

Modélisation numérique sous MATLAB

Par Rafic YOUNES

Enseignant - Chercheur à la Faculté de Génie – Université Libanaise
Responsable du DEA Mécanique E2M

R.Y. 02.04

1. Grandes lignes de la méthode des éléments finis
 - 1.1 Exposé de la démarche
 - 1.2 Un exemple détaillé
2. Utilisation du pdetool
3. Applications
 - 3.1 Application 1
 - 3.2 Application 2
 - 3.3 Application 3
 - 3.4 Application 4
 - 3.5 Application 5

Quelle place occupe le calcul dans l'industrie ? Quels sont les principaux champs d'application du calcul ?

Le calcul numérique permet à l'ingénieur d'**effectuer des simulations numériques de phénomènes physiques**. Le calcul occupe une place stratégique avec la CAO et les autres technologies de simulation (essais) dans le développement d'un produit complexe qui touche à différents domaines de la physique. Cela concerne les **industries automobiles, navales, aéronautiques, ferroviaires, mais aussi les industries lourdes: centrales électriques, plates-formes pétrolières, et le génie civil**.

Le calcul est indispensable lorsque l'on cherche à obtenir une solution optimisée pour réduire les coûts et les délais de fabrication. Grâce au calcul, même simplifié, l'ingénieur peut tester plusieurs configurations pour optimiser le comportement d'un modèle à une prestation donnée. **Cela évite de multiplier les prototypes et les essais tests réels**, les supports physiques ne servent plus à chercher une solution, ils permettent de la valider.

Le calcul s'applique aussi dans les domaines du « process ». Les procédés de fabrication tels que l'emboutissage, l'usinage grande vitesse, les dépôts de peinture, l'assemblage de tôlerie, la mise en forme des plastiques, peuvent être modélisés par éléments finis. Ici c'est une bonne représentation du comportement du phénomène physique qui sera recherchée pour **pouvoir vérifier et valider un procédé de fabrication d'une pièce**.

Enfin **le calcul de conception dans les bureaux d'études**, c'est sans doute le plus répandu car grâce aux outils de calcul simplifié dont disposent les logiciels de CAO modernes, **la simulation numérique fait partie des outils de conception pour obtenir un comportement défini a priori qui détermine le dimensionnement, donc le dessin, des pièces mécaniques**.

La méthode des éléments finis est de toutes les méthodes de discrétisation la plus utilisée car :

A) elle peut traiter des problèmes de géométrie complexe. B) elle couvre de nombreux domaines de la physique. C) les moyens informatiques actuels (puissance des calculateurs, outils de visualisation) la rende facile de mise en oeuvre. D) de nombreux logiciels généraux ou dédiés sont disponibles sur le marché, le tableau ci-dessous vous permettra de visiter les sites Internet proposés par les principaux fournisseurs de logiciel éléments finis.

Pdetool de MatLab, Abaqus, I-deas, Adina, Marc, Algor, Radioss, Ansys, Samcef, Catia, Systus

L'objectif de ce cours est de vous donner les bases essentielles sur lesquelles repose la méthode des éléments finis. Ensuite sont présentés quelques applications utilisant le pdetool de MatLab.

1 Exposé de la démarche :

La méthode consiste à rechercher une solution *approchée* de la solution exacte sous la forme d'un champ $\mathbf{F}(\mathbf{M},\mathbf{t})$ défini par morceaux sur des sous domaines de Ω . Les n sous domaines Ω_i doivent être tels que

$$\bigcup_{i=1}^n \Omega_i = \Omega \text{ et } \tilde{\Omega}_i \cap \tilde{\Omega}_j = \emptyset \quad \forall i \neq j$$

où $\tilde{\Omega}_i$ désigne l'intérieur de Ω_i . Autrement dit, les Ω_i sont une partition de Ω .

Les champs $\tilde{\mathbf{f}}(\mathbf{M},\mathbf{t})$, définis sur chaque sous domaine sont des champs choisis parmi une famille *arbitraire* de champs (généralement polynômes). La famille de champs locaux est appelée *espace des fonctions d'interpolation de l'élément*.

La famille de champs globaux $\tilde{\mathbf{F}}(\mathbf{M},\mathbf{t})$, obtenus par juxtaposition des champs locaux est appelée *espace des fonctions d'interpolation du domaine Ω* .

Le champ dans chaque sous domaine Ω_i est déterminé par un nombre fini de valeurs du champ (ou de valeurs de ses dérivées) en des points choisis arbitrairement dans le sous domaine, et appelés *noeuds*. Le champ local est une *interpolation* entre les valeurs aux noeuds. Le sous domaine muni de son interpolation est appelé *élément*.

Chercher une solution par éléments finis consiste donc à déterminer quel champ local on attribue à chaque sous domaine pour que le champ global $\tilde{\mathbf{F}}(\mathbf{M},\mathbf{t})$ obtenu par juxtaposition de ces champs locaux soit proche de la solution du problème.

Parmi les contraintes qu'on impose à la solution approchée cherchée, il y a souvent au moins une continuité simple (C_0) à la frontière entre les sous domaines.

La figure 1 montre une solution approchée discontinue d'un champ scalaire sur un domaine Ω de dimension 1. La famille de champs locaux est la famille des champs constants par morceaux.

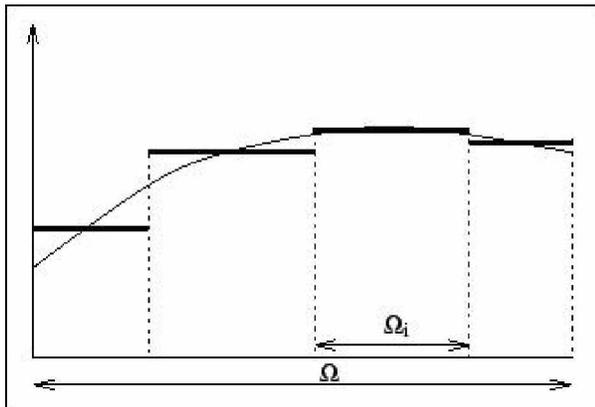


Figure 1: Solution approchée discontinue

La figure 2 montre une solution approchée continue C_0 d'un champ scalaire sur un domaine Ω de dimension 1. La famille de champs locaux est la famille des champs polynômes de degré 1.

La figure 3 montre une solution approchée continue C_1 d'un champ scalaire sur un domaine Ω de dimension 1. La famille de champs locaux est la famille des champs polynômes de degré 3.

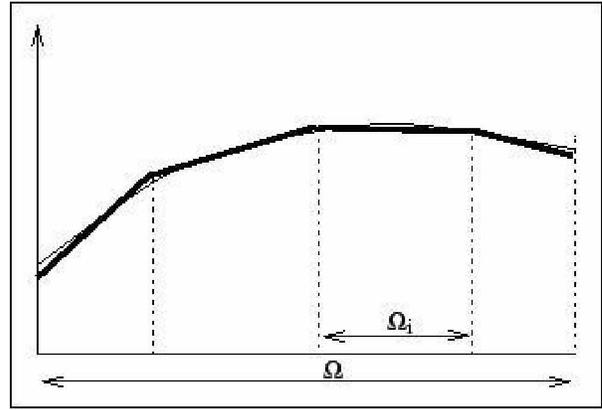


Figure 2: Solution approchée continue C_0

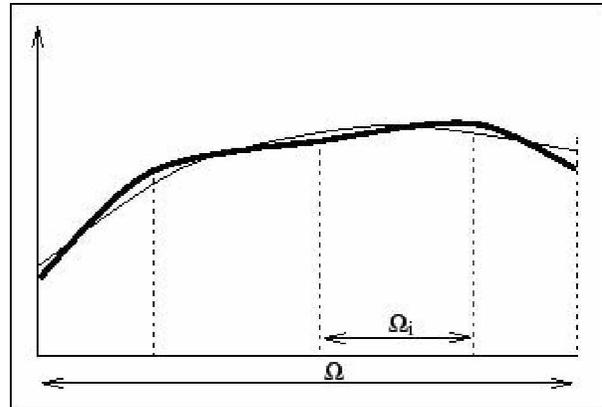


Figure 3: Solution approchée continue C_1

La qualité de la solution approchée dépend de la division en sous domaines (nombre et dimensions des sous domaines), du choix de la famille de champs locaux dans chaque sous domaine, et des conditions de continuité qu'on impose aux frontières des sous domaines (C_0, C_1, \dots). Une fois ces choix faits, il reste à rechercher, une combinaison de champs locaux qui satisfait approximativement les équations.

Pour résoudre un problème par la méthode des éléments finis, on procède donc par étapes successives :

1 - On se pose un problème physique sous la forme d'une équation différentielle ou aux dérivées partielles à satisfaire en tout point d'un domaine Ω , avec des conditions aux limites sur le bord $\partial\Omega$ nécessaires et suffisantes pour que la solution soit unique.

2 - On construit une formulation intégrale du système différentiel à résoudre et de ses conditions aux limites: C'est la *formulation variationnelle du problème*.

3 - On divise Ω en sous domaines : C'est le *maillage*. Les sous domaines sont appelés *mailles*.

4 - On choisit la famille de champs locaux, c'est à dire à la fois la position des noeuds dans les sous domaines et les polynômes (ou autres fonctions) qui définissent le champ local en fonction des valeurs aux noeuds (et éventuellement des dérivées). La maille complétée par ces informations est alors appelée *élément*.

5 - On ramène le problème à un problème discret : C'est la *discrétisation*. En effet, toute solution approchée est complètement déterminée par les valeurs aux noeuds des éléments. Il suffit donc de trouver les valeurs à attribuer aux noeuds pour décrire une solution approchée. Le problème fondamental de la méthode des éléments finis peut se résumer en deux questions :

(a) Comment choisir le problème discret dont la solution est <<proche>> de la solution exacte?

(b) Quelle signification donner au mot <<proche>> ?

6 - On résout le problème discret: C'est la *résolution*

7 - On peut alors construire la solution approchée à partir des valeurs trouvées aux noeuds et en déduire d'autres grandeurs : C'est le *post-traitement*.

8 - On visualise et on exploite la solution pour juger de sa qualité numérique et juger si elle satisfait les critères du cahier des charges : C'est *l'exploitation des résultats*.

Les étapes 1, 2, 3, 4 et 5 sont souvent rassemblées sous le nom de *prétraitement*.

Le travail de ces différentes étapes est assisté par les logiciels. Pour maîtriser leur utilisation, il est indispensable de comprendre les fondements de la méthode, notamment les phases 3 et 4, ne serait-ce que pour comprendre et choisir intelligemment parmi les options qu'ils proposent.

2 UN EXEMPLE DETAILLE :

On se propose de rechercher une solution approchée du problème suivant :

Trouver $f(x)$ dans le domaine $\Omega = [0, 1]$ satisfaisant l'équation différentielle :

$$\frac{d^2 f(x)}{dx^2} + 2 \frac{df(x)}{dx} - 3x = 0$$

avec les conditions aux frontières de Ω :

$$f(0) = 0 \text{ et } \left[\frac{df(x)}{dx} \right]_{x=1} = 0$$

Dans ce problème, Ω est un domaine de dimension 1 et le temps n'apparaît pas. Pour décrire le domaine, on n'a besoin que d'une seule variable x . L'équation à résoudre est donc une équation différentielle ordinaire. La frontière $\partial\Omega$ se réduit à deux points.

La solution exacte de ce problème est

$$f(x) = -\frac{3}{4}x + \frac{3}{4}x^2 - \frac{3}{8}e^2 + \frac{3}{8}e^{2-2x}$$

Son graphe est donné figure 6 page 8.

2.1 Choix du maillage

On divise *arbitrairement* Ω en trois mailles de même taille (voir figure 4):

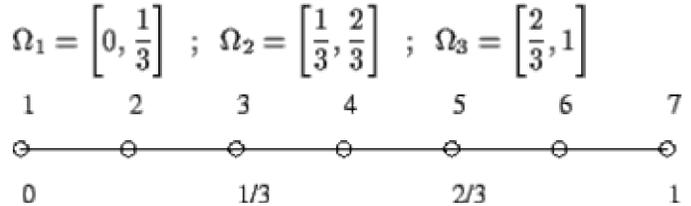


Figure 4: Maillage du problème

2.2 Choix des noeuds et des champs locaux

On décide de prendre des éléments à 3 noeuds, et pour la famille de champs locaux des polynômes de degré 2. Les noeuds sont choisis aux extrémités et au milieu de chaque maille.

On peut alors déterminer chaque champ local en fonction des valeurs aux 3 noeuds.

Remarquer que le fait d'avoir utilisé des noeuds aux extrémités de chaque élément présente deux avantages :

1 - Le nombre de noeuds est réduit, car il y a des noeuds communs à deux éléments.

2 - On assure ainsi une continuité C_0 de la solution approchée : les champs locaux de deux éléments voisins auront la même valeur à leur noeud commun.

Remarquer encore qu'il n'est pas nécessaire de prendre les éléments identiques : on aurait pu prendre certains à deux noeuds et d'autres à trois noeuds.

Chaque champ local peut donc s'exprimer en fonction des valeurs aux noeuds. En effet, il n'existe qu'un seul polynôme du second degré qui satisfasse les conditions suivantes :

$$a_1 x_1^2 + b_1 x_1 + c_1 = f_1$$

$$a_1 x_2^2 + b_1 x_2 + c_1 = f_2$$

$$a_1 x_3^2 + b_1 x_3 + c_1 = f_3$$

où x_1 , x_2 et x_3 sont les coordonnées des 3 noeuds de l'élément, et f_1 , f_2 et f_3 sont des valeurs aux 3 noeuds de l'élément.

On obtient :

$$a_1 = \frac{x_2 f_1 - x_2 f_3 + x_1 f_3 - f_1 x_3 - f_2 x_1 + f_2 x_3}{(x_2 - x_3)(x_1 - x_3)(x_1 - x_2)} \quad (2.1)$$

$$b_1 = -\frac{x_1^2 f_3 - x_1^2 f_2 + x_3^2 f_2 - f_1 x_3^2 - f_3 x_2^2 + f_1 x_2^2}{(x_2 - x_3)(x_1 - x_3)(x_1 - x_2)} \quad (2.2)$$

$$c_1 = \frac{x_1^2 x_2 f_3 - x_1^2 f_2 x_3 - f_3 x_2^2 x_1 + x_3^2 f_2 x_1 + f_1 x_2^2 x_3 - x_3^2 x_2 f_1}{(x_2 - x_3)(x_1 - x_3)(x_1 - x_2)} \quad (2.3)$$

Le polynôme avec les coefficients a_1 , b_1 et c_1 ci-dessus est appelé *fonction d'interpolation* de l'élément 1.

Les polynômes d'interpolation des 2 autres sont de la même forme : Pour le second, on remplace les indices 1, 2, 3 par les indices 3, 4, 5 ; et pour le troisième, on remplace les indices 1, 2, 3 par les indices 5, 6, 7.

Dans cet exemple, on a 7 noeuds. Si on donne une valeur arbitraire à chaque noeud, les fonctions d'interpolation sont déterminées et définissent par morceaux une fonction \tilde{f} continue C_0 sur le domaine Ω . L'espace des fonctions \tilde{F} , définies en 3 morceaux, est de dimension 7.

Les trois fonctions d'interpolation sur les 3 éléments sont :

$$\tilde{f}_1 = (18 f_3 - 36 f_2 + 18 f_1) x^2 + (-3 f_3 + 12 f_2 - 9 f_1) x + f_1 \quad (2.4)$$

$$\tilde{f}_2 = (18 f_5 - 36 f_4 + 18 f_3) x^2 + (-15 f_5 + 36 f_4 - 21 f_3) x + 3 f_5 - 8 f_4 + 6 f_3 \quad (2.5)$$

$$\tilde{f}_3 = (18 f_7 - 36 f_6 + 18 f_5) x^2 + (-27 f_7 + 60 f_6 - 33 f_5) x + 10 f_7 - 24 f_6 + 15 f_5 \quad (2.6)$$

2.3 Formulation variationnelle du problème

On montre en analyse fonctionnelle que

$$\int_{\Omega} \psi(x) h(x) dx = 0 \quad \forall \psi(x) \iff h(x) = 0 \quad \forall x \in \Omega$$

où $\psi(x)$ et $h(x)$ et Ω doivent satisfaire certaines conditions de régularité qu'on n'explicitera pas ici.

Autrement dit, résoudre l'équation différentielle est équivalent à chercher $f(x)$ tel que :

$$\int_{\Omega} \psi(x) \left(\frac{d^2 f(x)}{dx^2} + 2 \frac{df(x)}{dx} - 3x \right) dx = 0 \quad \forall \psi(x) \quad (2.7)$$

$$\text{avec } f(0) = 0 \text{ et } \left[\frac{df(x)}{dx} \right]_{x=1} = 0 \quad (2.8)$$

Cette formulation est une *formulation variationnelle* (on dit aussi *formulation intégrale* ou encore *formulation faible*) du problème.

On obtient une autre formulation variationnelle en utilisant l'intégration par parties :

$$\int_0^1 \psi(x) \frac{d^2 f(x)}{dx^2} dx = - \int_0^1 \frac{d\psi(x)}{dx} \frac{df(x)}{dx} dx + \left[\psi(x) \frac{df(x)}{dx} \right]_0^1$$

$$\int_0^1 \psi(x) \frac{df(x)}{dx} dx = - \int_0^1 \frac{d\psi(x)}{dx} f(x) dx + [\psi(x) f(x)]_0^1$$

Compte tenu des conditions aux limites, $f(0)=0$ et

$$\left[\frac{df(x)}{dx} \right]_{x=1} = 0, \text{ une autre formulation variationnelle}$$

du problème est donc : Trouver $f(x)$ tel que

$$- \int_0^1 \frac{d\psi(x)}{dx} \frac{df(x)}{dx} dx + 0 - \psi(0) \left[\frac{df(x)}{dx} \right]_{x=0} - 2 \int_0^1 \frac{d\psi(x)}{dx} f(x) dx + 2\psi(1)f(1) - 0 - 3 \int_0^1 \psi(x) x dx = 0 \quad \forall \psi(x) \quad (2.9)$$

Il est important de noter que dans les termes de bord de cette dernière formulation, les conditions aux limites sont prises en compte. En comparant les deux formulations variationnelles (2.7) et (2.9), on constate

que, dans la seconde forme, la fonction inconnue $f(x)$ intervient avec des dérivées d'ordre inférieur, et que par contre les dérivées des fonctions $\psi(x)$ apparaissent. L'équivalence entre ces deux formulations variationnelles et le problème initial est soumise à des conditions sur $f(x)$, $\psi(x)$ et Ω précisées par l'analyse fonctionnelle et qu'on supposera satisfaites.

Remarque :

La démarche suivie ici pour construire une formulation variationnelle du problème est purement mathématique. Il arrive souvent que la formulation variationnelle ait une interprétation physique. On peut donc aussi les établir par des raisonnements physiques.

2.4 Discrétisation du problème

Le principe de l'approximation par éléments finis est le suivant :

On choisit une formulation variationnelle (par exemple (2.9) et on cherche à satisfaire l'équation (2.9) avec les \tilde{f} déterminés par les valeurs aux noeuds définis précédemment. Il va de soit que ce problème, n'a généralement pas de solution si on garde la condition $\langle \langle \nabla \psi_i(\mathbf{x}) \rangle \rangle$ (sinon \tilde{f} serait une solution exacte !). On ne vérifiera donc cette équation *que pour certains* $\psi_i(\mathbf{x})$ *seulement*.

À chaque $\psi_i(\mathbf{x})$ correspondra une équation scalaire fonction des valeurs aux noeuds. Il suffit de les choisir indépendants et en nombre égal au nombre d'inconnues. Dans notre exemple, il en faut 7 pour déterminer les valeurs aux noeuds f_1, \dots, f_7 . Les fonctions $\psi(\mathbf{x})$ sont appelées *fonctions test* ou encore *fonctions de pondération*. On est ainsi ramené à un système algébrique de 7 équations à 7 inconnues.

Il y a une infinité de manières de choisir les 7 $\psi_i(\mathbf{x})$. Elles engendrent différentes variantes de la méthode des éléments finis. Chacune aboutit à une solution approchée différente. La seule condition impérative est que le choix des ψ_i doit conduire à un système algébrique à solution unique. Des théorèmes d'analyse numérique suggèrent certains choix, pour lesquels la solution approchée \tilde{f} est garantie unique et convergente vers la solution exacte ^{2.9} quand le maillage se raffine. Dans notre exemple, si on choisit la formulation (2.9), les fonctions $\psi_i(\mathbf{x})$ doivent avoir une dérivée non nulle partout, pour que la formulation ne soit pas triviale. Par contre, dans cette formulation, la dérivée seconde de la fonction $f(\mathbf{x})$, qui sera remplacée par l'approximation $\tilde{f}(\mathbf{x})$, n'apparaît plus. On pouvait donc choisir une famille d'approximation $\tilde{f}(\mathbf{x})$ plus simple.

On choisit par exemple les 7 fonctions test $\psi_1, \psi_2, \dots, \psi_7$ définies par la figure 5.

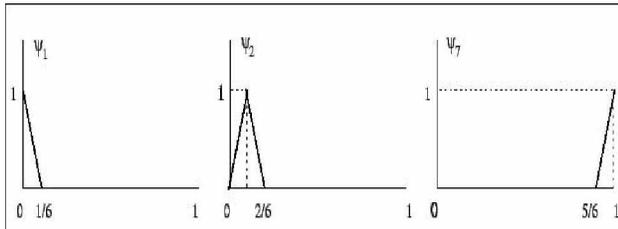


Figure 5: Les sept fonctions $\psi_i(\mathbf{x})$.

Leurs dérivées sont constantes (+6 ou -6) ou nulles et on a $\psi_i(\mathbf{x}_j) = \delta_{ij}$

Pour chaque fonction test ψ_i , on écrit une équation scalaire:

$$-\int_0^1 \frac{d\psi_i(x)}{dx} \frac{d\tilde{f}(x)}{dx} dx - \psi_i(0) \left[\frac{d\tilde{f}(x)}{dx} \right]_{x=0} - 2 \int_0^1 \frac{d\psi_i(x)}{dx} \tilde{f}(x) dx + 2\psi_i(1) \tilde{f}(1) - 3 \int_0^1 \psi_i(x) x dx = 0 \text{ pour } i = 1..7$$

où les $\tilde{\mathbf{f}}(\mathbf{x})$ sont des fonctions de \mathbf{x} et des valeurs aux noeuds $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_7$. Les intégrales de ces 7 équations sont faciles à calculer : elles se réduisent à des intégrales sur le segment où les dérivées de ψ_i sont non nulles.

La construction de ce système d'équations s'appelle l'assemblage. Pour un choix différent de fonctions $\psi_i(\mathbf{x})$, les intégrales sur chaque élément sont moins simples à calculer.

Par exemple, la première équation (avec ψ_1) est :

$$-\int_0^{1/6} \frac{d\psi_1(x)}{dx} \frac{d\tilde{f}(x)}{dx} dx - \psi_1(0) \left[\frac{d\tilde{f}(x)}{dx} \right]_{x=0} - 2 \int_0^{1/6} \frac{d\psi_1(x)}{dx} \tilde{f}(x) dx + 2\psi_1(1) \tilde{f}(1) - 3 \int_0^{1/6} \psi_1(x) x dx = 0$$

soit encore :

$$-\int_0^{1/6} (-6) \frac{d\tilde{f}(x)}{dx} dx - \left[\frac{d\tilde{f}(x)}{dx} \right]_{x=0} - 2 \int_0^{1/6} (-6) \tilde{f}(x) dx - 3 \int_0^{1/6} (1-6x)x dx = 0$$

où $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}^2 + \mathbf{b} \cdot \mathbf{x} + \mathbf{c}$ dans le segment $\left[0, \frac{1}{6}\right]$,

avec a, b et c définis en (2.1). On procède de même pour les 6 autres équations.

On obtient ainsi un système de 7 équations à 7 inconnues (les valeurs aux noeuds f_1, f_2, \dots, f_7) :

$$\begin{aligned} \frac{23}{6} f_1 - \frac{14}{3} f_2 + \frac{17}{6} f_3 - \frac{1}{72} &= 0 \\ 7 f_3 + 5 f_1 - 12 f_2 - \frac{1}{12} &= 0 \\ \frac{22}{3} f_4 + \frac{14}{3} f_2 - 12 f_3 - 1/6 f_5 + \frac{1}{6} f_1 - \frac{1}{6} &= 0 \\ 7 f_5 - 12 f_4 + 5 f_3 - \frac{1}{4} &= 0 \\ -12 f_5 + \frac{14}{3} f_4 + \frac{22}{3} f_6 + \frac{1}{6} f_3 - \frac{1}{6} f_7 - \frac{1}{3} &= 0 \\ 5 f_5 + 7 f_7 - 12 f_6 - \frac{5}{12} &= 0 \\ \frac{14}{3} f_6 - \frac{29}{6} f_7 + \frac{1}{6} f_5 - \frac{17}{72} &= 0 \end{aligned}$$

La solution de ce système donne les valeurs aux noeuds de l'approximation $\tilde{\mathbf{f}}(\mathbf{x})$ cherchée :

$$f_1 = \frac{4107}{54872} \quad f_2 = -\frac{7063}{8664} \quad f_3 = -\frac{236893}{164616}$$

$$f_4 = -\frac{25563}{13718} \quad f_5 = -\frac{350779}{164616}$$

$$f_6 = -\frac{374605}{164616} \quad f_7 = -\frac{127275}{54872}$$

Les choix faits précédemment (les fonctions tests et les fonctions d'interpolation) conduisent à un système d'équations non symétrique. On verra plus loin comment obtenir des systèmes symétriques, pour lesquels on dispose d'algorithmes numériques plus efficaces.

Les f_i étant déterminées, on connaît alors les interpolations dans les trois sous domaines, et donc aussi la solution approchée en juxtaposant ces interpolations.

2.6 Examen des résultats

La figure 6 montre une comparaison entre la solution exacte et la solution approchée.

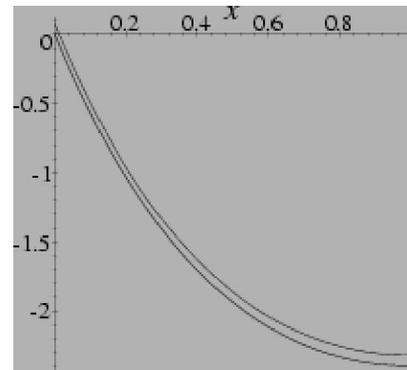


Figure 6: Comparaison des solutions exacte et approchée

On peut noter que puisqu'on n'a pas imposé aux fonctions $\tilde{\mathbf{f}}(\mathbf{x})$ de respecter exactement les conditions aux limites, celles-ci ne le sont qu'approximativement:

$$\tilde{f}(0) \simeq 0.0748 ; \left[\frac{d\tilde{f}(x)}{dx} \right]_{x=1} \simeq 0.0394$$

Bien que cela n'apparaisse pas clairement sur la figure 6, les dérivées sont bien discontinues :

$$\lim_{x \rightarrow \frac{1}{6}} \frac{d\tilde{f}(x)}{dx} \simeq -2.944 ; \lim_{\frac{1}{6} \leftarrow x} \frac{d\tilde{f}(x)}{dx} \simeq -3.017$$

À titre de comparaison on donne figure 7 le graphe de la solution approchée avec le même maillage mais avec des éléments à interpolation linéaire à deux noeuds. L'espace des $\tilde{\mathbf{F}}(\mathbf{x})$ n'est que de dimension 4 et la solution est beaucoup plus grossière.

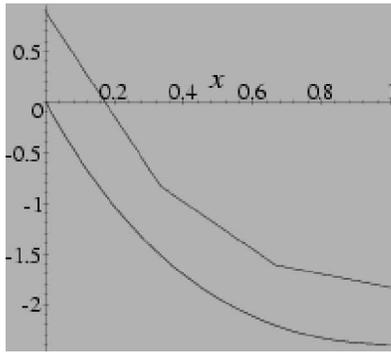


Figure 7: Approximation par 3 éléments linéaires

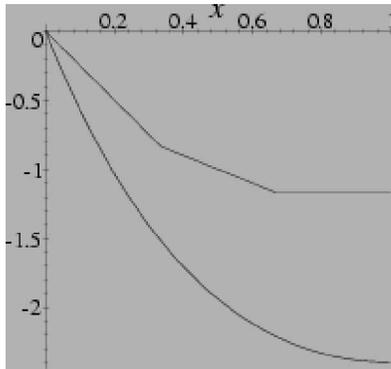


Figure 2.8: Approximation par 3 éléments linéaires, avec C.L. imposées exactes

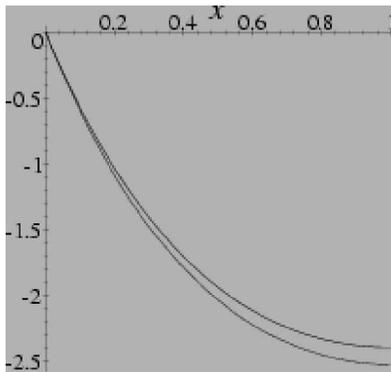


Figure 9: Approximation par 3 éléments d'ordre 2, avec C.L. imposées exactes

On donne aussi figure 8 le graphe avec 3 éléments à interpolation linéaire à deux noeuds où on a astreint les interpolations des éléments 1 et 3 à respecter exactement les conditions aux limites. L'espace des $\tilde{\mathbf{F}}(\mathbf{x})$ n'est plus que de dimension 2.

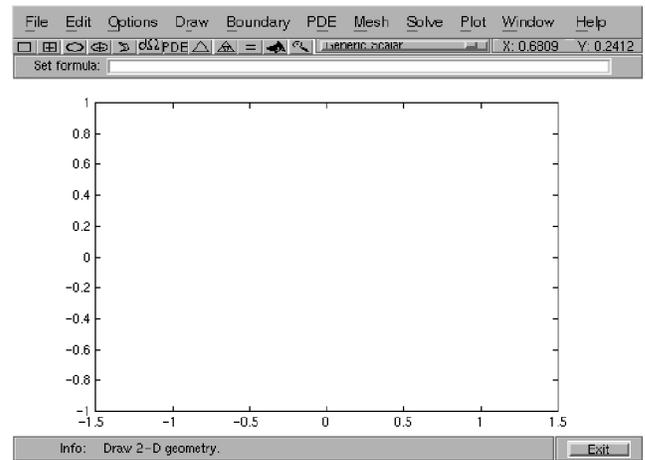
On donne enfin figure 9 le graphe avec 3 éléments à interpolation du second degré à 3 noeuds où on a astreint les interpolations des éléments 1 et 3 à respecter exactement les conditions aux limites. L'espace des $\tilde{\mathbf{F}}(\mathbf{x})$ est de dimension 5.

2. Utilisation de pdetool

Le pdetool sous MatLab est un outil informatique permettant le traitement des équations aux dérivées partielles en deux dimensions en utilisant la méthode des éléments finis. Ce document constitue un guide

simple permettant l'utilisation efficace de cette interface graphique.

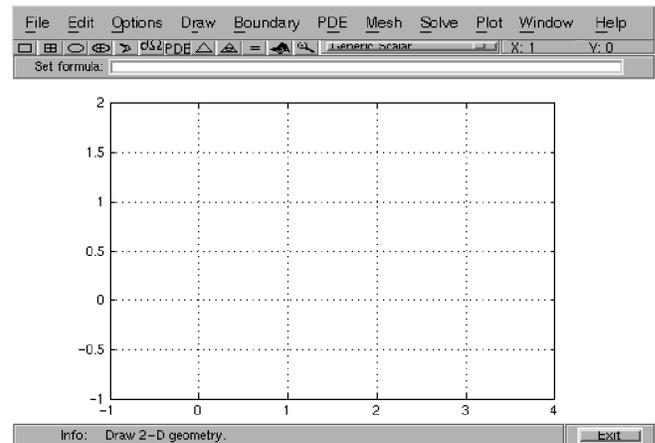
Tapez sous la fenêtre des commandes de MatLab la commande pdetool, vous devez recevoir l'interface graphique suivante.



Nous allons traiter l'équation de Laplace avec des conditions aux limites de type Dirichlet. Ce problème décrit généralement un équilibre des forces, un bilan d'énergie thermique ou électrostatique. L'exemple suivant peut être résolu à la main en faisant appel au développement par séries de Fourier. Ceci permet de vérifier l'exactitude des résultats numériques obtenus par pdetool. Cependant, vous pouvez compliquer par la suite votre problème à des cas non évidents pour un traitement analytique.

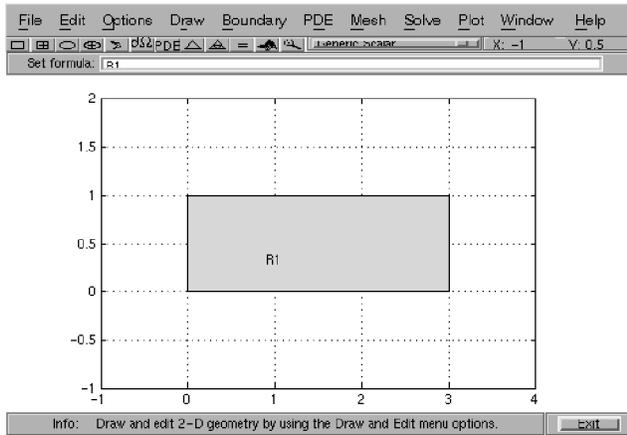
1. Dessin du domaine: supposons qu'il s'agit du rectangle $[0, 3] \times [0, 1]$.

Dans le menu Options, sélectionner Grid and Snap; puis choisir Axes' Limits entre $[-1 \ 4]$ et $[-1 \ 2]$. Faites Apply et fermer la boîte Axes' Limits. Le résultat sera affiché à travers la figure suivante.



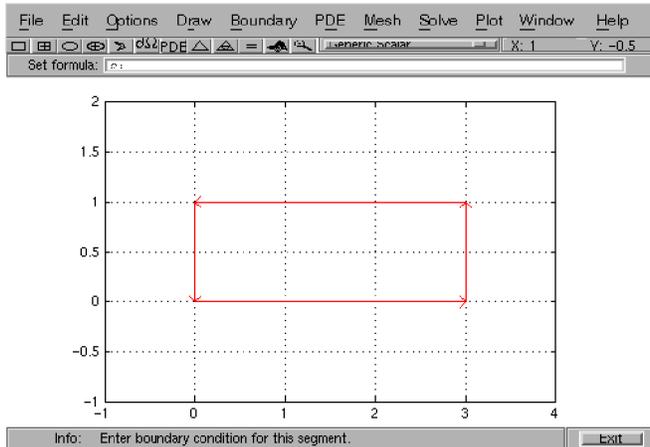
Dans le menu Draw, sélectionner Rectangle/Square (non centré).

Dans la zone du dessin, placer le curseur sur le point (0,0), cliquer en permanence avec le bouton gauche de la souris en déplaçant le curseur jusqu'au point (3,1). La région marquée sera désignée par R1.



2. Passer aux conditions aux limites: donner à la solution u recherché la valeur -1 sur les cotes horizontales, et $+1$ sur les cotes verticales.

Dans le menu Boundary, select Boundary Mode. La figure change comme suit :



Dans la région du dessin, double cliquer sur la cote gauche vertical du rectangle. La boite de dialogue boundary condition apparaît. La condition du type Dirichlet $h \cdot u = r$ est prise par défaut. La valeur $h=1$ est correcte; changer la valeur de r vers $+1$.

| Boundary condition $h \cdot u = r$ | | | |
|--|----------|--------------------------------|-------------|
| Condition | Coeffi | Value | Description |
| <input type="radio"/> Neumann | δ | <input type="text"/> | |
| <input checked="" type="radio"/> Dirichlet | δ | <input type="text"/> | |
| | h | <input type="text" value="1"/> | |
| | r | <input type="text" value="1"/> | |

Faite la même chose pour la cote verticale droite du triangle, c.a.d. donner à r la valeur -1 . Refaire aussi le même travail sur les deux cotes horizontales en donnant à r la valeur $+1$. Fermer la boite de dialogue.

3. Spécification des paramètres de l'équation PDE :

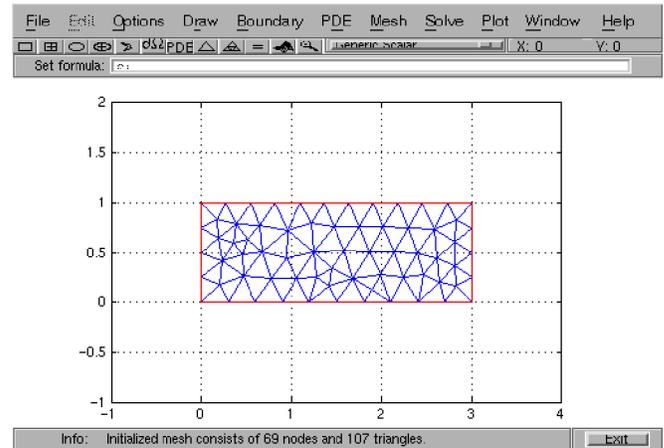
Dans le menu PDE, sélectionner PDE specification. La boite de dialogue pde specification apparaît; le problème elliptique de Laplace $-\text{div}(c \cdot \text{grad}(u)) + a \cdot u = f$, avec $a=0$, $f=0$, et $c=1$ est donné par défaut.

Changer les valeurs comme vous voulez et cliquer sur OK.

| Equatio $-\text{div}(c \cdot \text{grad}(u)) + a \cdot u = f$ | | |
|---|----------|----------------------------------|
| Type of PDE: | Coeffic | Value |
| <input checked="" type="radio"/> Elliptic | c | <input type="text" value="1.0"/> |
| <input type="radio"/> Parabolic | a | <input type="text" value="0.0"/> |
| <input type="radio"/> Hyperbolic | f | <input type="text" value="0"/> |
| <input type="radio"/> Eigenmodes | δ | <input type="text" value="1.0"/> |

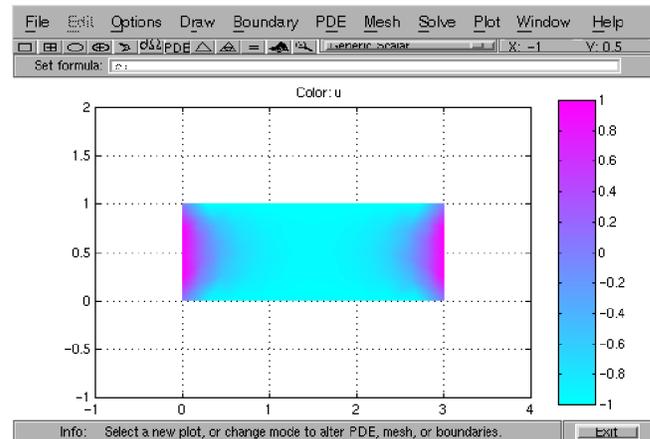
4. Génération du maillage:

Dans le menu Mesh, sélectionner Initialize Mesh. Vous obtenez un maillage régulier et standard.



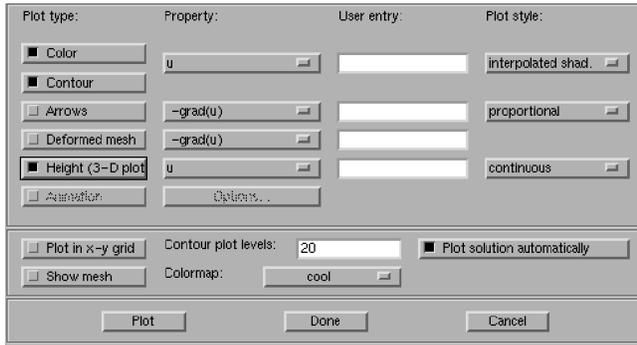
5. Résoudre le problème par la méthode éléments finis:

Dans le menu Solve, sélectionner Solve PDE. La figure dans la section du dessin change vers la solution du problème.

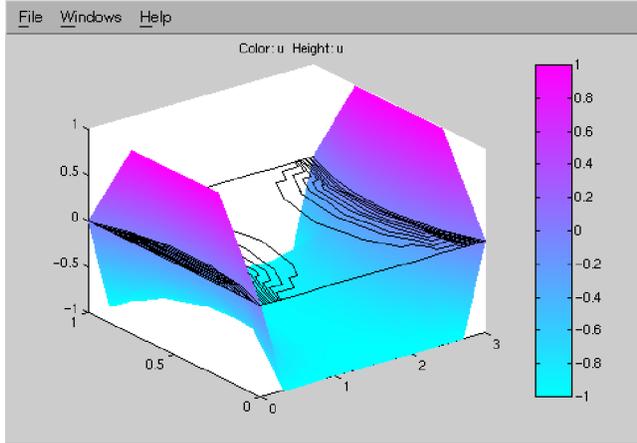


6. Génération des autres présentations pour la solution:

Dans le menu Plot, sélectionner Parameters. La boite de dialogue plot selection apparaît; cliquer sur les boutons de votre choix pour le type de présentation, les paramètres des abscisses et des ordonnées, la couleur et terminer en appuyant sur le bouton plot.

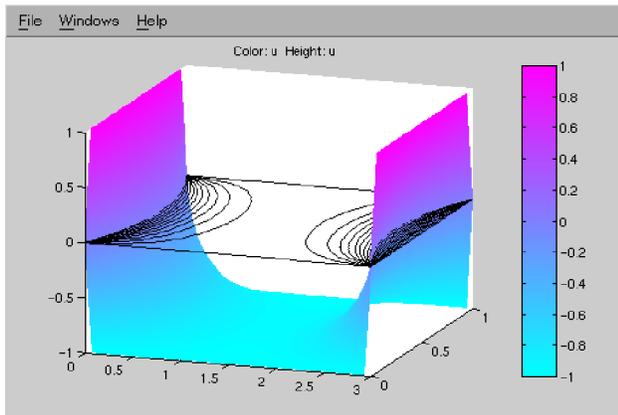


Si vous adoptez les paramètres de la figure ci-dessus, la nouvelle figure suivante apparaît.



En choisissant le bouton Done, vous annulez l'action de la boîte de dialogue plot selection.

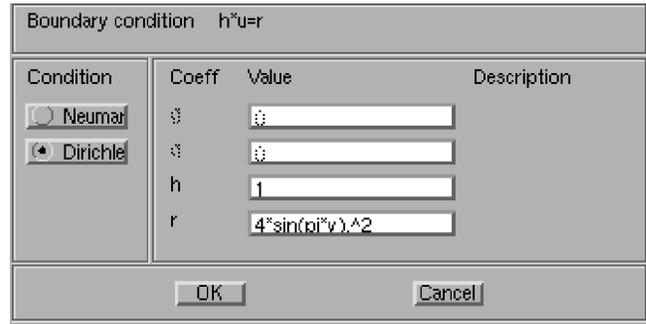
La forme des contours n'est pas assez satisfaisante, vous voulez un résultat plus précis. Revenez au menu Mesh, et sélectionner Refine mesh; Résoudre de nouveau; vous pouvez répéter ce processus autant de fois que c'est nécessaire. Aussi, avec le bouton gauche de la souris, vous pouvez changer l'angle de vision de la figure comme montré ci-dessous.



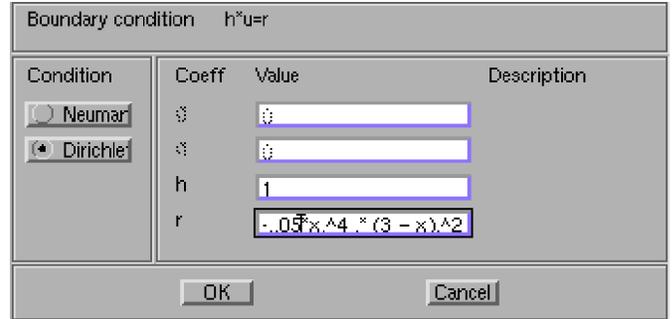
Résolution d'un problème similaire: L'équation de Laplace avec des conditions aux limites différentes.

1. Retournez au mode Boundary.

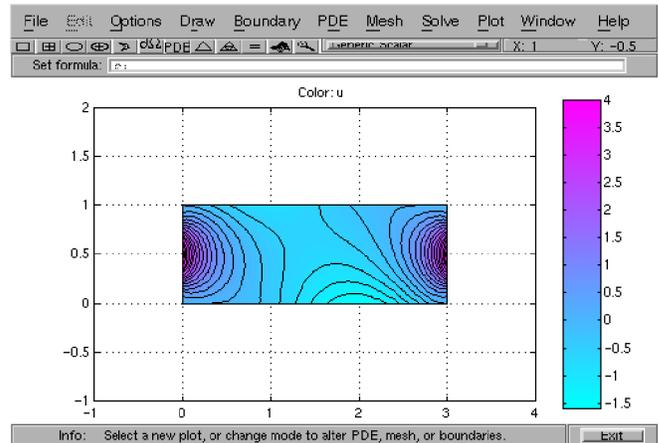
Choisir pour les cotés verticales du rectangle la condition au limite suivante $4 * \sin(\pi * y) .^2$.



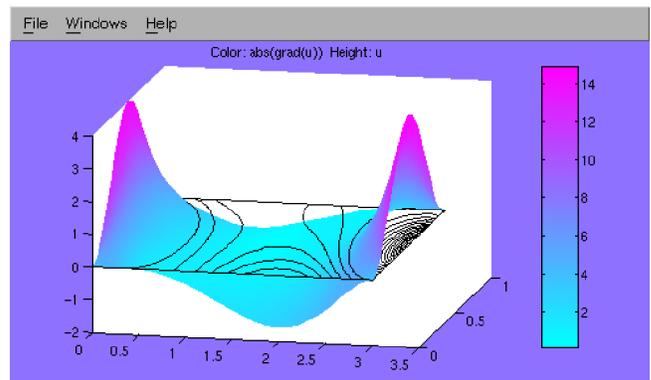
Donner pour la cote horizontale inférieure la condition $-.05 * x.^4 .* (3 - x).^2$ et pour la cote supérieur horizontale la condition $-.05 * x.^2 .* (3 - x).^4$.



2. Après résolution vous obtenez:



Sur la figure suivante, nous avons choisi la représentation graphique 3D de u, avec indices de couleur variant en fonction de "abs(grad(u))". Vous pouvez toujours utiliser la souris pour changer l'angle de vue de la figure.



Pour sauvegarder votre travail, cliquer dans le menu File sur save as... Donner à votre programme le nom "monpde1.m" et quitter MatLab. Plus tard, vous pouvez reprendre l'exécution de toutes les phases de votre programme en tapant sous la fenêtre des commandes la simple commande monpde1.

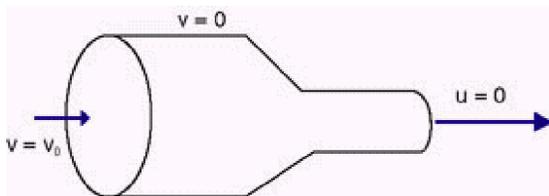
Il existe dans la barre du menu une série des boutons pouvant remplacer un nombre assez important des commandes contenues dans les sous rubriques de chaque menu. Il est important de les remarquer et les retenir.

4. Applications

4.1. Ecoulement dans un convergent :

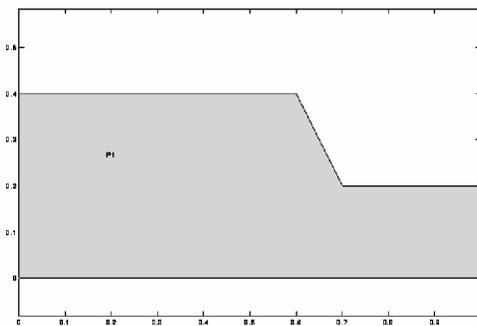
C'est un exemple d'écoulement potentiel, une application classique dans la mécanique des fluides incompressibles. Il s'agit d'un fluide newtonien qui traverse une tube conique.

Si le flux de fluide est irrotationnel, alors le vecteur v de vitesse est exprimé comme gradient d'un potentiel inconnu u , c.-à-d. $v = \text{grad}(u)$. Avec la densité constante, la conservation de la masse donne $\text{div}(v) = 0$, et nous devons ainsi résoudre l'équation $\text{div}(\text{grad}(u)) = 0$.



Le fluide entre de la gauche avec une vitesse constante $v = v_0$. Il sort à la droite avec une vitesse inconnue, mais avec un potentiel constant $u = 0$. Le débit est nul par les autres segments de frontière.

Le domaine numérique:



Le tube convergent dans un repère cylindrique.

En raison de la symétrie, nous pouvons ramener les coordonnées de cylindrique (x, r, θ) du problème à seulement deux coordonnées, (x, r) . À la frontière de création récente, nous présentons l'écoulement parallèle à l'axe des abscisses, ce qui induit une autre condition aux limites Neumann.

Conditions aux limites

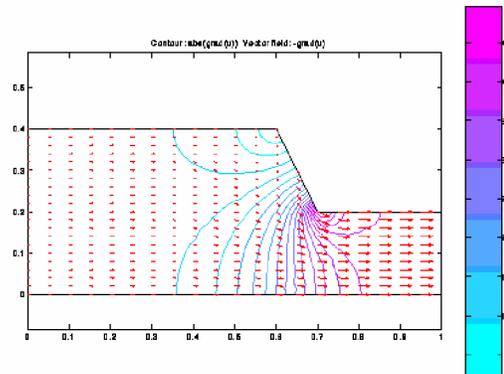
Sur les frontières pleines, la vitesse dans la direction normale disparaît, qui implique des états de frontière de Neumann, c.-à-d., vous indiquez le composant normal du $\text{grad}(u)$ à l'apport, la vitesse est perpendiculaire à la frontière, aussi un état de Neumann. La vitesse sur toutes les frontières ne peut pas déterminer le potentiel, parce qu'ajouter des constantes à u ne change pas v . L'écoulement est régi par la différence potentielle plutôt que le potentiel lui-même. À la sortie $x=1$, nous ne savons pas la vitesse, mais nous pouvons assumer un écoulement laminaire loin du goulot. Là nous pouvons fixer le potentiel à la constante, par exemple, $u=0$, qui est un état de frontière de Dirichlet.

L'équation

Employer des symétries et les coordonnées de cylindrique, les changements d'équation à : $-y * \text{grad}(u) = 0$ de $\text{div}(\cdot)$. Note : À l'apport devient vous devez indiquer le composant normal du $y * \text{grad}(u)$, qui est $10y$.

Solution

L'équation est linéaire même si les coefficients sont variables. Appuyer sur le bouton de solution produit la solution, qui est le potentiel u . Le champ v de vitesse est le gradient du potentiel et est visualisé en utilisant la parcelle de terrain de flèche pour le $\text{grad}(u)$.



Les lignes de niveau donne la taille de v , alors que les flèches donnent la distribution de vitesse.

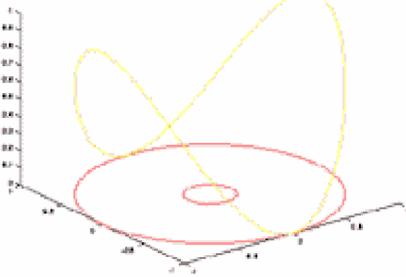
Le flux est inverse proportionnel au secteur de la section transversale. Puisque le secteur de la sortie est 4 fois plus petit que de l'apport, la vitesse augmente par un facteur 4, clairement vu en comparant la longueur des flèches au dans et à la sortie. Notez la singularité au coin de réentrée, où la vitesse devient infinie. Ceci indique la validité limitée du modèle potentiel d'écoulement pour des problèmes réalistes d'écoulement.

4.2 The Soap Bubble Problem

A soap film bounded by a wire frame has the shape determined by the minimization of its internal energy. The shape of the film is described by the *height* $u(x,y)$ over a given domain. The height is fixed at the domain boundary, by the height of the wire frame g . If the film is homogeneous, minimizing the internal energy is equivalent to minimizing its area, the integral of $s(u) = \sqrt{1+|\text{grad}(u)|^2}$.

This in its turn is equivalent to finding u such that

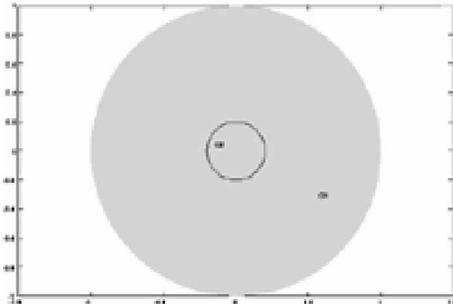
- $-\text{div}(\text{grad}(u)/s(u)) = 0$, inside the domain
 - $u(x,y) = g(x,y)$, on the domain boundary
- In this example you look at a wire frame with two circular components. The inner frame is at height 0 while the outer frame is given by $g(x,y) = x^2$.



The wire frame for a soap bubble. The computational domain

Geometry

First you draw the domain which is an annulus (a two-dimensional doughnut), i.e. the region between two concentric circles:



An annulus obtained by *subtracting* the small disk from the large one. This is expressed by the set formula **C1-C2**

Boundary conditions

Next you set the boundary conditions, $u = 0$ on the inner frame, and $u(x,y) = x^2$ on the outer frame:

| Condition type: | Coefficient | Value | Description |
|--|-------------|-------|-------------|
| <input type="radio"/> Neumann | g | 0 | |
| <input checked="" type="radio"/> Dirichlet | q | 0 | |
| | h | 1 | |
| | r | x^2 | |

Boundary conditions dialog box.

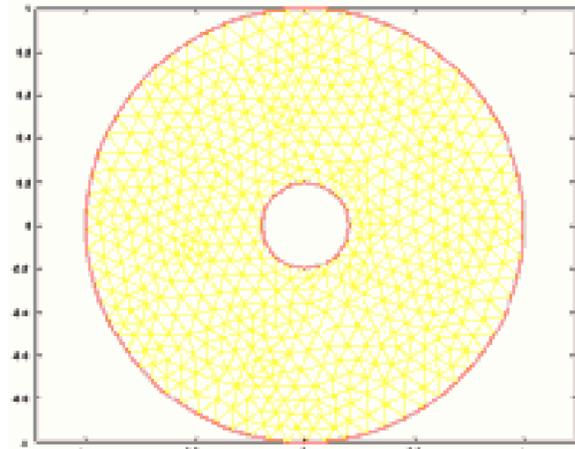
PDE Coefficients

| Type of PDE: | Coefficient | Value |
|---|-------------|-----------------------------|
| <input checked="" type="radio"/> Elliptic | c | $1./\sqrt{1+u_x.^2+u_y.^2}$ |
| <input type="radio"/> Parabolic | a | 0 |
| <input type="radio"/> Hyperbolic | f | 0 |
| <input type="radio"/> Eigenmodes | d | 1.0 |

Setting the equation coefficients amounts to typing in MATLAB statements. These can be constants, expressions containing the space variables x and y , the unknown u , its partial derivatives u_x and u_y , or even user-defined functions. The general form of the time-independent, elliptic, scalar PDE is $-\text{div}(c*\text{grad}(u)) + a*u = f$

In the soap bubble problem, you have $a = f = 0$ and $c = 1/\sqrt{1+u_x.^2+u_y.^2}$.

Mesh generation

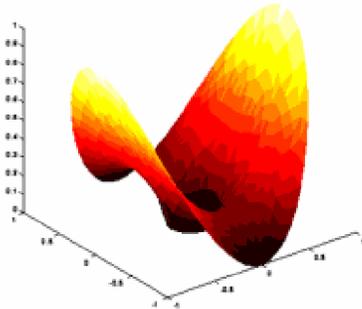


A triangularization of the domain

The mesh generator initializes a coarse mesh. Further refinements of the mesh are necessary to obtain a reasonable solution. Mesh generation is done automatically, but it you can also steer it by setting parameters. In this example the default mesh is initialized and then uniformly refined. The fine mesh contains 590 nodes and 1096 triangles.

Solvers

The Finite Element Method discretization is performed. Linear problems are solved by MATLAB's internal solver. However, the soap bubble equation is nonlinear, and you have to tell that to the solver. After setting the nonlinear option among the Solve parameters, the equations are discretized and solved. A damped Gauss-Newton iterative is employed. Set the Jacobian option to **Full**. While the solver works, you can see the reduction of the residual in your workspace window.



The shape of the soap film

4.3 Electrical Heating

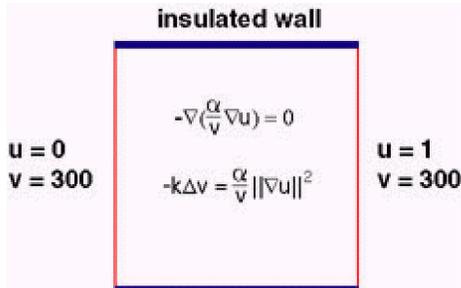
This is an example of a non-linear system of two state-variables, electric potential u and temperature v , in equilibrium.

An electric current passing through a metal will produce heat. It is known that the electric conductivity depends on the temperature of the conductor. Thus the heat and current distributions are related to each other.

The electric potential u satisfies the equation $-\text{div}(s \cdot \text{grad}(u)) = 0$, where $s = s(v)$ is a function of the temperature v . The current density $-s \cdot \text{grad}(u)$ produces heat at the rate $s|u|^2$. This is a source term in the heat equation. Over a wide range of temperatures, the electric conductivity is inverse proportional to the temperature. The system of PDEs modeling the electric and heat conductance is:

- $-\text{div}(s(v) \cdot \text{grad}(u)) = 0$
- $-K \cdot \text{div}(\text{grad}(v)) = s(v) / |\text{grad}(u)|^2$

Imagine a rectangular metal plate with a potential difference of unit size applied on two opposite faces. Those faces are at "room" temperature, 300 degrees Kelvin. The other two faces are insulated.



A metal plate with an electric current heating it.

The geometry is very simple, a unit square. Select the **Generic system** mode where you define the boundary conditions and the equations.

Boundary conditions

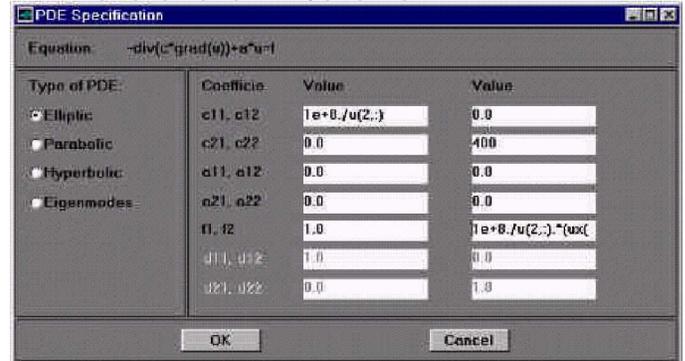
There are two kind of boundary conditions: On the insulated sides, we select zero (default) Neumann boundary conditions. On the other two sides, we select Dirichlet boundary conditions, setting the potentials $r1$ to 0 and 1, and the temperature $r2$ to 300.

The Equation

The generic system is:
 $-\text{div}(c11 \cdot \text{grad}(u)) - \text{div}(c12 \cdot \text{grad}(v)) + a11 \cdot u + a12 \cdot v = f1$

$$-\text{div}(c21 \cdot \text{grad}(u)) - \text{div}(c22 \cdot \text{grad}(v)) + a21 \cdot u + a22 \cdot v = f2$$

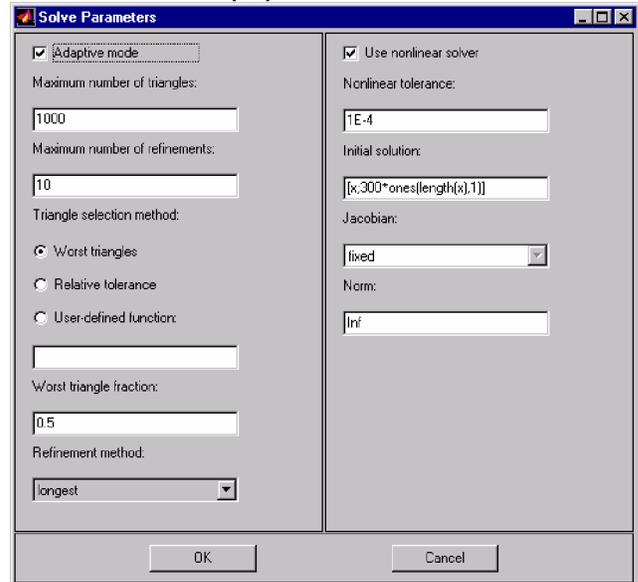
In this case, $c11 = 1e+8./u(2,:)$, $c22 = 400$, and $f2 = 1e+8./u(2,:).*(ux(1,:).^2+uy(2,:).^2)$.



Specification of a nonlinear system. u, ux, uy contain two rows, one for each unknown.

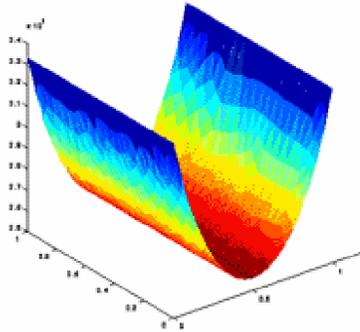
Mesh generation and solving

One uniform refinement of the default original mesh produce enough mesh-points (689 points in the unit square). Next you have to set the options for the solver. You have to tell the solver that the problem is nonlinear and you can "help" by giving a good initial guess of the solution. There are other parameters that are used for fine-tuning the solver in specific cases, and these should be used when the default values fail. The default initial guess $u=v=0$ will fail because the current density becomes infinite. You know the temperature on two sides, so set $u(2,:)=300$. We also know that the potential varies, e.g., from 0 to 1 in the x-direction, so set $u(1,:)=x$.



Options in the nonlinear solver. Click to enlarge.

Pressing the solve-button starts the nonlinear iterations (a damped Gauss-Newton iteration). The nonlinear solver performs 17 steps to reduce the residual to the specified tolerance. Using the **full** approximation of the Jacobian, the solution is found (up to the same tolerance) in only 4 steps. Since the computing time for the full Jacobian is long, this strategy should be used only for extremely tight tolerances.

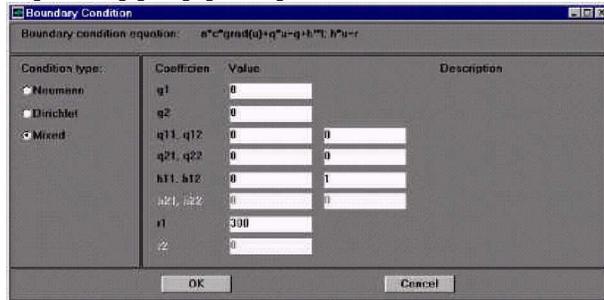


Heating due to electric currents.

Using the plot preferences, you can put different properties on the same image. The color scale is a thermometer showing the heat distribution in the metal plate. Note that the change in electric conductance is not small. This means that the equation for the electric potential u cannot be uncoupled from the heat equation.

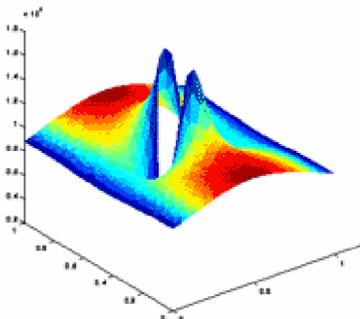
Mixed Boundary conditions

Note that the middle section of the plate becomes very hot. You can cool it by circulating "room temperature" air through a hole. Carve out a circular hole with diameter 0.2 in the center of the plate. At the new boundary, the temperature $u(2,:)$ is fixed again to 300 degrees Kelvin. Since this boundary is electrically insulated, the current passing through it is zero, i.e. a Neumann condition is needed. This is an example of mixed Neumann-Dirichlet conditions. The Dirichlet condition is expressed by the constraint $h*[0 \ 1; 1 \ 0]*[u;v]=[300;0]$



Mixed Neumann-Dirichlet boundary conditions. The electric potential satisfies a Neumann condition, while the temperature is constrained by the Dirichlet condition $v=300$.

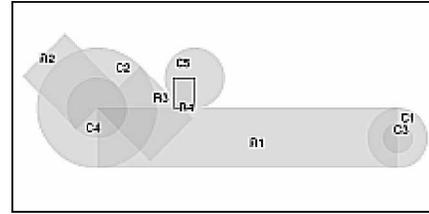
The cooling effect is substantial. The maximum temperature drops from 117 to 83 Celsius:



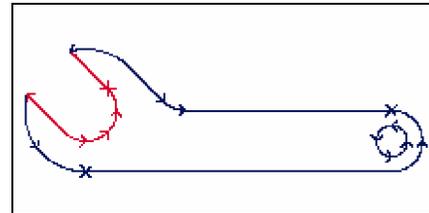
Cooling the heated plate.

The air cooling through the middle of the plate produces a significant drop in the temperature. Note the strong variations in the current density around the cooling hole, which makes the problem genuinely nonlinear.

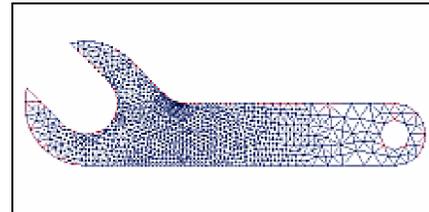
4.4. Stresses in a Wrench



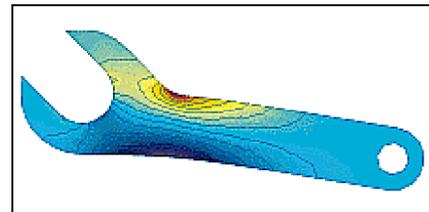
In draw mode you create a Constructive Solid Geometry model of the geometry. Different building blocks can be combined using a set formula. The picture on the right shows the Constructive Solid Geometry model of a wrench.



Define the boundary conditions by double clicking on the boundary segments. You can select Dirichlet, Neumann or mixed boundary conditions. Using the Dirichlet boundary condition you clamp the "claw" part of the wrench and with the Neumann boundary condition you apply a downward force to the handle.



In mesh mode the geometry is automatically meshed. The picture to the right shows the result of the adaptive mesh refinement algorithm. The triangular mesh has been refined the most where the variation in stress is large.



In plot mode you have an almost unlimited number of plot options. The picture on the left shows the x direction stresses visualized using contour and color plots. The plot clearly shows tension (red) and compression (dark blue) in different parts of the wrench. The displacements are shown by the distorted figure.