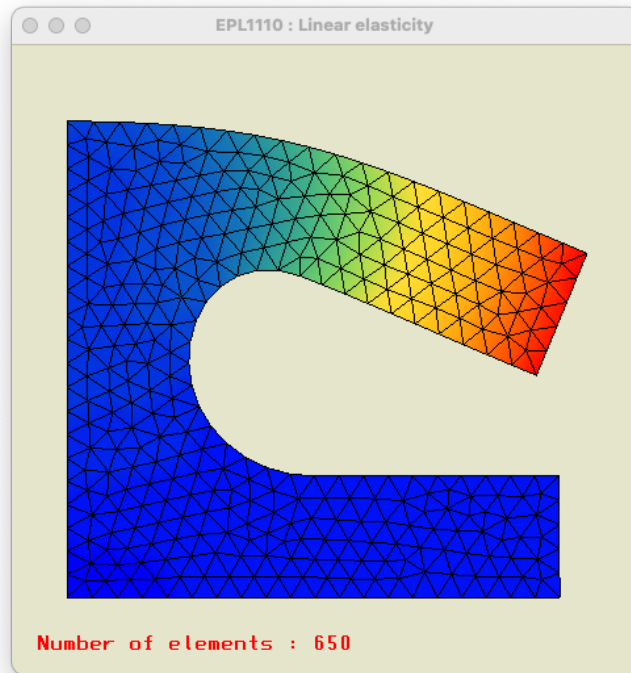


*Nathan Coppin, Miguel De Le Court, Insaf Draoui, Michel Henry,
Ange Ishimwe, Vincent Legat, Thomas Leyssen, Simon Yans*

1 Elasticité linéaire et éléments finis



L'objectif du projet est de vous initier aux difficultés de la mise au point et de la certification d'une application numérique. Rassurez-vous : il ne s'agit nullement de vous transformer en experts de l'architecture de grandes applications numériques ! Nous nous limiterons à l'écriture d'un tout petit programme C pour l'élasticité linéaire.

Objectifs du projet...

L'objectif du projet sera double cette année !

1. Tout d'abord, il vous sera demandé d'écrire un code informatique efficace, bien documenté et robuste pour résoudre des problèmes d'élasticité linéaire en deux dimensions. Votre programme devra pouvoir résoudre des problèmes en tensions ou en déformations planes ou des problèmes axisymétriques. Il devra pouvoir fonctionner avec des éléments triangulaires linéaires ou quadrilatères bilinéaires. En outre, vous devrez être capables de retrouver les composantes du champ de contraintes aux noeuds du maillage. Pour que votre programme soit efficace, il faut utiliser soit un solveur bande, soit un solveur frontal, soit un solveur itératif tel que les gradients conjugués.
2. Ensuite, il vous est demandé de trouver un problème à résoudre avec votre code : cela signifie écrire un programme pour générer le maillage correspondant, définir les paramètres matériels et analyser la solution trouvée. Et surtout, nous convaincre que simuler ce problème a un réel intérêt pratique !

Pour cette seconde partie, vous aurez un assistant de référence pour chaque groupe afin de lui proposer un problème original à résoudre ou éventuellement afin de lui demander de vous suggérer quelques idées de problèmes possibles : regardez aussi le monde qui vous entoure et repérez tous les pièces métalliques ou structures de la vie de tous les jours : support d'étagère, tuyauteries diverses, diapason, ponts, voutes de cathédrale ou que je sais encore.... Ne soyez pas trop ambitieux : la simplicité est aussi la clé du succès.

Evidemment, les deux derniers devoirs **GeoMesh** et **LinearElasticity** contiennent tous les indices pour réaliser votre projet. Dans la solution du dernier devoir, vous avez déjà la résolution des équations de l'élasticité linéaires pour des triangles linéaires et des quadrilatères bilinéaires avec un solveur plein... Il ne reste donc qu'à généraliser et à optimiser ce qui a été réalisé dans ce projet.

Les cours du vendredi 28 mai et du vendredi 5 juin seront aussi consacrés à la présentation des aspects plus techniques du projet et serviront de séances globales de questions-réponses. Finalement, les séances d'exercices pourront être efficacement utilisées comme séance de consultation auprès des tuteurs et des assistants.

2 Ce qu'il s'agit de faire !

2.1 Ecrire un code d'éléments finis !

On vous demande d'écrire un code C d'éléments finis qui résout de manière efficace un problème d'élasticité linéaire plane. Ce code n'aura aucune option graphique et ne pourra faire appel à aucune librairie externe : en particulier, vous ne pouvez pas utiliser la librairie de **gms** ou des solveurs itératifs tels que **petsc** ou d'autres choses. La totalité de l'implémentation doit être faite par vos soins : il est permis de vous inspirer partiellement du contenu **fem.c** et **fem.h** pour votre code, mais il est requis de bien isoler cela dans des fichiers distincts. En cas de doute, n'hésitez pas à consulter votre assistant(e) de référence. On utilisera une règle d'intégration à trois points sur les triangles, à quatre points sur les quadrilatères telles que définis dans le dernier devoir. Pour les intégrales sur les segments frontières, on utilisera une règle de

Le code sera soumis sous la forme d'un fichier **group001-vlegat-jfremacle.zip** contenant l'ensemble des sources du programme, un fichier **CMakeLists.txt** et un fichier **ReadMe.txt** contenant les instructions détaillées pour faire fonctionner le code. L'arborescence des fichiers correspondra à celle des devoirs et le nom du répertoire respectera exactement la forme suivante :

```
group001-vlegat-jfremacle\build
group001-vlegat-jfremacle\CMakeLists.txt
group001-vlegat-jfremacle\ReadMe.txt
group001-vlegat-jfremacle\data
group001-vlegat-jfremacle\src
```

Il faut évidemment adapter le numéro de groupe et les deux identifiants à votre cas particulier !

Le programme acceptera deux arguments de lignes pour spécifier les noms de deux fichiers. Par défaut, le programme ouvrira un fichier **data/mesh.txt** contenant le maillage et un fichier **data/problem.txt** contenant la description du problème. Les fonctions pour écrire et lire le maillage sont disponibles dans le dernier devoir et un exemple est également fourni pour le fichier décrivant le problème.

Le programme fournira les résultats obtenus dans les fichiers indiqués par le fichier **data/problem.txt**.

2.2 Concevoir et résoudre un problème particulier !

La seconde étape est la partie réellement créative du projet : il faut trouver, imaginer, découvrir un problème pour lequel vous allez utiliser votre programme ! Evidemment, vous pourriez vous contenter de résoudre le problème du dernier devoir, mais ce n'est ni très excitant, ni très original ! On vous demande donc de nous proposer un problème que vous allez résoudre :-)

Evidemment, si votre problème fait intervenir de la **transition**, de l'**innovation** ou de la **diversité** : c'est encore mieux ! La transition, c'est faire des éléments finis recyclables, durables et verts, l'innovation c'est faire de nouvelles choses et la diversité, c'est s'ouvrir à tous les éléments finis présents dans notre sympathique communauté universitaire :-)

Là, aussi c'est le défi qu'il vous appartient de relever avec brio !

Il sera donc nécessaire de créer par vous-même une géométrie comme vous l'avons fait dans le devoir **GeoMesh** et puis de visualiser vos résultats de manière originale et efficace.... Il s'agit donc d'écrire d'autres petits programmes qui effectuent ces tâches de pré-processing et de post-processing : mais; même si on vous demande de nous fournir les sources des programmes réalisés : ils ne seront ni analysés, ni corrigés.... seul, le résultat final obtenu est important ici ! Réaliser une animation de votre simulation est une option possible, ou même effectuer un test expérimental pour estimer les ordres de grandeur est aussi une option possible. Evidemment, tout sera considéré comme d'éventuels bonus.

Et finalement, il s'agira d'écrire un rapport de maximum 4 pages (sans page de garde !). Dans ce rapport, il vous est demandé d'expliquer la physique du problème analysé, d'analyser les résultats obtenus. Il est aussi demandé d'expliquer comment vous avez validé votre code et estimé la performance de ce dernier.

Les données pour le problème analysé et le rapport seront fourni par dans un fichier **group001-vlegat-jfremacle-rapport...** avec l'arborescence suivante :

```
group001-vlegat-jfremacle-rapport\build
group001-vlegat-jfremacle-rapport\CMakeLists.txt
group001-vlegat-jfremacle-rapport\ReadMe.txt
group001-vlegat-jfremacle-rapport\data\mesh.txt
group001-vlegat-jfremacle-rapport\data\problem.txt
group001-vlegat-jfremacle-rapport\rapport\group001-vlegat-jfremacle.pdf
group001-vlegat-jfremacle-rapport\rapport\src\
group001-vlegat-jfremacle-rapport\src
```

Il faut y inclure les sources pour compiler les programmes éventuels de post-processing et de pre-processing que vous avez réalisés. Y joindre les fichiers de données du problème analysés avec obligatoirement au moins un fichier **mesh.txt** et un fichier **problem.txt** Dans le répertoire **rapport**, vous incluez obligatoirement le rapport sous format **pdf** avec le nom **group001-vlegat-jfremacle.pdf** : dans le répertoire **rapport\src**, vous êtes invités à inclure les fichiers source **docx** ou **tex** de votre rapport. Si vous avez utilisé **github** pour écrire votre code et/ou votre rapport, vous pouvez l'indiquer dans le fichier **ReadMe.txt**. Il est permis d'éventuellement fournir ce dernier fichier sous le format **ReadMe.md**, si cela correspond davantage à votre manière de travailler.

2.3 Et concrètement ?

Plus précisément, il vous est demandé :

1. de concevoir un programme d'éléments finis pour l'élasticité linéaires avec des triangles linéaires ou

des quadrilatères bilinéaires.

2. d'écrire **le code séquentiel le plus rapide possible** en n'utilisant que les instructions standards du C.
3. d'écrire **le code qui alloue le moins de mémoire possible**¹ et la libère correctement.
4. d'écrire **un code élégant et facile à comprendre**² pour un néophyte.
5. (option) d'implémenter un solveur linéaire frontal.
6. (option) d'implémenter un algorithme de renumération des noeuds ou des éléments.
7. (option) de calculer les valeurs des composantes de tension aux noeuds.
8. de rédiger une note de synthèse d'au maximum 4 pages pour votre mission de conseil. Fournir une estimation de l'ordre de précision du résultat obtenu en expliquant comment vous avez validé votre code numérique. Produire quelques illustrations pertinentes pour l'analyse de la solution. Expliquer comment vous avez optimisé votre programme afin qu'il soit le plus rapide possible. Ne pas recopier les développements théoriques du syllabus, ne pas recopier l'énoncé du problème, ne pas fournir des diagrammes incompréhensibles, ne pas donner des tableaux de chiffres indigestes. L'orthographe, le soin et la présentation seront conformes à celles d'une note fournie par un bureau d'études professionnel.
9. (option) de réaliser une animation de votre simulation et du problème analysé.
10. (option) d'effectuer une analyse expérimentale du problème analysé.

2.4 Evaluation du projet

L'évaluation du projet se fera sur la base suivante sur un total de 20 points :

| | |
|--|---|
| Programme d'éléments finis /10 | |
| Programme qui fonctionne et fournit un résultat qui semble cohérent | 2 |
| Style, élégance du code et gestion de la mémoire | 2 |
| Précision des résultats | 2 |
| Rapidité du code de calcul | 2 |
| Implémentation des éléments quadratiques | 2 |
| (bonus) Implémentation d'un solveur frontal | 4 |
| (bonus) Implémentation d'un algorithme de renumération des noeuds-éléments | 2 |
| (bonus) Implémentation du calcul des tensions aux noeuds | 2 |
| Problème analysé /10 | |
| Analyse physique et originalité du problème | 2 |
| Analyse et validation des résultats | 2 |
| Rapport | 2 |
| Interview | 4 |
| (bonus) Animation | 2 |
| (bonus) Etude de l'état de l'art pour le problème | 2 |
| (bonus) Démarche expérimentale | 2 |
| (bonus) Transition, innovation et diversité | 2 |

¹Evidemment, ce critère peut être vu comme l'inverse du précédent... Car pour éviter des calculs, il est parfois utile de stocker des résultats en mémoire ! Il s'agira donc d'être rapide tout en n'étant pas trop gourmand en mémoire.

²Oui, c'est un critère un peu subjectif, c'est pourquoi ce critère sera évalué par un Jury des assistants et enseignants qui n'ont pas tous les mêmes idées sur le bon style de programmation.

Comme vous pourrez le constater, il est parfaitement possible d'obtenir 20/20 sans réaliser aucun bonus.

Votre projet (programme, rapport, animation) sera soumis à votre assistant de référence selon les modalités qu'il vous indiquera en temps utile : cet assistant de référence testera votre code et vérifiera qu'il arrive bien à l'exécuter. Il est évidemment primordial de rencontrer et d'interagir avec votre assistant de référence avant la date de l'échéance du projet.

Pour rappel, toutes les soumissions seront soumises à un logiciel anti-plagiat. En cas de fraude flagrante, les cas de plagiat seront soumis au Jury des examens. Vous êtes invités à consulter la page web de l'Université pour avoir une petite idée des sanctions possibles dans ce cas !

2.5 Assistant de référence...

Est-ce que cela ressemble aux projets de l'années précédentes ?

Un peu mais on a essayé d'être différents !

En particulier, chaque groupe est vraiment invité à réaliser quelque chose de particulier et de différent des autres groupes, même si vous allez tous réaliser le même programme d'éléments finis. Ecrire le code n'est pas compliqué, mais avoir un rapport et un problème original est un vrai défi : et là, vous ne pourrez pas faire appel à vos condisciples car chaque groupe devra avoir un problème distinct. Pour vous aider dans cet aspect particulier, tous les groupes auront un assistant de référence qu'ils pourront contacter ³

| | Groupes à coacher et à interviewer :-) |
|--------------------|---|
| Nathan Coppin | 2, 9, 17, 24, 31, 40, 47, 57, 64, 71, 78, 85, 92, 99, 106, 113, 120, 127, 134, 141, 148, 155, 162, 169, 176 |
| Miguel De Le Court | 3, 10, 18, 25, 32, 41, 48, 58, 65, 72, 79, 86, 93, 100, 107, 114, 121, 128, 135, 142, 149, 156, 163, 170, 177 |
| Insaf Draoui | 4, 11, 19, 26, 33, 42, 49, 59, 66, 73, 80, 87, 94, 101, 108, 115, 122, 129, 136, 143, 150, 157, 164, 171, 178 |
| Michel Henry | 5, 12, 20, 27, 34, 43, 50, 60, 67, 74, 81, 88, 95, 102, 109, 116, 123, 130, 137, 144, 151, 158, 165, 172, 179 |
| Ange Ishimwe | 6, 14, 21, 28, 36, 44, 51, 61, 68, 75, 82, 89, 96, 103, 110, 117, 124, 131, 138, 145, 152, 159, 166, 173, 180 |
| Thomas Leyssen | 7, 15, 22, 29, 38, 45, 55, 62, 69, 76, 83, 90, 97, 104, 111, 118, 125, 132, 139, 146, 153, 160, 167, 174, 181 |
| Simon Yans | 8, 16, 23, 30, 39, 46, 56, 63, 70, 77, 84, 91, 98, 105, 112, 119, 126, 133, 140, 147, 154, 161, 168, 175, 182 |

Votre assistant de référence sera également présent lors de votre interview : c'est votre allié dans ce projet, il peut vous aider en cas de problème informatique ou de difficultés particulières : c'est évidemment votre interlocuteur privilégié pour le projet.

2.6 Grand Prix International 2023 de l'Elément le Plus Fini

Le meilleur projet (en termes de programmation et d'analyse physique) recevra le Grand Prix de l'Elément le Plus Fini d'un montant de 100 euros. Ce prix a été rendu possible grâce à la contribution anonyme d'un ancien étudiant soucieux de promouvoir la qualité de la formation. La proclamation des résultats se fera sur le web avant le 15 juin 2023. Les gagnants du prix seront invités à prendre contact avec le titulaire du cours. La mesure de la rapidité du programme sera effectuée par deux tests indépendants avec le maillage le plus raffiné. Le classement final sera une moyenne entre le classement en termes de rapidité d'exécution, de l'originalité du problème analysé, de la qualité du rapport et d'esthétique du code.

Les gagnants eu prix acceptent que leur contributions soient publiées sur le site web de cours pour la

³En fonctions de modalités fournies par chaque assistant, soit lors des périodes de consultance qu'il aura définies, soit sur rendez-vous préalable.

postérité des éléments finis.

2.7 Mon binôme a disparu....

Si pour une raison précise, vous ne souhaitez pas faire le projet avec votre binôme. Il faut envoyer une-mail au titulaire du cours ainsi qu'à votre compagnon de travaux. Il faut aussi veiller à envoyer un email à votre binôme défaillant pour l'informer de votre *mise en demeure* de participer au projet. La note du second étudiant sera alors nulle pour le projet. Si cet étudiant s'est désinscrit du cours ou a clairement abandonné son année d'étude, cela ne devrait poser aucun problème pour lui !

Attention : c'est le seul cas de figure admis. Si votre binôme souhaite réellement participer au projet, vous ne pouvez pas l'exclure de manière unilatérale. Mais si sa participation est réellement inexistante ou nulle, vous le mentionnez dans le rapport du projet et nous en tiendrons compte dans la note du projet. Deux étudiants isolés peuvent remettre un projet commun : il suffit aussi de le mentionner dans le rapport soumis. Vous indiquerez clairement dans votre rapport que vous avez fait le projet seul