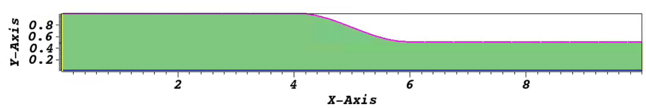

Figure 1: Géometrie de GDL (2D)

La dimension est la suivante:

- largeur (horizontale, Ox) 10
- épaisseur (vertical, Oy):
  - zone non écrasé: épaisseur à 1
  - zone écrasé: épaisseur à 0.5

Quatre bords sont définie: Gauche et Droit (jaune), Haut (violet) et Bas (bleue)

Modèle mathématique On considère la Conduction de la Chaleur dont l'équation comprend une opérateur de diffusion avec le coefficient anisotrope matriciel:

$$div(-Dgrad(T)) = q \tag{1}$$

avec:
T température K
q source par e.x. densité du flux de chaleur W/m^2

D coefficient de conductivité, $W/m/K$. Dans le cas d'anisotropie, il s'agit d'une matrice de taille 3x3 en tridimensionnel et 2x2 en bidimensionnel. La forme de matrice D dépend de la zone sur la géométrie:

- zone sur laquelle le bord est droit, non incliné: D est diagnonale
  $D = \begin{pmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{pmatrix}$ avec $D_{xx} = D_{zz} >> D_{yy}$
- zone dont le bord est courbé et incliné: D est pleine symétrique
  $D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ - & Dyy & D_{yz} \\ - & - & D_{zz} \end{pmatrix}$ avec $D_{diagonal} >> D_{extra-diagnonal}$
  La matrix D dans cette zone est le produit matriciel $D = M.D_{diagonal}.M^{-1}$ avec M la matrice de rotation (unitaire) en fonction de l'angle d'inclination.
  - en 2D:
    $M = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$
  - en 3D
    $M = \begin{bmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{bmatrix} = [\vec{u}, \vec{v}, \vec{w}]$
    avec $\vec{v} = \nabla T(\theta)$ vecteur normal du plan de bord d'écrasement GDL,
    $\vec{u} = \vec{v} \times \vec{OY}$,
    $\vec{w} = \vec{v} \times \vec{u}$
    La propriétaire de la matrice est $M^{-1} = M^T$ et $\det M = 1$

- zone d'écrasement: la magnitude de D dépend linéairement aux valeurs de conductivité de la zone non écrasement et écrasement qui correspondent respectivement à l'épaisseur maximal et minimal de GDL.
  $\alpha = \frac{ep - ep_{ec}}{ep_{non\_ec} - ep_{ec}}$

  $D = \alpha D_{non\_ec} + (1 - \alpha) D_{ec}$

Condition limite

- adiabatique sur les bords Gauche et Haut

- dirichlet $T = 0$ sur le bord Bas

- température externe imposé $T = 1$ avec coefficient d'échange imposé mix

- zone non écrasement ($x = [0,4]$) h_imp égale à 0.

- zone écrasement ($x = [6,10]$) h_imp égale à 10.

- zone transitoire ($x = (4,6)$) h_imp est en fonction de x (continue) $10.0, 5(1. + \sin(0, 5\Pi(x - 5)))$

Maillage La discrétisation schéma utilise un non structuré maillage (VEF)

Sonde on crée deux sondes

- le champ température sur la ligne diagonale de la géométrie P1(0,1) et P2(10,0)
- et le flux sur le bord Haut

Data fichiers

Deux fichiers .data sont défini:

- gradT.data Pour estimer (par MEDCoupling) l'angle de rotation θ du bord Haut et interpolation cet angle dans la géométrie.
- PEMFC_2D_anisotrope_pac_withMEDCoupling.data Pour calcul la conduction en prenant en compte l'anisotropie.

Fichier gradT.data

```
# dimension #
dimension 2

# domaine #
domaine dom

# probleme de condution de la chaleur #
Pb_conduction pb

# BEGIN MESH #
Lire_med family_names_from_group_names dom Mesh_1 Mesh_1.med
# END MESH #

# discretisation VEF #
vefprep1b dis

# time scheme #
scheme_euler_implicit scheme

Read scheme
{
    tinit 0.
    tmax 10.
```

```
    dt_min 1e-7
    dt_max 1.
    dt_impr 1.
    dt_sauv 1.
    seuil_statio 1.e-6
    facsec 1e6
    facsec_max 1e8
    Solveur Implicite
    {
      Solveur petsc cholesky { quiet }
    }
}

# physics properties of medium  #
solide sol
read sol
{
    rho champ_uniforme 1 1.0
    lambda champ_uniforme 1 1.
    Cp champ_uniforme 1 1.0
}

# associate the problem with the geometry, the time scheme and the
    physic properties #
associate pb dom
associate pb scheme
associate pb sol

# meshing the geometry with the given discretisation method #
discretize pb dis

# definition of the problem #
read pb
{
  # Resolution div (-grad u) = 0 donc sans source, sans coefficient
    , steady state #
  conduction
  {
    diffusion { }
    initial_conditions {
      temperature champ_uniforme 1 1.
    }
    boundary_conditions {
      Gauche paroi_flux_impose champ_front_uniforme 1 0.
      Droit paroi_flux_impose champ_front_uniforme 1 0.
      Haut paroi_temperature_imposee champ_front_uniforme 1 1.
      Bas paroi_temperature_imposee champ_front_uniforme 1 0.
    }
  }

  # post traitement  #
  post_processing
  {
    # define the champs for exporting the coefficient anisotrope #
    definition_champs
    {
      # gradient of temperature grad(u) = [ux uy]^T #
```

```
      gradT gradient {
        source refchamp { Pb_champ pb temperature }
      }
    }

    format med_major fields dt_post 1.
    {
      temperature elem
      temperature som
      gradT elem
      gradT som
    }
  }
}

# solve the problem #
solve pb

end
```

Fichier PEMFC_2D_anisotrope_pac_withMEDCoupling.data

```
# PARALLEL OK #
# Heat Conduction 2D anisotrope #
dimension 2

# domain name #
domaine dom

#  physic problem: heat conduction  #
pb_conduction pb

# BEGIN MESH #
Lire_med family_names_from_group_names dom dom gradT_0000.med
# END MESH #

# BEGIN PARTITION
Partition dom
{
    /* Choose Nb_parts so to have ~ 25000 cells per processor */
  Partition_tool metis { nb_parts 2 }
  Larg_joint 2
  zones_name dom
}
End
END PARTITION #

# BEGIN SCATTER
Scatter dom.Zones dom
END SCATTER #

# discretisation VDF for the domain 2D #
vefprep1b dis

# time scheme #
Scheme_euler_implicit scheme
# schema_backward_differentiation_order_2 scheme #
Read scheme
```

```
{
  # Time step #
  # Initial time [s] #
  tinit 0
  # Min time step #
  # dt_min 1e-9 #
  # Max time step #
  # dt_max 1. #
  # facsec maximal possible value for the heat transfer #
  facsec 1000
  facsec_max 5000
  # .out files printing period #
  dt_impr 1e-3 # Note: small value to print at each time step #
  # .sauv files printing period #
  dt_sauv 1e-3
  # Stop if one of the following criteria is checked: #
  # End time [s] #
  tmax 10.0
  # Max number of time steps #
  # nb_pas_dt_max 10 #
  # max number of implicit solver #
  max_iter_implicite 50
  # Convergence threshold (see .dt_ev file) #
  seuil_statio_relatif_deconseille 1
  seuil_statio 1e-5
  Solveur Implicite
  {
      Solveur petsc cholesky { quiet }
  }
}

# physics properties of medium  #
solide sol
read sol
{
  rho champ_uniforme 4 1. 1. 1. 1.
  lambda champ_fonc_med last_time conduc2d_mod.med dom CONDUCTIVITY_ELEM_dom elem 0
  Cp champ_uniforme 4 1. 1. 1. 1.
}


# associate the physic problem with the geometry, the time scheme and the physic properties #
associate pb dom
associate pb scheme
associate pb sol

# meshing the geometry with the given discretisation method #
discretize pb dis

# definition of the physic problem: heat conduction #
read pb
{
  # Resolution of the laplacien equation modifying the heat equation #
  conduction
```

```
  {
    diffusion { Tensor }
    initial_conditions {
      temperature champ_uniforme 1 0.0
      # temperature champ_fonc_med last_time prepare_0000.med dom temperature elem 0 #
    }
    boundary_conditions {
      Gauche paroi_flux_impose champ_front_uniforme 1 0.
      Droit paroi_flux_impose champ_front_uniforme 1 0.
      Haut paroi_echange_externe_impose h_imp champ_front_fonc_xyz 1 (x[4.)*0.+(x>4.)*(x<6.)*10.*0.5*(1.+SIN(0.5*Pi*(x-5.)))+(x]6.)*(10.) T_ext champ_front_uniforme 1 1.
      Bas paroi_temperature_imposee champ_front_uniforme 1 0.
    }
  }

  # post traitement  #
  post_processing
  {
    definition_champs
    {
      # gradient of temperature grad(u) = [ux uy]^T #
      gradT gradient {
        source refchamp { Pb_champ pb temperature }
      }
    }
    probes {
      sonde_temperature temperature periode 0.01 segment 501 0. 1. 10. 0.
    }
    format lata     # lata for VisIt tool #
    fields dt_post 0.1 # warning small value #
    {
      temperature elem
      temperature som
      gradT som
      diffusivite_thermique elem
      capacite_calorifique elem
      masse_volumique elem
    }
  }
}

imprimer_flux dom { Haut Bas }

# solve the problem #
solve pb

end
```